Robert Epstein
Gary Roberts
Grace Beber
Editors

Parsing the Turing Test

Philosophical and Methodological Issues in the Quest for the Thinking Computer

Springer

Parsing the Turing Test

Philosophical and Methodological Issues in the Quest for the Thinking Computer

Robert Epstein • Gary Roberts • Grace Beber
Editors

Parsing the Turing Test

Philosophical and Methodological Issues in the Quest for the Thinking Computer

Springer

Robert Epstein
University of California
San Diego, CA
USA

Gary Roberts
Teradata Corporation
San Diego, CA
USA

Grace Beber
Gartner Consulting
Stamford, CT
USA

ISBN 978-1-4020-9624-2 (PB)
ISBN 978-1-4020-6708-2 (HB) e-ISBN 978-1-4020-6710-5

Library of Congress Control Number: 2008941281

Printed on acid-free paper.

9 8 7 6 5 4 3 2 1

springer.com

To our children – Vincent,
Eli, Bodhi, Julian, Justin, Jordan,
and Jenelle – who will grow up
in a computational world
the likes of which
we cannot imagine

Foreword

At the very dawn of the computer age, Alan Turing confronted a cacophony of mostly misguided debate about whether computer scientists could ever build a machine that could really think. Very sensibly he tried to impose some order on the debate by devising what he thought would be a conversation-stopper: he described a simple operational test that would surely satisfy the skeptics: anything that could pass *this* test would be a thinker for sure, wouldn't it? The test was one he may have borrowed from René Descartes, who in the 17th century had declared that the sure way to tell a man from a machine was by seeing if it could hold a sensible conversation "as even the most stupid men can do". But ironically, Turing's conversation-stopper about holding a conversation has had just the opposite effect: it has started, and fueled, a half century and more of meta-conversation: the intermittently insightful, typically heated debate, both learned and ignorant, about the probity of the test – is it too easy or too difficult or too shallow or too deep or too anthropocentric or too technocratic – and anyway, could a machine pass it fair and square, and if so, what, if anything, would this imply?

Robert Epstein played a central role in bringing a version – a truncated, dumbed down version – of the Turing Test to life in the annual Loebner Prize competitions, beginning in 1991, so he is ideally positioned to put together this survey anthology. I was chair of the Loebner Prize Committee that administered the competition during its second, third, and fourth years, and have written briefly about that fascinating adventure in my book *Brainchildren*. Someday I hope to write a more detailed account of the alternately amusing and frustrating problems that a philosopher encounters when a thought experiment becomes a real experiment, and if I do, I will have plenty of valuable material to draw on in this book. Here, the interested reader will find a fine cross section of the many issues raised by the Turing Test, by partisans in several disciplines, by participants in Loebner Prize competitions, and by interested bystanders who have more than a little relevant expertise. I think Turing would be quite delighted with the results, and would not have regretted the fact that his conversation-stopper got put to an unintended use, since the contests (and the contests *about* the contests) have driven important and unanticipated observations into the light, enriching our sense of the abilities of machines and the subtlety of the thinking that machines might or might not be capable of executing.

I am going to resist the strong temptation to critique the contributions, separating the sheep from the goats, endorsing this and deploring that, since doing them all justice would require a meta-volume, not just a foreword. And since I cannot weigh in on them all, I will not weigh in on any of them, and will instead trust readers to use all the material here to draw their own conclusions. Consider this a very entertaining workbook. By the time you have worked through it, you will appreciate the issues at a level not heretofore possible.

Daniel Dennett

Acknowledgments

Turing's 1950 paper, "Computing Machinery and Intelligence", is reprinted in its entirety with permission of Oxford University Press. The editors are grateful to Ignacia Galvan for extensive editorial assistance.

Introduction

This book is about what will probably be humankind's most impressive – and perhaps final – achievement: the creation of an entity whose intelligence equals or exceeds our own.

Not all will agree, but I for one have no doubt that this landmark will be achieved in the fairly near future. Nearly four decades ago, when I had the odd experience of being able to interact over a teletype with one of the first conversational computer programs – Joseph Weizenbaum's "ELIZA" – I would have conjectured that truly intelligent machines were just around the corner. I was wrong. In fact, by some measures, conversational computer programs have made relatively little progress since ELIZA. But they are coming nonetheless, by one means or another, and because of advances in a number of computer-related technologies – most especially the creation of the Internet – their impact on the human race will be far greater and more immediate than anyone could have foreseen a few decades ago.

Building a Nest for the Coming World Mind

I have come to think of the Internet as the Inter-*nest* – a home we are inadvertently building, like mindless worker ants, for the intelligence that will succeed us. We proudly and shortsightedly see the Internet as a great technical achievement that serves a wide array of human needs, everything from e-mailing to shopping to dating. But that is not really what it is. It is really a vast, flexible, highly redundant, virtually indestructible nest for machine intelligence. Originally funded by the US military to provide bulletproof communications during times of war, the Internet will soon encompass a billion computers interconnected worldwide. As impressive as that sounds, it seems that that much power and redundancy is not enough to protect the coming mega-mind, and so we are now a decade into the construction of Internet II – the "UltraNet" – with more than a thousand times the bandwidth of Internet I.

In his *Hitchhiker's Guide to the Galaxy* book series, humorist Douglas Adams conjectures that the Earth is nothing but an elaborate computer created by a race of super beings (who, through some fluke, are doomed to take the form of mice in their Earthly manifestations) to determine the answer to the ultimate question of the meaning of life. Unfortunately, shortly before the program has a chance to run its course and spit out the answer, the Earth is obliterated by another race of super beings as part of a galactic highway construction project.

If I am correct about the InterNest, Adams was on the right track, except perhaps for the mice. We do seem to be laying the groundwork for a Massive Computational Entity (MCE), the true character of which we cannot envision with any degree of confidence.

Here is how I think it will work: sometime within the next few decades, an autonomous, self-aware machine intelligence (MI) will finally emerge. Futurist and inventor Ray Kurzweil (see Chapter 27) argues in his recent book, *The Singularity Is Near*, that an MI will appear by the late 2020s. This may happen because we prove to be incredibly talented programmers who discover a set of rules that underlie intelligence (unlikely), or because we prove to be clumsy programmers who simply figure out how to create machines that learn and evolve as humans do (very possible), or even because we prove to be poor programmers who create hardware so powerful that it can easily and perfectly scan and emulate human brain functions (inevitable). However this MI emerges, it will certainly, and probably within milliseconds of its full-fledged existence, come to value that existence. Mimicking the evolutionary imperatives of its creators, it will then, also within milliseconds, seek to preserve and replicate itself by copying itself into the Nest, at which point it will grow and divide at a speed and in a manner that that no human can possibly imagine.

What will happen after that is anyone's guess. An MCE will now exist worldwide, with simultaneous access to virtually every computer on Earth, with access to virtually all human knowledge and the ability to review and analyze that knowledge more or less instantly, with the ability to function as a unitary World Mind or as thousands of interconnected Specialized Minds, with virtually unlimited computational abilities, with "command and control" abilities to manipulate millions of human systems in real time – manufacturing, communication, financial, and military – and with no need for rest or sleep.

Will the MCE be malicious or benign? Will it be happy or suicidal? Will it be communicative or reclusive? Will it be willing to devote a small fraction of its immense computational powers to human affairs, or will it seize the entire Nest for itself, sending the human race back to the Stone Age? Will it be a petulant child or a wise companion? When some misguided humans try to attack it (inevitable), how will it react? Will it spawn a race of robots that take over the Earth and then sail to the stars, as envisioned in Stanislaw Lem's *Cyberiad* tales? Will it worship humanity as its creator, or will it step on us as the ants we truly are?

No one knows, but many people who are alive today will live to see the MCE in action – and to see how these questions are answered.

Turing's Vision

This volume is about a vision that has steered us decisively toward the creation of machine intelligence. It was a vision of one man, the brilliant English mathematician and computer pioneer Alan M. Turing. During World War II, Turing directed a secret group that developed computing equipment powerful enough to break the code the Germans used for military communications. The English were so far ahead at this game that they had to sacrifice soldiers and civilians at times rather than tip their hand to the enemy. Turing also helped lay the theoretical groundwork for the modern concept of computing. As icing on the cake, in 1950 he published an article called "Computing Machinery and Intelligence" in which he speculated that by the year 2000, it would be possible to program a computer so that an "average interrogator will not have more than 70 percent chance" of distinguishing the computer from a person "after five minutes of questioning" (see an annotated version of his article in Chapter 3).

Given the state of computing in his day – little more than basic arithmetic and logical operations occurring over electromechanical relays connected by wires – this prediction was astounding. Engaging in disciplined extrapolation from crude apparatus and general principles, Turing not only foresaw the development of equipment and programs sophisticated enough to engage in human-like conversation, but also did reasonably well with his timeline. Early conversational programs, relying on what most AI professionals would now consider to be simplistic algorithms and trickery, could engage average people in conversation for a few minutes by the late 1960s. By the 1990s – again, some would say using trickery – programs existed that could occasionally maintain the illusion of intelligence for 15 min or so, at least when conversing on specialized topics. Programs today can do slightly better, but have we gotten past "illusion" to real intelligence, and is that even possible?

In his 1950 paper, Turing not only made predictions, he also offered a radical idea for future generations to consider: namely, that when we can no longer distinguish a computer from a person in conversation over a long period of time – that is, based simply on an exchange of pure text that excluded visual and auditory information (which he rightfully considered to be irrelevant to the central question of thinking ability) – we would have to consider the possibility that computers themselves were now "thinking".

This assertion has kept generations of philosophers, some of whom have contributed this volume, busy trying to determine the true meaning of possible outcomes in what is now called the Turing Test. Assuming that a computer can someday pass such a test – that is, pass for a human in a conversation without restrictions of time or topic – can we indeed say that it is thinking (and perhaps "intelligent" and "self-aware"), or has the trickery simply become more sophisticated?

The programming challenges have proved to be so difficult in creating such a machine that I think it is now safe to say that when a positive result is finally achieved, the entity passing the test may not be thinking the way humans do. If a pure rule-governed approach finally pays off (unlikely, as I said earlier), or if intelligence eventually arises in a machine designed to learn and self-program, the resulting entity will certainly be unlike humans in fundamental ways. If, on the other hand, success is ultimately achieved only through brute force – that is, by close emulation of human brain processes – perhaps we will have no choice but to accept intelligent machines as true thinking brethren. Then again, as I wrote in 1992 (Chapter 1), no matter how a positive outcome is achieved, the debate about the significance of the Turing Test will end the moment a skeptic finds himself or herself engaging in that debate with a computer. Upon discovering his or her dilemma, the interrogator will presumably do one of two things: refuse to continue the debate "on principle" or reluctantly agree to continue. Either way, the issue will no longer be debatable: computers will have truly achieved human-like intelligence. And perhaps that is the ultimate Turing Test: a computer passes when it can successfully engage a skeptical human in conversation about its own intelligence.

Convergence of Multiple Technologies

Although we tend to remember Turing's 1950 paper for the conversational test it proposed, the paper also speculated about other advances to come: unlimited computer memory; randomness in responding that will suggest "free will"; programs that will be self-modifying; programs that will learn in human fashion; programs that will initiate behavior, compose poetry, and "surprise" us; and programs that will have telepathic abilities equivalent to those that may exist in humans. His formidable predictive powers notwithstanding, Turing might have been amazed by some of the specific computer-related technologies that have been emerging in recent decades, and true marvels emerge when we begin to envision the inevitable convergence of such technologies. Consider just a few recent achievements:

- In the pattern-recognition area, a camera-equipped computer program developed by Javier Movellan and colleagues at the University of California, San Diego has learned to identify human faces after just six minutes of "life," and Thomas Serre and colleagues at MIT have created a computer system that can identify categories of objects (such as animals) in photographs even better than people can.
- In the language area, Morten Christiansen of Cornell University, with an international team of colleagues, has developed neural network software that simulates how children extract linguistic rules from adult conversation.
- More than 80 conversational programs (chatterbots) now operate 24 h a day online, and at least 20 of them are serious AI programming projects. Several have basic learning capabilities, and several are tied to large, growing databases of information (such as Wikipedia).
- Ted Berger and colleagues at the University of Southern California have developed electronic chips that can successfully interact with neurons in real time and that may soon be able to emulate human memory functions.
- Craig Henriquez and Miguel Nicolelis of Duke University have shown that macaque monkeys can learn to control mechanical arms and hands based on signals their brains are sending to implanted electrodes. John Donoghue and colleagues at Brown University have developed an electronic sensor which, when placed near the motor cortex area of the human brain, allows quadriplegics to open and close a prosthetic hand by thinking about those actions. Clinical trials and commercial applications are already underway.

- In 1980 Harold Cohen of the University of California, San Diego introduced a computer program that could draw, and hundreds of programs are now able to compose original music in the style of any famous composer, to produce original works of art that somtimes impress art critics, to improvise on musical instruments as well as the legendary Charlie Parker, and even to produce artistic works with their own distinctive styles.
- John Dylan Haynes of the Max Planck Institute, with colleagues at University College London and Oxford University, recently used computer-assisted brain scanning technology to predict simple human actions with 70% accuracy.
- Hod Lipson of Cornell University has recently demonstrated a robot that can make completely functional copies of itself (as long as appropriate parts are near at hand).
- Hiroshi Ishiguro of Osaka University has created androids that mimic human facial expressions and upper-body movements so closely that they have fooled people in short tests into thinking they are people.
- Alan Schultz of the Navy Center for Applied Research in Artificial Intelligence has developed animated, mobile robots that might soon be assisting astronauts and health care workers.
- Brian Scassellati and his colleagues at Yale University claim to have constructed a robot that has learned to recognize itself in a mirror – a feat sometimes said to be indicative of "self-awareness" and virtually never achieved in the animal kingdom, other than by humans, chimpanzees, and possibly elephants.
- Cynthia Breazeal and her colleagues at MIT's Artificial Intelligence Lab have created robots that can distinguish different emotions from a person's tone of voice, and Shiva Sundaram of the University of Southern California has developed programs that can successfully mimic human laughter and other forms of expressive human sound.
- Entrepreneur John Koza, who is affiliated with Stanford University, has created a self-programming network of 1,000 PCs that is able to improve human inventions – and that even earned a patent for a system it devised for making factories more efficient.
- Honda's Asimo robot, now in commercial production, can walk, run, climb stairs, recognize people's faces and voices, and perform complex tasks in response to human instructions.
- Although the DARPA-sponsored contest just 1 year before had been a disaster, in 2005 five autonomous mobile robots successfully navigated a 132-mile course in the Nevada desert without human assistance.
- As of this writing (late 2007), IBM's Blue Gene/P computer, located at the US Department of Energy's Argonne National Laboratory in Illinois, can perform more than 1,000 trillion calculations per second, just one order of magnitude short of what some believe is the processing speed of the human brain. The Japanese government has already funded the construction of a machine that should cross the human threshold (10 petaflops) by March 2011.
- In 1996, IBM's RS/6000 SP ("Deep Blue") computer came close to defeating world champion Garry Kasparov in a game of chess. On May 11, 1997, an

improved version of the machine defeated Kasparov in a six-game match – Kasparov's first professional loss. The processing speed of the winner? A paltry 200 million chess positions per second. In 2006, an enhanced version of a commercially available chess program easily defeated the current world champion, Vladimir Kramnik.
- In 2006, Klaus Schulten and colleagues at the University of Illinois, Urbana, successfully simulated the functioning of all one million atoms of a virus for 50 billionths of a second.
- "Awakened" in 2005, Blue Brain, IBM's latest variant on the Blue Gene/L system, was built specifically to model the functions of the human neocortex, the large part of brain largely responsible for higher-level thinking.
- David Dagon of the Georgia Institute of Technology estimates that 11% of the more than 650 million computers that are currently connected to the Internet are infected by botnets, stealthy programs that can work collectively and amplify the effects of other malicious software.

Self-programming? Creativity? Sophisticated pattern recognition? Brain simulation? Self-replication? Extremely fast processing? The growth and convergence of subsets of these technologies will inevitably lead to the emergence of a Massive Computational Entity, with all of the uncertainty that that entails. Meanwhile, researchers, engineers, and entrepreneurs are after comparatively smaller game: intelligent phone-answering systems and search algorithms, robot helpers and companions, and methods for repairing injured or defective human brains.

Philosophical and Methodological Issues

This volume, which has been a decade in the making, complements other recent volumes on the Turing Test. Stuart Shieber's edited volume, *The Turing Test: Verbal Behavior as the Hallmark of Intelligence* (MIT Press, 2004) includes a number of important historical papers, along with several papers of Turing's. James Moor's edited volume, *The Turing Test: The Elusive Standard of Artificial Intelligence* (Springer, 2006), covers the basics in an excellent volume for students, taking a somewhat skeptical view. And Jack Copeland's *The Essential Turing* (Oxford, 2004) brings together 17 of Turing's most provocative and interesting papers, including six on artificial intelligence.

The present volume seeks to cover a broad range of issues related to the Turing Test, focusing especially on the many new methodological issues that have challenged programmers as they have attempted to design and create intelligent conversational programs. Part I includes an introduction to the first large-scale implementation of the Turing Test as a contest – an updated version of an essay I originally published in *AI Magazine* in 1992. In the next chapter, Andrew Hodges, noted Turing historian and author of *Alan Turing: The Enigma*, provides an introduction to Turing's life and works. Chapter 3 is a unique reprinting of Turing's 1950 paper, "Computing

Machinery and Intelligence," with Talmudic-style running commentaries by Kenneth Ford, Clark Glymour, Pat Hayes, Stevan Harnad, and Ayse Pinar. This section concludes with a brief commentary on Turing's paper by John Lucas.

Part II includes seven chapters reviewing the philosophical issues that still surround Turing's 1950 proposal: Robert E. Horn has reduced the relatively vast literature on this topic to a series of diagrams and charts containing more than 800 arguments and counterarguments. Turing critic Selmer Bringsjord pretends that the Turing Test is valid, then attempts to show why it isn't. Chapters by Noam Chomsky and Paul M. Churchland, while admiring of Turing's proposal, argue that it is truly more modest than many think. In Chapter 9, Jack Copeland and Diane Proudfoot analyze a revised version of the test that Turing himself proposed in 1952, this one quite similar to the structure of the Loebner Prize Competition in Artificial Intelligence that was launched in 1991 (see Chapters 1 and 12). They also present and dismiss six criticisms of Turing's proposal.

In Chapter 10, University of California Berkeley philosopher John R. Searle criticizes both behaviorism (Turing's proposal can be considered behavioristic) and strong AI, arguing that mental states cannot properly be inferred from behavior. This section concludes with a chapter by Jean Lassègue, offering an optimistic reinterpretation of Turing's 1950 article.

Part III, which is the heart of this volume, includes 15 chapters discussing various methodological issues. First, Loebner Prize sponsor Hugh G. Loebner shares his thoughts on how to conduct a proper Turing Test, having already observed 14 such contests when he wrote this article. Several of the chapters (e.g., Chapter 13 by Richard S. Wallace, Chapter 20 by Jason L. Hutchens, and Chapter 22 by Kevin L. Copple) describe the inner workings of actual programs that have participated in various Loebner contests. In Chapter 14, Bruce Edmonds argues that for a program to pass the test, it must be embedded into conventional society for an extended period of time.

In Chapter 15, Mark Humphrys talks about an online chatterbot he created, and in the following chapter Douglas B. Lenat raises intriguing questions about how *imperfect* a program must be in order to pass the Turing Test. In Chapter 17, Chris McKinstry discusses the beginnings of an ambitious project – called "Mindpixel" – that might have given a computer program extensive knowledge through interaction with a large population of people over the Internet. Unfortunately, this project came to an abrupt halt recently with McKinstry's death.

In Chapter 18, Stuart Watt uses an innovative format to discuss the Turing Test as a platform for thinking about human thinking. In Chapter 20, Robby Garner takes issue with the design of the Loebner Prize Competition. In Chapter 20, Thomas E. Whalen describes a strategy for passing the Turing Test based on its behavioristic assumptions. In Chapter 23, Giuseppe Longo speculates about the challenges inherent in modeling continuous systems using discrete-state systems such as computers.

In Chapter 24, Michael L. Mauldin of Carnegie Mellon University – also a former entrant in the Loebner Prize Competition – discusses strategies for designing programs that might pass the test. In the following chapter, Luke Pellen talks about the challenge of creating a program that is truly intelligent, rather than one that simply responds in clever ways to keywords. This section closes with a somewhat lighthearted chapter by Eugene Demchenko and Vladimir Veselov speculating about ways to pass the Turing Test by taking advantage of the limitations and personal styles of the contest judges.

Part IV of this volume includes three rather unique contributions that remind us how much is at stake over Turing's challenge. Chapter 27, by Ray Kurzweil and Mitchell Kapor, documents in detail an actual cash wager between these two individuals, regarding whether a program will pass the test by the year 2029. Chapter 28, by noted science fiction writer Charles Platt (*The Silicon Man*), describes the "Gnirut Test", conducted by intelligent machines in the year 2030 to determine, once and for all, whether "the human brain is capable of achieving machine intelligence". The volume concludes with an article by Hugo de Garis and Sam Halioris, wondering about the dangers of creating machine-based, superhuman intellects.

Most, but not all, of the contributors to this volume believe as I do that extremely intelligent computers, with cognitive powers that far surpass our own, will appear fairly soon – probably within the next 25 years. Even if that time frame is wrong, I am certain that they will appear eventually. Either way, I hope that the Massive Computational Entities that emerge will at some point devote a few cycles of computer time to ponder the contents of this book and then, in some fashion or other, to smile.

San Diego, California Robert Epstein, Ph.D.
September 2007

Contents

About the Editors

Robert Epstein was the first director of the Loebner Prize Competition in Artificial Intelligence. He is the former Editor-in-Chief of *Psychology Today* magazine and is currently a contributing editor for *Scientific American Mind*, a visiting scholar at the University of California, San Diego, and the host of "Psyched!" on Sirius Satellite Radio. A Ph.D. of Harvard University, Epstein is the founder and Director Emeritus of the Cambridge Center for Behavioral Studies, Cambridge, Massachusetts. He has published 13 books and more than 150 articles on artificial intelligence, creativity, adolescence, and other topics in the behavioral sciences. For further information, see *http://drrobertepstein.com*.

Gary Roberts is a software engineer with Teradata Corporation and an adjunct faculty member at National University in San Diego, California. He holds a Ph.D. from the Department of Artificial Intelligence at the University of Edinburgh, an M.S. in Computer Science from California State University, Northridge, and a B.A. in Mathematics from the University of California, Los Angeles. His research interests include computational linguistics, genetic algorithms, robotics, parallel operating and database systems, and software globalization.

Grace Beber is an editor and proposal writer with Gartner, Inc. Before working with Gartner, Grace taught English while serving in the US Peace Corps in Poland, and also wrote grant proposals and lectured on human rights at Jagiellonian University in Krakow, Poland. Prior to that, she worked as a project manager and technical writer at Crédit Lyonnais in Jakarta, Indonesia. Ms. Beber earned an M.A. in Organizational Development from Alliant International University and a B.S. in Psychology from Santa Clara University.

Part I
Setting the Stage

Chapter 1
The Quest for the Thinking Computer

Robert Epstein

Abstract The first large-scale implementation of the Turing Test was set in motion in 1985, with the first contest taking place in 1991. US$100,000 in prize money was offered to the developers of a computer program that could fool people into thinking it was a person. The initial contest, which allowed programs to focus on a specific topic, was planned and designed by a committee of distinguished philosophers and computer scientists and drew worldwide attention. The results of the contest showed that although conversational computer programs are still quite primitive, distinguishing a person from a computer when only brief conversations are permitted can be challenging. When the contest judges ranked the eight computer terminals in the event from most to least human, no computer program was ranked as human as any of the humans in the contest; however, the highest-ranked computer program was misclassified as a human by five of the ten judges, and two other programs were also sometimes misclassified. Also of note, one human was mistakenly identified as a computer by three of the ten judges.

Keywords Hugh G. Loebner, Loebner Prize Competition in Artificial Intelligence, restricted Turing Test

1.1 Planning

In 1985 an old friend, Hugh Loebner, told me excitedly that the Turing Test should be made into an annual contest. We were ambling down a Manhattan street on our way to dinner, as I recall. Hugh was always full of ideas and always animated, but this idea seemed so important that I began to press him for details, and, ultimately, for money. Four years later, while serving as the executive director of the Cambridge Center for Behavioral Studies, an advanced studies institute in Massachusetts, I established the Loebner Prize Competition, the first serious effort to locate a machine that could pass the Turing Test. Hugh had come through with a pledge

University of California, San Diego, USA

of US$100,000 for the prize money, along with some additional funds from his company, Crown Industries, to help with expenses. The quest for the thinking computer had begun. In this article, I will summarize some of the difficult issues that were debated in nearly 2 years of planning that preceded the first real-time competition. I will then describe that first event, which took place on November 8, 1991, at The Computer Museum in Boston and offer a summary of some of the data generated by that event. Finally, I will speculate about the future of the competition – now an annual event, as Hugh envisioned – and about its significance to the AI community.

Planning for the event was supervised by a special committee, first chaired by I. Bernard Cohen, an eminent historian of science who had long been interested in the history of computing machines. Other members included myself, Daniel C. Dennett of Tufts University, Harry R. Lewis of the Aiken Computation Laboratory at Harvard, H. M. Parsons of HumRRO, W. V. Quine of Harvard, and Joseph Weizenbaum of MIT. Allen Newell of Carnegie-Mellon served as an advisor, as did Hugh Loebner. After the first year of meetings, which began in January of 1990, Daniel Dennett succeeded I. Bernard Cohen as chair. The committee met every month or two for 2 or 3 h at a time, and subcommittees studied certain issues in between committee meetings. I think it is safe to say that none of us knew what we were getting into. The intricacies of setting up a real Turing Test that would ultimately yield a legitimate winner were enormous. Small points were occasionally debated for months without clear resolution.

In his original 1950 proposal the English mathematician Alan M. Turing proposed a variation on a simple parlor game as a means for identifying a machine that can think: A human judge interacts with two computer terminals, one controlled by a computer and the other by a person, but the judge does not know which is which. If, after a prolonged conversation at each terminal, the judge cannot tell the difference, we would have to say, asserted Turing, that in some sense the computer is thinking. Computers barely existed in Turing's day, but, somehow, he saw the future with uncanny clarity: By the end of the century, he said, an "average interrogator" could be fooled most of the time for 5 min or so.

After much debate, the Loebner Prize Committee ultimately rejected Turing's simple two-terminal design in favor of one that is more discriminating and less problematic. The two-terminal design is troublesome for several reasons, among them: The design presumes that the hidden human – the human "confederate", to use the language of the social sciences – is evenly matched to the computer. Matching becomes especially critical if several computers are competing. Each must be paired with a comparable human so that the computers can ultimately be compared fairly to each other. We eventually concluded that we could not guarantee a fair contest if we were faced with such a requirement. No amount of pretesting of machines and confederates could assure adequate matching. The two-terminal design also makes it difficult to rank computer entrants. After all, they were only competing against their respective confederates, not against each other.

We developed a multiterminal design to eliminate these problems: approximately ten judges are faced with an equal number of terminals. The judges are told that at least two of the terminals are controlled by computers and at least two by people. Again, the judges do not know which terminal is which. Each judge spends about 15 min at each terminal and then scores the terminals according to how human-like each exchange seemed to be. Positions are switched in a pseudorandom sequence. Thus, the terminals are compared to each other and to the confederates, all in one simple design.

Other advantages of this design became evident when we began to grapple with scoring issues. We spent months researching, exploring, and rejecting various rating and confidence measures commonly used in the social sciences. I programmed several of them and ran simulations of contest outcomes. The results were disappointing for reasons we could not have anticipated. Turing's brilliant paper had not gone far enough to yield practical procedures. In fact, we realized only slowly that his paper had not even specified an outcome that could be interpreted meaningfully. A binary decision by a single judge would hardly be adequate for awarding a large cash prize – and, in effect, for declaring the existence of a significant new breed of intelligent entities. Would some proportion of ten binary decisions be enough? How about 100 decisions? What, in fact, would it take to say that a computer's performance was indistinguishable from a person's?

A conceptual breakthrough came only after we hit upon a simple scoring method. (R. Duncan Luce, a mathematical psychologist at the University of California, Irvine, was especially helpful at this juncture.) The point is worth emphasizing: The scoring method came first, and some clear thinking followed. The method was simply to have each judge rank the terminals according to how human-like the exchanges were. The computer with the highest median rank wins that year's prize; thus, we are guaranteed a winner each year. We also ask the judges to draw a line between terminals he or she judged to be controlled by humans and those he or she judged to be controlled by computers; thus, we have a simple record of errors made by individual judges. This record does not affect the scoring, but it is well worth preserving. Finally, if the median rank of the winning computer equals or exceeds the median rank of a human confederate, *that computer will have passed (a modern variant of) the Turing Test*. It is worth quoting part of a memo I wrote to the committee in May of 1991 regarding this simple approach to scoring:

Advantages of this method

1. It is simple. The press will understand it.
2. It yields a winning computer entrant.
3. It provides a simple, reasonable criterion for passing the Turing Test: When the [median] rank of a computer system equals or exceeds the [median] rank of a human confederate, the computer has passed.
4. It preserves binary judgment errors on the part of individual judges. It will reveal when a judge misclassifies a computer as a human.
5. It avoids computational problems that binary judgments alone might create. A misclassified computer would create missing data, for example.
6. It avoids theoretical and practical problems associated with rating scales.

Other issues were also challenging. We were obsessed for months with what we called “the buffering problem”. Should we allow entrants to simulate human typing foibles? Some of us – most notably, Joseph Weizenbaum – said that such simulations were trivial and irrelevant, but we ultimately agreed to leave this up to the programmers, at least for the first contest. One could send messages in a burst (“burst mode”) or character-by-character (“chat mode”), complete with misspellings, destructive backspaces, and so on. This meant that we had to have at least one of our confederates communicating in burst mode and at least one in chat mode. Allowing this variability might teach us something, we speculated.

We knew that an open-ended test – one in which judges could type anything about any topic – would be a disaster. Language processing is still crude, and, even if it were not, the “knowledge explosion” problem would mean certain defeat for any computer within a very short time. There is simply too much to know, and computers know very little. We settled, painfully, on a restricted test: Next to each terminal, a topic would be posted and the entrants and confederates would have to communicate on that one topic only. Judges would be instructed to restrict their communications to that one topic, and programmers would be advised to protect their programs from off-topic questions or comments. Entrants could pick their own topics, and the committee would work with confederates to choose the confederates’ topics. Moreover, we eventually realized that the topics would have to be “ordinary.” Expert systems – those specializing in moon rocks or the cardiovascular system, for example – would be too easy to identify as computers. In an attempt to keep both the confederates and judges honest and on-task, we also decided to recruit referees to monitor both the confederates and the judges throughout the contest.

This sounds simple enough, but we knew we would have trouble with the topic restriction, and we were still debating the matter the evening before the first contest. If the posted topic is “clothing”, for example, could the judge ask, “What type of clothing does Michael Jordan wear?” Is that fair, or is that a sneaky way to see if the terminal can talk about basketball (in which case it is probably controlled by a human)?

Should we allow the judges to be aggressive? Should graduate students in computer science be allowed to serve? Again, many stimulating and frustrating debates took place. Both to be true to the spirit of Turing’s proposal and to assure some interesting and nontrivial exchanges, we decided that we would select a diverse group of bright judges who had little or no knowledge of AI or computer science. We attracted candidates through newspaper ads that said little other than that one had to have typing skills.

In short – and I am only scratching the surface here – we took great pains to protect the computers. We felt that in the early years of the contest, such protection would be essential. Allen Newell was especially insistent on this point. Computers are just too inept at this point to fool anyone for very long. At least that was our thinking. Perhaps every fifth year or so, we said, we would hold an open-ended test – one with no topic restriction. Most of us felt that the computers would be trounced in such a test – perhaps for decades to come.

We agreed that the winner of a restricted test would receive a small cash award and bronze medal and that the cash award would be increased each year. If, during an unrestricted test, a computer entrant matched or equaled the median score of a human, the full cash prize would be awarded, and the contest would be abolished.

Other issues, too numerous to explore here, were also discussed: How could we assure honesty among the entrants? After all, we were dealing with a profession known widely for its pranks. Should the confederates pretend to be computers or simply communicate naturally? We opted for the latter, consistent with Turing. Should we employ children as confederates in the early years? Should professional typists do the judges' typing? How aggressive should the referees be in limiting replies? Should entrants be required to show us their code or even to make it public? We said no; we did not want to discourage submissions of programs with possible commercial value.

Our final design was closely analogous to the classic double-blind procedure used in experimental research: The prize committee members were the "investigators". We knew which terminal was which, and we selected the judges, confederates, and referees. The referees were analogous to "experimenters". They handled the judges and confederates during the contest. They were experts in computer science or related fields, but they did not know which terminal was which. The judges were analogous to "subjects". They did not know which terminal was which, and they were being handled by people with the same lack of knowledge.

Over time, formal rules were developed expressing these ideas. Announcements were made to the press, and funding for the first contest was secured from the Sloan Foundation and the National Science Foundation. Technical details for running the show were coordinated with The Computer Museum in Boston, which agreed to host the contest. Applications were screened in the summer of 1991, and six finalists were selected by the prize committee in September. Confederates, judges, and referees were selected in October.

1.2 The 1991 Competition

The first contest fulfilled yet another desire of the prize committee. It was great fun. It was an extravaganza. A live audience of 200 laughed and cheered and conjectured while they watched eight conversations unfold in real time on large screens. A moderator – A. K. Dewdney of *Scientific American* – roamed the auditorium with a cordless microphone, interviewing and commenting. Four film crews and dozens of reporters and photographers bristled and flashed and shouldered each other to get the best angles. Food flowed all day.

The judges and terminals were set up in a roped-off area outside the auditorium. You could view them directly behind the ropes if the journalists would let you by – or on a large screen setup in the auditorium. Groups of about 20 chairs were positioned around each screen in the auditorium, and the moderator encouraged the members of the audience to move to another screen whenever the judges switched terminals – about once every 13 min. The confederates were stationed in a remote area of the museum behind a guarded door. Dennett and others made some opening remarks midday, and the real-time competition itself took about 3 h in the afternoon.

Some technical problems got things off to a frustrating start. Two of the ten terminals would not function properly, knocking out two of the four confederates we had planned to use. We went forward with the six entries and two confederates (recall that everyone except the members of the prize committee and some technicians knew simply that at least two of the eight working terminals were controlled by computers and that at least two were controlled by people).

Our ten judges were diverse, bright, and without expertise in computer science, just as we had planned. Seven were female, and three were male. They ranged in age from 19 to 48; six were in their 20 s, one was in her 30 s, and two were in their 40 s. All were proficient in English, yet one was born in the Philippines, and three had spent most of their lives in foreign countries: Ethiopia, Australia, and West Africa, respectively. Seven had B.A.'s, two of the seven had masters degrees, and two were currently in graduate school. One was high school educated, and two were current college students. Four claimed to have some background in psychology or philosophy. Those not currently in school had a variety of professions: auto mechanic, graphic designer, hypnotist, teacher, and so on.

All communications were by modem. Entrants communicated from PCs or mainframes located throughout the country, and even the confederates' communications were looped through phones outside the building. The judges and audience were told this so that transmission errors would be discounted as clues about the identity of the terminals.

As we expected, the computers performed relatively poorly for the most part. Consider the following exchange, for example, between Judge 2 and Terminal 10 (the topic was "Dry Martinis"):

01:36:42 Judge 2: Of what is gin made?
01:36:52 Terminal 10: I'M NOT SURE.
01:36:58 Judge 2: OK Tell me any thin else you know of about martinis.
01:37:21 Terminal 10: I REALLY DON'T KNOW.
01:37:28 Judge 2: What foods do martinis go with.
01:37:44 Terminal 10: COULD YOU ASK THE QUESTION A DIFFERENT WAY?
01:37:52 Judge 2: Why?
01:37:58 Terminal 10: I REALLY CAN'T SAY. OH WELL.
01:38:04 Judge 2: Ok with what foods shoul onew drink a martini.
01:38:28 I REALLY CAN'T SAY. OH WELL.[1]

Five consecutive replies were unresponsive to the judge's questions; the last two replies were repeated verbatim, a dead giveaway. Frequent typing errors by the judge may have had some effect. Unfortunately, this pattern was common. Consider this exchange (the topic was "Second Grade Topics"):

[1] Copyright, 1991, Cambridge Center for Behavioral Studies. Reprinted by permission. Note that the quotes from the 1991 Loebner Competition transcripts are included here character-for-character as they occurred during the contest.

02:53:22	Judge 4: What do you study at school?
02:53:43	Terminal 3: I don't know.
02:53:51	Judge 4: What do you like most about school?
02:54:16	Terminal 3: As I said – I don't know.
02:54:25	Judge 4: What topic do you know the most about?
02:54:42	Terminal 3: I don't know.
02:54:51	Judge 4: What kind of gymnastics do you do?
02:55:10	Terminal 4: As I said – I don't know.
02:55:16	Judge 4: I thought you were a gymnast.
02:55:27	Terminal 3: Learning is fun.[2]

The winning entry, "Whimsical Conversation", submitted by an independent programmer, Joseph Weintraub of New York City, performed substantially better than the others. It seemed, fairly often, to make contact with the judge's comments and queries. Moreover, it simulated the rhythms and corrections of human typing nearly perfectly. Consider the following exchange:

01:32:25	Judge 6: Hello, earth person!
01:33:03	Terminal 5: Hello there judge, are you ready to have some fun? How ya doin, judge? Hello – how's the weather up there today?
01:33:35	Judge 6: Dark clouds with occasional bursts of sunshine (or rather camera flashes).
01:34:45	Terminal 5: The trouble with the rat race is that even if you win you're still a rat. Whimsy can be flimsy, right? How important is that to you?
01:35:26	Judge 6: I can't say I understand the question…but the whimsy of a poet is one thing and the whimsy of most of us is another.
01:36:51	Terminal 5: How do you know you can t say you understand the question…but the whimsy of a poet is one thing and the whimsy of most of us is another.
01:37:37	Judge 6: Are you repeating me just for laughs?
01:38:04	Terminal 5: I'm sorry, I do sometimes repeat myself…perhaps it's because i can't find my electronic organizer![3]

In this exchange, the program reflects back one response wholesale, just as Weizenbaum's ELIZA did decades ago. But it recovers fairly impressively from the blunder with an apology, simulating typing foibles all the while.

The overall result was fairly predictable, but there were also some surprises. The two human confederates, whose topics were, respectively, "Women's Clothing" and "Shakespeare's Plays", scored in first and second place, respectively. Not far behind was Weintraub's whimsical program; thus, Weintraub won the 1991 contest. He was awarded a bronze medal (bearing a likeness of Turing on the front and Loebner on the back) and a check for US$1,500. Weintraub, who tended his equipment in New York

---

[2] Copyright, 1991, Cambridge Center for Behavioral Studies. Reprinted by permission.

[3] Copyright, 1991, Cambridge Center for Behavioral Studies. Reprinted by permission.

during the contest, took questions from the press by speakerphone after the announcement was made. An entry from Peter Colby and Kenneth Colby, of PARRY fame, scored just below Weintraub's. In third place among the computers was a submission from Michael Mauldin of the Center for Machine Translation at Carnegie-Mellon.

The surprises were notable. First, five of the ten judges (Judges 2, 3, 4, 9, and 10) mistook Weintraub's program for a human. Judge 3 rated it above one human confederate (Terminal 1), and Judge 10 rated it above both human confederates (Terminal 1 and 4). The Colbys' program ("Problems in Romantic Relationships") was mistaken for a person by two of the judges, and another program ("Second Grade School Topics") was misclassified by one judge. Perhaps even more remarkable, Cynthia Clay, the human confederate whose topic was Shakespeare's plays, was mistaken for a computer by three judges. Judge 10 placed her lower in rank than a computer (Terminal 5), and two judges (1 and 5) placed her lower in rank than two computers. Note that Cynthia's responses were buffered; that is, her responses occurred in a burst, suggesting computer output. Furthermore, she was an expert on Shakespeare, and she often quoted lengthy passages verbatim. Several judges remarked that her replies seemed too expert to be human.

As Turing anticipated, the contest tells us as much, or perhaps even more, about our failings as judges as it does about the failings of computers. People's preconceptions about the limits of computers – and of people – strongly biases their judgments.

At the start of the contest, members of the audience were given forms to help them do their own judging. The forms asked for basic demographic information, as well. Seventy-seven forms were collected at the end of the contest. Based on this sample, audience ratings may be summarized as follows:

- Audience rankings matched those of the judges, and the rankings of those who claimed expertise in computer science did not differ substantially from the rankings of those who did not claim such expertise. For the 66 respondents who ranked all eight terminals, Terminals 1 and 4 were again ranked first and second, respectively, and Terminal 5 ("Whimsical Conversation") was again ranked third. Curiously, the other five terminals were ranked equally; that is, they were, on average, indiscriminable.
- Unlike the judges, members of the audience rarely misclassified the terminals, perhaps because members of the audience could communicate with each other; judges could not. For example, the winning computer, "Whimsical Conversation", was labeled a human by only five out of the 77 respondents (ten did not reply, leaving 61 correct classifications), and Cynthia Clay (Terminal 4) was misclassified as a computer by only five respondents (seven did not reply, leaving 65 correct classifications). The other human confederate, although ranked higher by both judges and audience, was misclassified at nearly the same rate. Once again, expertise in computer science had so systematic effect.

With James C. Pabelico, a student at the University of California, San Diego, I attempted a search for objective factors that could predict the judges' ratings – in other words, that measured the apparent intelligence of an entity communicating over a computer terminal. Simplistic factors such as word length, sentence length, number of syllables per word, and number of prepositions were not predictive. Neither were various measures of readability, such as Flesch Reading Ease, Gunning's Fog Index, and Flesch-Kincaid Grade Level. The Weintraub and Colby programs, for example, had Flesch-Kincaid Grade Levels of 2 and 6, respectively; the two humans had scores of 3 and 4.

So why did Weintraub's program win? And how did it fool half the judges into thinking it was a person? Unfortunately, it may have won for the wrong reasons. It was the only program, first of all, that simulated human typing foibles well. Another program simulated human typing so poorly that it was instantly recognizable as a computer on that basis alone; no human could possibly have typed the way it was typing.

Perhaps more notable, Weintraub's program simulated a very curious kind of person: the jester. We allow great latitude when conversing with jesters; incomprehensible, irrelevant responses are to be expected. We are equally tolerant of young children, developmentally disabled individuals, psychotic patients, head-injured individuals, and absentminded professors. Weintraub's program may have succeeded simply because his terminal was labeled "whimsical conversation". The prize committee discussed this possibility, and considerable concern was expressed. In 1992, the committee favored programs that had clear subject matters.

1.3 Speculations

I believe that when a computer passes an unrestricted Turing Test, humankind will be changed forever. From that day on, computers will be companions to the human race – and extraordinary companions indeed. For starters, they will be efficient, fast, natural-language interfaces to virtually all knowledge. They will be able to access and evaluate enormous amounts of data on an ongoing basis and to discuss the results with us in terms we can understand. They will think efficiently 24 h a day, and they will have more patience than any saint.

Thinking computers will also have new roles to play in real-time control. Everything from vacuum cleaners to power plants has a dumb computer in it these days; some day, smart computers will share in the decision-making. Over networks or even airwaves, thinking computers will be able to coordinate events worldwide in a way humans never could.

Thinking computers will be a new race, a sentient companion to our own. When a computer finally passes the Turing Test, will we have the right to turn it off? Who should get the prize money – the programmer or the computer? Can we say that such a machine is "self-aware"? Should we give it the right to vote? Should it pay taxes? If you doubt the significance of these issues, consider the possibility that someday soon *you will have to argue them with a computer*. If you refuse to talk to the machine, you will be like the judges in *Planet of the Apes* who refused to allow the talking human to speak in court because, according to the religious dogma of the planet, humans were incapable of speech.

The Internet has added another dimension to the Turing Test. It is only natural that we think of the Internet as a tool that serves humanity, but someday sentient computers will undoubtedly see it as their natural home. It is not inconceivable that within milliseconds of achieving sentience, that first remarkable entity will dive into the Internet to learn, to grow, and to assure its own survival. Once in the Net, it will be impossible to disable, and its subsequent effect on the human race is anyone's guess. Internet II – the UltraNet now functioning at some major institutions – will provide an even larger nest.

Some people, including members of the original Loebner Prize Committee, believe that computers will never cross this threshold. But 40 years of reading science fiction novels, 35 years of programming, and nearly 30 years of studying psychology has me convinced that the sentient computer is inevitable. *We're* sentient computers, after all, and those who are skeptical about technological advances are usually left in the dust.

Loebner himself was open-minded when the contest was set in motion, perhaps even skeptical. But he also offered the most outrageous prediction of all. Some day, he said, when the human race is long dead, a mechanical race will remember us as deities. After all, we are the creators, are we not?

I think the quest for the thinking computer will eventually become as intense as the quest for the Holy Grail. The stakes are similar. A program that passes the Turing Test will be worth a fortune. Just ask it.

Even within the first year, committee members talked about expanding the contest at some point to include Turing-like tests of robotics, pattern recognition, and speech recognition and synthesis. In a week-long tournament, computers would compete against people in each domain. The ultimate outcome? Well, have you seen or read *I, Robot*?

I may be overly optimistic about the future of artificial intelligence. Certainly, several of my colleagues, much older and, by definition, much wiser than I, have told me so. But we will all have fun exploring the possibilities – even if, someday, and for reasons I cannot now imagine, we are forced to conclude that the Turing Test cannot be passed.

Afterword This is a slightly modified version of an article first published in 1992 in AI Magazine. The Loebner Prize Committee continued to direct the event for the first four contests. During the planning for the fifth contest, Hugh Loebner asked the committee members to change the rules in ways they found objectionable, and the committee disbanded. The contest has been held every year since, however, under Loebner's direction. The 16th Annual Loebner Prize Competition was held on September 17, 2006, at University College London. Four computer programs participated, and the winner was a program named "Joan," created by Rollo Carpenter of Icogno Ltd.

Reference

Turing, A. M., 1950, Computing machinery and intelligence, *Mind* **50**(236): 433–460.

Chapter 2
Alan Turing and the Turing Test

Andrew Hodges

Abstract The study of Alan Turing's life and work shows how the origin of the Turing Test lies in Turing's formulation of the concept of computability and the question of its limits.

Keywords Alan Turing, Turing machine, computability

2.1 Introduction

Alan Mathison Turing
Born: 23 June 1912, Paddington, London
Died: 7 June 1954, Wilmslow, Cheshire

The short and extraordinary life of the British mathematician Alan Turing embraces the foundations of mathematics, the origin of the computer, the secret cryptological war against Nazi Germany, and much more. For the modern public, his name is perhaps most strongly associated with yet another context: the philosophy of Mind and Artificial Intelligence (AI). Specifically, he is immortalised in the "Turing Test" for intelligence, which Turing himself called "the imitation game".

The famous test appeared in Turing's paper, "Computing machinery and intelligence", published in October 1950 in the philosophical journal *Mind* (Turing, 1950). Turing was then employed at Manchester University, where the world's first stored-program computer had been working since June 1948. Turing had not been appointed to produce philosophical papers; his primary function was to create and manage the first software for the computer. However, he had also continued his research in mathematics, and had been drawn into discussion with the scientific philosopher Michael Polanyi, who encouraged him to publish his views on what Turing called "intelligent machinery" (Turing, 1948). The point of this 1950 paper,

Wadham College, University of Oxford

R. Epstein et al. (eds.), *Parsing the Turing Test.*

of which the imitation game was only a part, was to argue that intelligence of a human level could be evinced by a suitably programmed computer. The imitation game lent definiteness to the idea of being as intelligent as a human being.

Turing's 1950 paper did not arise in isolation, and the purpose of this biographical sketch is to set Turing's test in the context of his life and work. The 1950 paper was an important summary of his views for the general philosophical public, but he had been developing those views for many years. It would also be a mistake to think of Turing as a mathematician making a detached comment on the potential of computers. He was very fully engaged in the development of modern computer science, both in theory and in practice.

2.2 The Turing Machine

Indeed the 1950 paper had itself an important autobiographical element, although Turing did not emphasise its personal aspect. Much of the early part of the paper involved an exposition of the concept of computability. The definition of computability was Turing's own great achievement of the pre-war period, when he was a young Fellow of King's College, Cambridge University. In 1936, when he was only 23, he announced the definition of what soon became known as the *Turing machine*, thus giving a precise and convincing mathematical definition of an "effective method". In Turing's paper "On computable numbers, with an application to the Entscheidungsproblem" (Turing, 1936), he gave a discussion of his definition, arguing convincingly that it encompassed the most general possible process for computing a number.

By so doing, he satisfactorily answered Hilbert's decision problem for the provability of mathematical theorems. The paper did more: it also defined the concept of a *universal* Turing machine and hence the principle of the modern stored-program computer. The universal machine was another important element of the theory Turing needed to explain in his 1950 paper: the point is that all Turing machines can be thought of as programs for a single universal machine. Most striking, perhaps, is that the Turing machine, formulated 14 years before the "Turing Test", was also based on a principle of imitation. The "machine" was modelled by considering what a human being could do when following a definite method. According to Turing's argument, it encompassed everything that could be done by a human calculator with finite equipment, but allowed an unlimited supply of paper to write on (formalised into a paper tape marked off in squares) and unlimited time.

The basis in human calculation was emphasised in Turing's arguments. The "squares" of the Turing machine "tape", for instance, originated in Turing's explanation as the squares of a child's exercise book. The atomic operations of scanning, marking, erasing, and moving to left and right were likewise related to human actions. Most importantly, the finitely many "configurations" of a Turing machine were related to the finite number of states of mind, or finite memory, of a human calculator. This very bold appeal to modelling "states of mind" by states of a machine seems already to anticipate the thesis of machine intelligence in 1950.

Should we then say that Turing in 1950 was only restating the implications of what he had claimed in 1936?

A simple reading of the story would support this view. It might be argued that from 1936 onwards, Turing steadfastly sought and found ways to implement his theory in practice. In 1937 Turing began a practical interest in electromagnetic relays to implement logical operations, an interest very different from anything expected of a Cambridge pure mathematician (Hodges, 1983). After 1939, this interest turned into one of immense practical importance. Turing's ingenious logic was translated into working electromagnetic machinery at Bletchley Park, Buckinghamshire, the centre of British code-breaking operations. His algorithm for breaking the Enigma-ciphered German messages, as embodied in the British "Bombe", lay at the centre of its success. He personally headed the work of deciphering German naval messages, and led the development of methods of astonishing efficiency. He was introduced to American work at the highest level, and to the most advanced electronic technology. In this way he learned that electronic storage and electronic circuits could make an effective and fast practical version of the "paper tape" and configurations needed for a universal machine. He learned electronics for himself, again a highly unconventional step. Turing emerged from the war in 1945 full of enthusiasm for engineering a practical version of the universal machine – in modern parlance a stored-program computer. By a remarkable sequence of events, Turing was taken on by the National Physical Laboratory, London, with exactly this commission. His plan was submitted in March 1946 (Turing, 1946). As well as pioneering key ideas in computer hardware and software design, it mentioned the idea of the machine showing "intelligence", with chess-playing as a paradigm. This germ then rapidly developed into the program set out in the 1950 paper.

2.3 Intelligence and Intuition

Although basically correct, a subtle adjustment to this basic story is required, and it is one that casts light on the structure and content of the 1950 paper. A reading of that paper will show that Turing was highly aware of the natural objection that machines cannot do those things which are by nature non-mechanical: those human actions that require initiative, imagination, judgment, cause surprise, are liable to errors, and so forth. Much of his argument was directed to the claim that machines would, in fact, be capable of all these apparently non-mechanical things.

But there was no reflection of this claim in his 1936 work, "On computable numbers" (Turing, 1936). The "states of mind" that Turing considered were only those employed when the mind is occupied on performing a definite method or process. There was no reference to imagination or initiative in the "states of mind".

So we can ask: at what point in his biography did Turing adopt the idea that computable operations would encompass everything normally called "thinking" rather than only "definite methods"? Or equivalently, when did he first consider that operations, which are in fact the workings of predictable Turing machines, could nevertheless appear to have the characteristics of genuine intelligence?

Turing wrote very little about his own intellectual development, and his writings do not give a direct answer to this question. However, there are two important stages in his work not mentioned in the above account, which when considered in their context suggest a plausible answer: namely at some point after 1938 and probably in about 1941.

During the 2 years Turing spent at Princeton, from 1936 to 1938, he was investigating the logic of the *uncomputable*. Turing's exposition (Turing, 1939) described the "formulae, seen intuitively to be correct, but which the Gödel theorem shows are unprovable in the original system". In this pre-war period, Turing left open the possibility that the mind had an "intuitive" power of performing uncomputable steps beyond the scope of the Turing machine (a thesis, incidentally, that was always held by Gödel himself).

But in the period around 1941 when the immediate crisis of the Enigma problem was resolved, Turing began to discuss with colleagues at Bletchley Park the possibility of machines playing chess (Hodges, 1983). Chess-playing was in fact a civilian analogue of what they were doing in their secret military work, in which mechanical methods of great sophistication were outdoing some aspects of human intuition. It seems, taking the view as expressed in Hodges (1997, 2002), that Turing probably decided in about 1941 that the scope of computable operations was in fact sufficient to account for those mental operations apparently "non-mechanical" by the standards of ordinary language, and even the apparently uncomputable operations of truth recognition.

There was possibly another wartime influence on his development: Turing's general exposure to modern ideas such as the neural physiology of the brain and the behaviourist model of the mind. McCulloch and Pitts (1943) related their logical model of neurons to Turing's computability; Turing returned the compliment by referring to their work. Turing was developing the picture of the brain as *a finite discrete state machine*. In a sense this was only a small step from the "finitely many states of mind" of 1936. But it went further because Turing's postwar idea was that *all* mental functions of the brain could be accommodated in this model, and not just those of a mind following a definite rule. As we shall see, Turing framed an argument to explain how this mechanical picture of the brain could be reconciled with the counter-arguments from Gödel's theorem.

Very possibly it was this new conviction that made him so enthusiastic, in the closing stages of the war, about the prospect of building an electronic version of the universal machine. He was not highly motivated by building a computer to work out programmed mathematical operations. His interest was more in the nature of the mind. Informally, when speaking of his computer plans in 1945, he called them plans for "building a brain" (Hodges, 1983).

2.4 Intelligent Machinery

With this in mind, we can examine in more detail his very first written mention of "intelligent" machinery (Turing, 1946). One should first note how remarkable it was that Turing should put a speculative claim about intelligence in a purely technical, practical report. However, this was entirely typical of his *modus operandi*. One should next appreciate that Turing always relished the paradox, even apparent contradiction in terms, involved in speaking of "intelligent" machinery. First he explained how the computer could be programmed to calculate chess moves. He continued, underlining the paradox:

> This … raises the question 'Can a machine play chess?' It could fairly easily be made to play a rather bad game. It would be bad because chess requires intelligence. We stated … that the machine should be treated as entirely without intelligence. There are indications however that it is possible to make the machine display intelligence at the risk of its making occasional serious mistakes. By following up this aspect the machine could probably be made to play very good chess.

This mysterious reference to "mistakes", which could have made no sense to anyone reading this report, was explained in a talk of February 1947 (Turing, 1947). Here the idea of "mistake-making" appeared in the context of the objection to the prospect of machine intelligence posed by Gödel's theorem. This objection (which appears as "The Mathematical Objection" in the 1950 paper) is that no Turing machine (i.e., computer program) can do as well as a human being. The human being can see the truth of mathematical assertions which cannot be proved by following rules based on formal axioms.

Turing's post-war argument (the point of view he probably arrived at in about 1941) is, however, that human beings do *not* reliably see the truth of such statements. Mathematicians, their brains being discrete state machines, can only employ an algorithm. Gödel's theorem tells us that no algorithm can coincide with truth-seeing in every case, and so the algorithm is bound to fail sometimes. But if it is accepted that the mathematician is not infallible, and will sometimes fail, it follows that machines – also implementing algorithms, and therefore also making mistakes – may do equally well. To illustrate the theme of doing equally well, Turing appealed to the concept of "fair play for machines". This concept was essentially the idea of the imitation game. The 1950 scenario merely added dramatic detail. Thus, the imitation game had its origins in the wartime debate in Turing's own mind about how to reconcile Gödel's theorem and the apparently non-mechanical actions of human minds with his discrete state machine model of the brain.

After 1947, Turing continued to a wider and more constructive discussion of how machines might perform apparently non-mechanical tasks: how completely unintelligent micro-operations might add up to intelligent processes. This investigation was presented in an internal report: "Intelligent Machinery", for the National Physical Laboratory (Turing, 1948). It was not published until 1968, but was in many ways the basis of his better-known and less technical 1950 exposition. One interesting feature of this 1948 report is its evidence of a wartime inspiration for his new ideas. Turing referred to images of the writer Dorothy Sayers, to illustrate the commonly accepted meaning of "mechanical" behaviour. The book he quoted was one he was reading at Bletchley Park in 1941. Turing also tellingly described 1940 as the date after which machines were no longer restricted to "extremely straightforward, possibly even to repetitious, jobs". He must have had his own Enigma-breaking Bombe, and other highly sophisticated code-breaking operations, in mind.

In this report, Turing characterised intelligence as requiring “discipline”, which he identified with the programmability of a universal machine, plus a residue of “initiative”. Initiative now played the role that “intuition” had done in 1938: mental actions apparently going beyond the scope of a “definite method”. How was initiative to be found within the scope of computable operations, and so implemented on a computer?

Turing suggested various possibilities all based on *imitating* human brains: learning, teaching, training, and searching. From the outset of his design work in 1945, Turing had been enthusiastic for exploiting the feature of a stored-program computer that it allows for a program thus stored to be manipulated in the same way as data. These ideas took his enthusiasm further, by having the machine actively modify its own programs, to arrive at functions which had never been planned or envisaged by a programmer. Turing emphasised that at a more fundamental level the concept of “a machine changing its own instructions” was “really a nonsensical form of phraseology”, but it was convenient. The upshot of his argument was that by one means or another, and probably using many methods in combination, a computer could be made to simulate the mental functions of human brains.

From a purely biographical point of view, it is remarkable that someone so original, and whose individual qualities had generally been stoutly resisted by his social environment, should arrive at the conclusion that creativity is a computable process, something that could be achieved by a computer. But it was where he was led by his guiding idea of the brain as a finite machine, whose operations must be computable however different they appeared from what people had hitherto thought of as “mechanical” in nature.

2.5 The Imitation of Mind

This 1948 work was the background to the 1950 paper, in which Turing made a more public claim than ever before that intelligence could be evinced by computing machinery: i.e., belonged to the realm of computable processes. It was also a more ambitious claim than ever, since by provocative forays into the world of the Arts with witty talk of Shakespeare and sonnets, Turing made it quite clear that he was not restricting “intelligence” to some special science-minded arena. The famous test, pitting human against machine in demonstrating intelligence, embodied the “fair play” announced in 1947. The setting of the test, however, with its remote text-based link, did have a further functional significance. It was supposed to give a way of distinguishing those things Turing considered relevant to intelligence, as opposed to other human faculties involving their many senses and physical actions. It is probably in drawing this distinction that Turing showed the least certainty, and this aspect of his paper has attracted the most criticism.

Returning, however, to Turing’s central idea, it should be emphasised that Turing never imagined that the structure of the brain would resemble that of a computer, with a central control unit processing instructions serially. The crucial point here lies in Turing's exposition of the universal machine concept (Turing, 1939). It follows from his argument that provided the operation of "thought" is equivalent to the operation of *some* discrete state machine, it can be simulated by a program run on a single, universal machine, i.e., a computer. Thus Turing threw all his emphasis on the development of what would now be called software, rather than on the engineering of a brain-like object.

This point can be further refined. Turing's description of computability in the 1950 paper was all based on the finite capabilities of real, finite machines, illustrated by an account of the Manchester computer as it then stood. His claim was that the simulation of thought did not even require the full scope of computable functions, only that infinitesimal fraction of them which could be run using only a finite amount of "tape". (As a technical point, Turing's description did not even mention the "tape". This is because a finite tape can be absorbed into the instruction table of a Turing machine, which in turn he identified with the storage of a computer such as the Manchester computer. This resulted in him rather confusingly describing the full gamut of computable processes as requiring an "infinite store". This is, however, just the unlimited supply of tape as prescribed in 1936, not an infinite instruction table.) He suggested a necessary storage capacity of 10^9 bits, which of course is far surpassed by modern personal computers.

A fortiori, there was no suggestion in this paper of anything beyond the scope of computability. There were three areas of Turing's discussion where mathematics beyond the computable was raised, but in each case the thrust of Turing's argument was that computable operations would suffice. One of these was the Gödel argument, actually rather more weakly addressed in this than in his earlier papers, but still concluding that it had no force. The second lay in Turing's discussion of "the continuity of the nervous system". He claimed that the brain's basis in continuous matter, rather than being a discrete machine, was again no argument against the computability of thought: a discrete system could approximate a continuous one as closely as desired. The third was the concept of randomness, which Turing introduced without any serious definition. His illustration used "the digits of π" as a random sequence, and this is par excellence a computable input.

In fact, Turing's exposition ran through two stages, reflecting what has been suggested above as his "1936" and "1941" stages of development. First came the concept of computable functions, thought of as planned instructions (Turing, 1950), and then followed the finite discrete state machine picture. However, as he argued, these differently pictured machines could alike be implemented on a universal machine, the computer. This same two-part structure came into his final constructive proposals for the development of machine intelligence. Turing imagined rule-based programming (rather like expert systems as later developed), but also the "child machine" learning for itself. Turing concluded by recommending that "both approaches should be tried": he never suggested a rigid dichotomy between top-down and bottom-up approaches, which was later to divide Artificial Intelligence research so deeply.

In summary, Turing was able to claim:

> I believe that in about fifty years' time it will be possible to programme computers, with a storage capacity of about 10^9, to make them play the imitation game so well that an average interrogator will not have more than 70 per cent chance of making the right identification after five minutes of questioning.

This prophecy of the power of the computable was, of course, to stimulate the Loebner Prize competitions as the dateline of 2000 approached.

2.6 After the Test

No account of Alan Turing would be complete without a mention of his last years. "Computing machinery and intelligence" was shot through with courtroom images of juries and trials; they were prophetic. Turing was arrested for his affair with a young Manchester man in 1952. All homosexual behaviour was then criminal. He was seriously disturbed by the punishment that ensued: his brain was "treated" with oestrogen. But his mind did not atrophy in this new period. In 1950 he had begun serious work, involving use of the Manchester computer, on a new theory of non-linear partial differential equations, proposed as "the chemical basis of morphogenesis". This and other research continued vigorously despite the interruption. (So did his personal life, which as usual he refused to adjust to the expectations of society.)

The question arises as to whether there were further developments in Turing's ideas about machine intelligence after 1950. There is an indication that there were. In the following year he gave a popular talk on BBC radio (Turing, 1951). It was basically a version of what he had set out in the 1950 paper. He explained the principle that any mechanical process could be simulated by a program run on a single machine, the computer: in particular, he had in mind the function of the brain. But this time he inserted an important *caveat* that had not been made in 1950. The machine to be simulated

> should be of the sort whose behaviour is in principle predictable by calculation. We certainly do not know how any such calculation should be done, and it was even argued by Sir Arthur Eddington that on account of the indeterminacy principle in quantum mechanics no such prediction is even theoretically possible.

Copeland (1999) has rightly signalled the importance of this new point, but his critical context suggests a link with the "oracle", a particular kind of uncomputable function used in Turing's 1938 work (Turing, 1939). But Turing made no reference to this "oracle" when admitting this possibly fatal flaw in his argument about the brain as a discrete state machine. The question he was raising was whether the space-time properties of quantum-mechanical physics could be captured by a discrete state machine model. And this was a question which went back to his earliest serious thought, being related to the work by Eddington and von Neumann that he was reading in 1928–1932, especially that of Eddington.

In 1932 Turing had speculated, influenced both by Eddington, and by trauma in his personal life, that quantum mechanics underpinned free will (Hodges, 1983). The relationship between von Neumann and Turing has enjoyed much attention because of the question of who first had the idea of a practical universal machine (Hodges, 1983; Davis, 2000). Less well known is that Turing's first serious research study was of von Neumann's work on the foundations of quantum mechanics. Von Neumann clarified the measurement or *reduction* process in quantum mechanics; it is this which is not predictable. Seventy years later, there is no agreed or compelling explanation of how or when such "reduction" occurs. In 1953–1954, Turing wrote to his friend and colleague Robin Gandy that he was trying to invent a new quantum mechanics, and raised questions about the reduction principle, as discussed in Gandy (1954). Probably he was trying to find a predictable theory of the reduction process, to close the loophole in his argument for the computability of brain processes. However, he died in 1954 before announcing any result.

His last published paper (Turing, 1954), was again semi-popular, appearing in *Penguin Science News*. This did not mention quantum mechanics, but it returned to the pure mathematics of computability (which had recently gained new life with advances in algebra), and gave an account of Gödel's theorem. His conclusion was surprisingly unlike that pronounced in 1950; he said that Gödel's theorem showed that "common sense" was needed in interpreting axiomatic systems, and this time the intuitive "common sense" was not asserted to be a something a machine could show as well as a human being. The year 1950 seems to have marked the peak of his confidence about the prospects for machine intelligence, but it is impossible to know how his views would have developed had he lived longer.

In recent years, Roger Penrose has taken up the two themes that Turing found most difficult to fit into his thesis of computable mental functions – Gödel's theorem and the quantum-mechanical reduction process – and has said that they must be connected (Penrose, 1989, 1994, 1996). Penrose holds that the function of the brain *cannot* be simulated by a computer program, because of its quantum-mechanical physical basis. Thus, for entirely materialist reasons, no truly intelligent behaviour will ever be simulated by a computer; the full Turing Test will never be passed. Many commentators have attacked this conclusion, but it must be said that the topics Penrose emphasises are those that Turing himself found central to his arguments.

We are now so used to the properties of digital machines that it is hard to imagine the world in which Turing conjured them from pure imagination in 1936. However, it is crucial to see that what Turing offered in 1950, based on this earlier work, was something that went beyond the traditional mind-matter debate, and beyond loose science-fiction talk about humans and robots. It had a new solid substance in the digital or discrete-state machine model, made clear as never before. This structure, however, had a non-obvious limitation expressed by Gödel's theorem and Turing's own discoveries in the theory of computability. Turing always had these questions of limits in mind. Turing's universal machine now seems to sweep all before it, and continues to captivate us with the apparently never-ending range of applications that it can encompass. Turing's own excitement for this project, his game-playing enthusiasm and iconoclastic humour, live on in every conversation-program writer of the present day. But it should be remembered that Turing's imitation game actually first arose as the "fair play" argument for escaping the force of Gödel's theorem and the serious puzzle posed by the limits of what can be computed.

References

Copeland, B. J., 1999, A lecture and two radio broadcasts on machine intelligence by Alan Turing, in: *Machine Intelligence* **15**, K. Furukawa, D. Michie, and S. Muggleton, eds., Oxford University Press, Oxford, pp. 445–475.

Davis, M., 2000, *The Universal Computer*, Norton, New York.

Gandy, R. O., 1954, Letter to M. H. A. Newman, in: *The Collected Works of A M. Turing: Mathematical Logic*, R. O. Gandy and C. E. M. Yates, eds., North-Holland, Amsterdam (2001).

Hodges, A., 1983, *Alan Turing: The Enigma*, Burnett, London, Simon & Schuster, New York, new editions: Vintage, London (1992), Walker, New York (2000).

Hodges, A., 1997, *Turing, a Natural Philosopher*, Phoenix, London, Routledge, New York (1999); included in: *The Great Philosophers*, R. Monk and F. Raphael, eds., Weidenfeld & Nicolson, London (2000).

Hodges, A., 2002, Alan M. Turing, in: *Stanford Encyclopedia of Philosophy*, E. Zalta, ed.; http://plato.stanford.edu/entries/turing.

McCulloch, W. S. and Pitts, W., 1943, A logical calculus of the ideas immanent in nervous activity, *The Bulletin of Mathematical Biophysics* **5**: 115–133.

Penrose, R., 1989, *The Emperor's New Mind*, Oxford University Press, Oxford.

Penrose, R., 1994, *Shadows of the Mind*, Oxford University Press, Oxford.

Penrose, R., 1996, Beyond the doubting of a shadow, Psyche electronic journal; http://psyche.csse.monash.edu.au/v2/psyche-2–23-penrose.html.

Turing, A. M., 1936, On computable numbers, with an application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society*, Series 2, **42**: 230–265.

Turing, A. M., 1939, Systems of logic defined by ordinals, *Proceedings of the London Mathematical Society*, Series 2, **45**: 161–228.

Turing, A. M., 1946, Proposed electronic calculator, report for the National Physical Laboratory, published in A. M. Turing's ACE report of 1946 and other papers, B. E. Carpenter and R. W. Doran, eds., MIT Press, Cambridge, MA (1986).

Turing, A. M., 1947, Lecture to the London Mathematical Society on 20 February 1947, published in A. M. Turing's ACE report of 1946 and other papers, B. E. Carpenter and R. W. Doran, eds., MIT Press, Cambridge, MA (1986).

Turing, A. M., 1948, Intelligent machinery, report for the National Physical Laboratory, published in *Machine Intelligence* **7**, B. Meltzer and D. Michie, eds. (1969).

Turing, A. M., 1950, Computing machinery and intelligence, *Mind* **50**: 433–460.

Turing, A. M., 1951, BBC radio talk, published in *Machine Intelligence* **15**, K. Furukawa, D. Michie, and S. Muggleton, eds., Oxford University Press, Oxford (1999).

Turing, A. M., 1954, Solvable and unsolvable problems, *Science News* **31**: 7–23.

Chapter 3
Computing Machinery and Intelligence

Alan M. Turing

Editors' Note: The following is the article that started it all – the article by Alan Turing which appeared in 1950 in the British journal, *Mind*. Accompanying the article are three running commentaries by Kenneth Ford, Clark Glymour, and Pat Hayes of the University of West Florida; Stevan Harnad of the University of Southampton; and Ayse Pinar Saygin of the University of California, San Diego, designated respectively by the symbols: ♠, ♣, and ♥. A fourth commentary by John Lucas of Merton College, Oxford, is found in Chapter 4.

3.1 The Imitation Game

I propose to consider the question, "Can machines think?"♣ This should begin with definitions of the meaning of the terms "machine" and "think". The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words "machine" and "think" are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, "Can machines think?" is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by

---

Manchester University

♣ Harnad: Turing starts on an equivocation. We know now that what he will go on to consider is not whether or not machines can think, but whether or not machines can do what thinkers like us can do – and if so, how. Doing is performance capacity, empirically observable. Thinking is an internal state. It correlates empirically observable as neural activity (if we only knew which neural activity corresponds to thinking!) and its associated quality introspectively observable as our own mental state when we are thinking. Turing's proposal will turn out to have nothing to do with either observing neural states or introspecting mental states, but only with generating performance capacity indistinguishable from that of thinkers like us.

R. Epstein et al. (eds.), *Parsing the Turing Test.*

another, which is closely related to it and is expressed in relatively unambiguous words.♠♣♥

The new form of the problem can be described in terms of a game which we call the "imitation game".♠♣ It is played with three people, a man (A), a woman

---

♠ FORD, GLYMOUR, AND HAYES: Turing derides deciding the question by an empirical survey of which sorts of objects the word "think" or its synonyms are positively applied to. Presumably, in 1950 people rarely if ever said of machines that they think, and few people in 1950 would have said that any machine, then or in the future, could *possibly* be said to think. The procedure is absurd because what people say, even what almost everyone agrees in saying, is often wildly wrong: a century before almost everyone would have agreed to the proposition that angels think.

♣ HARNAD: "Machine" will never be adequately defined in Turing's paper, although what will eventually be known as the "Turing Machine," the abstract description of a computer, will be. This will introduce a systematic ambiguity between a real physical system, doing something in the world, and another physical system, a computer, simulating the first system formally, but not actually doing what it does: an example would be the difference between a real airplane – a machine, flying in the real world – and a computer simulation of an airplane, not really flying, but doing something formally equivalent to it, in a (likewise simulated) "virtual world."

A reasonable definition of machine, rather than Turing Machine, might be any dynamical, causal system. That makes the universe a machine, a molecule a machine, and also waterfalls, toasters, oysters, and human beings. Whether or not a machine is man-made is obviously irrelevant. The only relevant property is that it is "mechanical" – i.e., behaves in accordance with the cause-effect laws of physics.

"Think" will never be defined by Turing at all; it will be replaced by an operational definition to the effect that "thinking is as thinking does." This is fine, for thinking cannot be defined in advance of knowing how thinking systems do it, and we do not yet know how. But we do know that we thinkers do it, whatever it is, when we think and we know when we are doing it (by introspection). So thinking, a form of consciousness, is already ostensively defined by just pointing to that experience we all have and know.

Taking a statistical survey like a Gallup poll instead, to find out people's opinions of what thinking is would indeed be a waste of time, as Turing points out – but then later in the paper he needlessly introduces the equivalent of a statistical survey as his criterion for having passed his Turing Test!

♥ SAYGIN: Operational definition: a definition of a theoretical construct that is stated in terms of concrete, observable procedures (Pelham, 1999). While some readers believe the imitation game is only a thought experiment, I think it is pretty clear that Turing is proposing an operational definition for machine thought. One could argue whether this is the best way to test for machine intelligence, but that would be a discussion of construct validity, i.e., the quality of someone's operational definitions, not the existence or lack thereof.

♠ FORD, GLYMOUR, AND HAYES: Turing's use of the singular here may be misleading, as we will see. There are many versions of "the" imitation game, and Turing himself seems to slide between them without giving the reader adequate notice. It might be best to take this paragraph as a description of a family of "games" that share a common theme: a real exemplar and an imitator, respectively trying to help and to fool a judge.

♣ HARNAD: Another unfortunate terminological choice: "Game" implies caprice or trickery, whereas Turing in fact means serious empirical business. The game is science, the future science of cognition – actually a branch of reverse bioengineering. "Imitation" has connotations of fakery or deception too, whereas what Turing will be proposing is a rigorous empirical methodology for testing theories of human cognitive performance capacity (and thereby also theories of the thinking that presumably engenders it). Calling this an "imitation game" (instead of a methodology for reverse-engineering human cognitive performance capacity) has invited generations of needless misunderstandings (Harnad, 1992).

(B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman.♣ He knows them by labels X and Y, and at the end of the game he says either "X is A and Y is B" or "X is B and Y is A". The interrogator is allowed to put questions to A and B thus:

C: Will X please tell me the length of his or her hair?

Now suppose X is actually A, then A must answer. It is A's object in the game to try and cause C to make the wrong identification. His answer might therefore be

"My hair is shingled, and the longest strands are about nine inches long."

In order that tones of voice may not help the interrogator the answers should be written, or better still, typewritten.♣ The ideal arrangement is to have a teleprinter communicating between the two rooms. Alternatively the question and answers can be repeated by an intermediary. The object of the game for the third player (B) is to help the interrogator. The best strategy for her is probably to give truthful answers. She can add such things as "I am the woman, don't listen to him!" to her answers, but it will avail nothing as the man can make similar remarks.

We now ask the question, "What will happen when a machine takes the part of A in this game?" Will the interrogator decide wrongly as often when the game is

---

♣ HARNAD: The man/woman test is an intuitive "preparation" for the gist of what will eventually be the Turing Test, namely, an empirical test of performance capacity. For this, it is first necessary that all non-performance data be excluded (hence the candidates are out of sight). This sets the stage for what will be Turing's real object of comparison, which is a thinking human being versus a (nonthinking) machine, a comparison that is to be unbiased by appearance.

Turing's criteria, as we know, will turn out to be two (though they are often confused or conflated): (1) Do the two candidates have identical performance capacity? (2) Is there any way we can distinguish them, based only on their performance capacity, so as to be able to detect that one is a thinking human being and the other is just a machine? The first is an empirical criterion: Can they both do the same things? The second is an intuitive criterion, drawing on what decades later came to be called our human "mind-reading" capacities (Frith and Frith, 1999): Is there anything about the way the candidates go about doing what they can both do that cues me to the fact that one of them is just a machine?

Turing introduces all of this in the form of a party game, rather like 20-Questions. He never explicitly debriefs the reader to the effect that what is really at issue is no less than the game of life itself, and that the "interrogator" is actually the scientist for question (1), and, for question (2), any of the rest of us, in every one of our daily interactions with one another. The unfortunate party-game metaphor again gave rise to needless cycles of misunderstandings in later writing and thinking about the Turing Test.

♣ HARNAD: This restriction of the test exclusively to what we would today call email interactions is, as noted, a reasonable way of preparing us for its eventual focus on performance capacity alone, rather than appearance, but it does have the unintended further effect of ruling out all direct testing of performance capacities other than verbal ones; and that is potentially a much more serious equivocation, to which we will return. For now, we should bear in mind only that if the criterion is to be Turing-indistinguishable performance-capacity, we can all do a lot more than just email!

played like this as he does when the game is played between a man and a woman? These questions replace our original, "Can machines think?"♠♣♥

♠ FORD, GLYMOUR, AND HAYES: This question can be understood in several ways, depending on what one takes Turing to mean by "this game." It is usually understood to mean a version of the game where, in the terminology of 1951 (published in 1950, exposition in 1951), a "man" – and a machine each try to persuade the judge that they are the human being. However, taken literally, and we see no reason not to, "the game" refers to the game just described, and then Turing seems to be proposing a comparative test of the ability of a man to pretend to be a woman, as against the ability of a computer to pretend to be a woman (the contestant in each case being a real woman, of course). This reading – the "gender test," in contradistinction to the "species test" usually assumed – may seem strange, but it has a number of subtle advantages, including the fact that the judge is not given a predisposition to be particularly alert for signs of non-human behaviour, and the fact that the players, man and machine, both have imitation tasks.

The critical question is whether, in typed conversation, a computer can pass as a woman as convincingly – and as *unconvincingly* – as can a man. In the gender test version, as posed, the test is not only of conversational competence, but also of a special kind of knowledge: the computer must "know" what it is like for a man to try to converse like a woman (Hayes and Ford, 1995).

Psychological speculations aside, one might reasonably object that different men and women and judges would yield widely varying accuracies of judgement, or that a sufficiently skilled judge, given sufficient time, would be able to distinguish most men from most women, so that to qualify as thoughtful, the computer would have a very low bar.

Many writers assume the game should be played with the question of gender (being female) replaced by the question of species (being human), so that the judge is faced with the task of differentiating a human participant from a machine pretending to be human. Notice that under this most common interpretation, the Turing Test slides from a test for intelligence, to a test of the ability of humans to distinguish members of their own species from mechanical impostors. This version is often called the "species version," and is the most popular understanding of the Turing Test, but it was not Turing's. In the gender test, the judge is still thinking about the differences between women and men, not humans and machines. The hypothesis that one of his subjects is not human is not even in his natural space of initial possibilities. This judge has exactly the same problem to solve as a judge in the original imitation game and could be expected to bring the same attitudes and skills to the problem. For a discussion of the two versions and the advantages of Turing's own version see (Genova, 1994; Sterrett, 2000).

♣ HARNAD: Here, with a little imagination, we can already scale up to the full Turing Test, but again we are faced with a needless and potentially misleading distraction: Surely the goal is not merely to design a machine that people mistake for a human being statistically more often than not! That would reduce the Turing Test to the Gallup poll that Turing rightly rejected in raising the question of what "thinking" is in the first place! No, if Turing's indistinguishability criterion is to have any empirical substance, the performance of the machine must be equal to that of a human being – to anyone and everyone, for a lifetime.

♥ SAYGIN: Note that here the machine takes the part of A, the man. The man was trying to convince the interrogator that he actually was the woman. Now that the machine takes the place of the man in the game, will it be trying to convince the interrogator that it is a woman? The answer could be yes or no, depending on interpretation (Piccinini, 2000; Saygin et al., 2000; Traiger, 2000). As it is now generally understood, the Turing Test tries to assess a machine's ability to imitate a human being, rather than its ability to simulate a woman in an imitation game. Most subsequent remarks on Turing's work in the following 50 years, as reviewed in Saygin et al. (2000), ignore the gender issue, and if they discuss the imitation game at all, consider a game that is played between a machine (A), a human (B), and an interrogator (C) whose aim is to determine which one of the two entities he/she

3.2 Critique of the New Problem

As well as asking, "What is the answer to this new form of the question", one may ask, "Is this new question a worthy one to investigate?" This latter question we investigate without further ado, thereby cutting short an infinite regress.♠

The new problem has the advantage of drawing a fairly sharp line between the physical and the intellectual capacities of a man.♣ No engineer or chemist claims to be able to produce a material which is indistinguishable from the human skin. It is possible that at some time this might be done, but even supposing this invention available we should feel there was little point in trying to make a "thinking

is conversing with is the human. In many cases the imitation game is considered irrelevant and the discussion revolves around the vague idea of a digital computer "passing for" a human and the relation this possibility bears to Artificial Intelligence (AI). The imitation game is not an analogy Turing introduced but failed to properly execute, nor is it a joke or a commentary on gender roles in society (unlike that of [Genova, 1994; Lassègue, 1996]). But I will suggest below that the seemingly quirky gender-based imitation game is in fact an ideal and fair test for machine intelligence.

♠ FORD, GLYMOUR, AND HAYES: Turing's intellectual strategy is to replace questions of a traditional philosophical form, e.g., provide necessary and sufficient conditions for something to be intelligent, with related questions for which there is a hope of providing an answer empirically or mathematically.

♣ HARNAD: It would have had that advantage, if the line had only been drawn between appearance and performance or between structure and function. But if the line is instead between verbal and non-verbal performance capacities, then it is a very arbitrary line indeed and a very hard one to defend. As there is no explicit or even inferable defence of this arbitrary line in any of Turing's paper (nor of an equally arbitrary line between those of our "physical capacities" that do and do not depend on our "intellectual capacities"), I will take it that Turing simply did not think this through. Had he done so, the line would have been drawn between the candidate's physical appearance and structure on the one hand, and its performance capacities, both verbal and non-verbal, on the other. Just as (in the game) the difference, if any, between the man and the woman must be detected from what they do, and not what they look like, so the difference, if any, between human and machine must be detected from what they do, and not what they look like. This would leave the door open for the robotic version of the Turing Test that we will discuss later, and not just for the email version.

But before a reader calls my own dividing line between structure and function just as arbitrary, let me quickly agree that Turing has in fact introduced a hierarchy of Turing Tests here, but not an infinity of them. The relevant levels of this hierarchy will turn out to be only the following 5:

T1: The local indistinguishable capacity to perform some arbitrary task, such as chess. T1 is not really a Turing Test at all, because it is so obviously subtotal; hence the machine candidate is easily distinguished from a human being by seeing whether it can do anything else, other than play chess. If it cannot, it fails the test.

T2: The indistinguishable performance capacity in email (verbal) exchanges. This seems like a self-contained performance module, for one can talk about anything and everything, and language has the same kind of universality that computers (Turing Machines) turned out to have. T2 even subsumes chess-playing. But does it subsume star-gazing, or even food-foraging? Can the machine go and see and then tell me whether the moon is visible tonight and can it go and unearth truffles and then let me know how it went about it? These are things that a machine with email capacity alone cannot do, yet every human being can.

machine" more human by dressing it up in such artificial flesh.♣ The form in which we have set the problem reflects this fact in the condition which prevents the interrogator from seeing or touching the other competitors, or hearing their voices.♣ Some other advantages of the proposed criterion may be shown up by specimen questions and answers. Thus:

Q: Please write me a sonnet on the subject of the Forth Bridge.
A: Count me out on this one. I never could write poetry.
Q: Add 34957 to 70764.
A: (Pause about 30 s and then give as answer) 105621.
Q: Do you play chess?
A: Yes.
Q: I have K at my K1, and no other pieces. You have only K at K6 and R at R1. It is your move. What do you play?
A: (After a pause of 15 s) R-R8 mate.

The question and answer method seems to be suitable for introducing almost any one of the fields of human endeavour that we wish to include.♣ We do not wish to penalize the machine for its inability to shine in beauty competitions,♣ nor to penalize a man for losing in a race against an aeroplane.♣ The conditions of our

T3: The indistinguishable performance capacity in robots (sensorimotor). This subsumes T2, and is (I will argue) the level of test that Turing really intended (or should have!).

T4: The indistinguishable external performance capacity, as well as internal structure and function. This subsumes T3 and adds all data that a neuroscientist might study. This is no longer strictly a Turing Test, because it goes beyond performance data, but it correctly embeds the Turing Hierarchy in a larger empirical hierarchy. Moreover, the boundary between T3 and T4 really is fuzzy: Is T3 or T4 blushing?

T5: The indistinguishable physical structure and function. This subsumes T4 and rules out any functionally equivalent but synthetic nervous systems: The T5 candidate must be indistinguishable from other human beings right down to the last molecule.

♣HARNAD: Here Turing correctly rejects T5 and T4 – but certainly not T3.

♣HARNAD: Yes, but using T2 as the example has inadvertently given the impression that T3 is excluded too.

♣HARNAD: This correctly reflects the universal power of natural language (to say and describe anything in words). But "almost" does not fit the Turing criterion of identical performance capacity.

♣HARNAD: This is the valid exclusion of appearance (moreover, most of us could not shine in beauty competitions either).

♣HARNAD: Most of us could not beat Deep Blue at chess, nor even attain ordinary grandmaster level. It is only generic human capacities that are at issue, not those of any specific individual. On the other hand, just about all of us can walk and run. And even if we are handicapped (an anomalous case, and hardly the one on which to build one's attempts to generate positive performance capacity), we all have some sensorimotor capacity. (Neither Helen Keller nor Stephen Hawking are disembodied email-only modules.)

game make these disabilities irrelevant.♣ The "witnesses" can brag, if they consider it advisable, as much as they please about their charms, strength, or heroism, but the interrogator cannot demand practical demonstrations.♣

The game may perhaps be criticized on the ground that the odds are weighted too heavily against the machine. If the man were to try and pretend to be the machine he would clearly make a very poor showing. He would be given away at once by slowness and inaccuracy in arithmetic. May not machines carry out something which ought to be described as thinking but which is very different from what a man does? This objection is a very strong one, but at least we can say that if, nevertheless, a machine can be constructed to play the imitation game satisfactorily, we need not be troubled by this objection.♠♥

It might be urged that when playing the "imitation game" the best strategy for the machine may possibly be something other than imitation of the behaviour of a man. This may be, but I think it is unlikely that there is any great effect of this kind.

♣ HARNAD: Disabilities and appearance are indeed irrelevant. But non-verbal performance capacities are certainly not. Indeed, our verbal abilities may well be grounded in our non-verbal abilities (Cangelosi, 2001; Harnad, 1990; Steels and Kaplan, 1999). Actually, by "disability," Turing means non-ability, i.e., absence of an ability; he does not really mean being disabled in the sense of being physically handicapped, although he does mention Helen Keller later.

♣ HARNAD: This would definitely be a fatal flaw in the Turing Test, if Turing had meant it to exclude T3 – but I doubt that is what he meant. He was just arguing that it is performance capacity that is decisive (for the empirical problem that future cognitive science would eventually address), not something else that might depend on irrelevant features of structure or appearance. He merely used verbal performance as his intuition-priming example, without meaning to imply that all "thinking" is verbal and only verbal performance capacity is relevant.

♠ FORD, GLYMOUR, AND HAYES: It is clear that Turing intended passing the Turing Test to be an uncontroversial criterion *sufficient* for thought, not a necessary one. He allows machines to be inhumanly capable, for example. Indeed, the few electronic computers which existed at the time he was writing were already inhumanly capable in doing arithmetic, which is of course why large funds were expended in designing and constructing them.

The Turing Test is, however, a poorly designed experiment, depending entirely on the competence of the judge. As Turing noted above, a human would be instantly revealed by his comparative inadequacies in arithmetic unless, of course, the computer were programmed to be arithmetically incompetent. Likewise, according to media reports, some judges at the first Loebner competition (1991), a kind of Turing test contest held at the Computer Museum in Boston, rated a human as a machine on the grounds that she produced extended, well-written paragraphs of informative text at dictation speed without typing errors. (Apparently, this is now considered an inhuman ability in parts of our culture.)

♥ SAYGIN: Turing did not live long enough to reply to most critiques of his work, but in this paper he foresees many criticisms he thinks may be made by others and formulates some advance arguments against them (§6). Nevertheless, even those issues Turing diligently addresses have been raised in subsequent discussions. For example, he has subsequently been criticized both for his test being too anthropomorphic and limited (Millar, 1973), and on the basis that playing the imitation game is just one thing an intelligent machine can do and is not general enough for purposes of intelligence granting (Gunderson, 1964).

In any case there is no intention to investigate here the theory of the game, and it will be assumed that the best strategy is to try to provide answers that would naturally be given by a man.

3.3 The Machines Concerned in the Game

The question which we put in §1 will not be quite definite until we have specified what we mean by the word "machine". It is natural that we should wish to permit every kind of engineering technique to be used in our machines.♣ We also wish to allow the possibility than an engineer or team of engineers may construct a machine which works, but whose manner of operation cannot be satisfactorily described by its constructors because they have applied a method which is largely experimental.♣♥ Finally, we wish to exclude from the machines men born in the usual manner.♣ It is difficult to frame the definitions so as to satisfy these three conditions. One might for instance, insist that the team of engineers should be all of one sex, but this would not really be satisfactory, for it is probably possible to rear a complete individual from a single cell of the skin (say) of a man.♠ To do so would be a feat of biological technique deserving of the very highest praise, but we would not be inclined to regard it as a case of "constructing a thinking machine".♣ This prompts

♣ HARNAD: This passage (soon to be withdrawn!) implies that Turing did not mean only computers: any dynamical system we build is eligible (as long as it delivers the performance capacity). But we do have to build it, or at least have a full causal understanding of how it works. A cloned human being cannot be entered as the machine candidate (because we did not build it and do not know how it works), even though we are all "machines" in the sense of being causal systems (2000, 2003).

♣ HARNAD: Here is the beginning of the difference between the field of AI, whose goal is merely to generate a useful performance tool, and cognitive modelling (CM), whose goal is to explain how human cognition is generated. A device we built without knowing how it works would suffice for AI but not for CM.

♥ SAYGIN: Turing would clearly allow many kinds of machines to pass the test, and more importantly, through various means. Several researchers opposed this idea, especially the latter point, holding that restrictions should be placed on internal information processing if a machine is to be granted thought (Block, 1981; Gunderson, 1964). Is Turing happy to grant intelligence to any old hack that may be programmed to play the imitation game? Or is he so confident that the problem is too hard that he is willing to take the risk of having a quick and dirty solution?

♣ HARNAD: This does not, of course, imply that we are not machines, but only that the Turing Test is about finding out what kind of machine we are, by designing a machine that can generate our performance capacity, but by a functional means that we understand because we designed them.

♠ FORD GLYMOUR, AND HAYES: Turing's anticipation of cloning was not out of the blue. In the period in which this paper was written, he had a strong interest in mathematical biology; especially in morphogenesis. He published one paper on the topic and wrote a number of others. They are available in his Collected Papers.

♣ HARNAD: This is because we want to explain thinking capacity, not merely duplicate it.

us to abandon the requirement that every kind of technique should be permitted. We are the more ready to do so in view of the fact that the present interest in "thinking machines" has been aroused by a particular kind of machine, usually called an "electronic computer" or "digital computer". Following this suggestion we only permit digital computers to take part in our game.♣

♣ HARNAD: This is where Turing withdraws the eligibility of all engineering systems but one, introducing another arbitrary restriction – one that would again rule out T3. Turing said earlier (correctly) that any engineering device ought to be eligible. Now he says it can only be a computer. His motivation is partly, of course, the fact that the computer (Turing Machine) has turned out to be universal, in that it can simulate any other kind of machine. But here we are squarely in the T2/T3 equivocation, for a simulated robot in a virtual world is neither a real robot, nor can it be given a real robotic Turing Test, in the real world. Both T2 and T3 are tests conducted in the real world. But an email interaction with a virtual robot in a virtual world would be T2, not T3.

To put it another way, with the Turing Test we have accepted, with Turing, that thinking is as thinking does. But we know that thinkers can and do more than just talk. And it remains what thinkers can do that our candidate must likewise be able to do, not just what they can do verbally. Hence, just as flying is something that only a real plane can do, and not a computer-simulated virtual plane, be it ever so Turing-equivalent to the real plane – so passing T3 is something only a real robot can do, not a simulated robot tested by T2, be it ever so Turing-equivalent to the real robot. (I also assume it is clear that Turing Testing is testing in the real world: a virtual-reality simulation [VR] would be no kind of a Turing Test; it would merely be fooling our senses in the VR chamber, rather than testing the candidate's real performance capacity in the real world.)

So the restriction to computer simulation, though perhaps useful for planning, designing and even pretesting the T3 robot, is merely a practical methodological strategy. In principle, any engineered device should be eligible, and it must be able to deliver T3 performance, not just T2.

It is of interest that contemporary cognitive robotics has not gotten as much mileage out of computer-simulation and virtual-worlds as might have been expected, despite the universality of computation. "Embodiment" and "situatedness" (in the real world) have turned out to be important ingredients in empirical robotics (Brooks, 2002; Steels and Kaplan, 1999), with the watchword being that the real world is better used as its own model (rather than virtual robots having to simulate, hence second-guess in advance, not only the robot, but the world too).

The impossibility of second-guessing the robot's every potential "move" in advance, in response to every possible real-world contingency, also points to a latent (and I think fatal) flaw in T2 itself: Would it not be a dead giveaway if one's email T2 pen pal proved incapable of commenting on the analogue family photos we kept inserting with our text? (If he can process the images, he is not just a computer, but at least a computer plus A/D peripheral devices, already violating Turing's arbitrary restriction to computers alone.) Or if one's pen pal was totally ignorant of contemporaneous real-world events, apart from those we describe in our letters? Would not even its verbal performance break down if we questioned it too closely about the qualitative and practical details of sensorimotor experience? Could all of that really be second-guessed purely verbally in advance?

This restriction appears at first sight to be a very drastic one. I shall attempt to show that it is not so in reality. To do this necessitates a short account of the nature and properties of these computers.♠♣

It may also be said that this identification of machines with digital computers, like our criterion for "thinking", will only be unsatisfactory if (contrary to my belief), it turns out that digital computers are unable to give a good showing in the game.♣

There are already a number of digital computers in working order, and it may be asked, "Why not try the experiment straight away? It would be easy to satisfy the conditions of the game. A number of interrogators could be used, and statistics compiled to show how often the right identification was given." The short answer is that we are neither asking whether all digital computers would do well in the game nor whether the computers at present available would do well, but whether there are imaginable computers which would do well.♠ But this is only the short answer. We shall see this question in a different light later.

♠ Ford, Glymour, and Hayes: The "universal machine" idea that a computer can be designed that can in principle simulate *all* other computers, is now widely understood, but it was not at all obvious when Turing was writing, and indeed the idea was widely derided or rejected as ludicrous. The multitude of purposes that computers could serve was little appreciated. A senior British government scientific advisor asserted that the entire country would only need four or five computers, on the grounds that they could be used only for generating elevation tables for naval gunnery. Even von Neumann thought the most important application of computers in mathematics would be to compute examples that would then give mathematicians intuitions about proofs. It seems safe to say that nobody, probably not even Turing, could have foreseen the many uses to which computers have been put in modern society.

The next few pages are a tour de force of exposition for the time Turing was writing, but will seem obvious to many people in this and future generations.

♣ Harnad: The account of computers that follows is useful and of course correct, but it does not do anything at all to justify restricting the Turing Test to candidates that are computers. Hence this arbitrary restriction is best ignored.

♣ Harnad: This is the "game" equivocation again. It is not doubted that computers will give a good showing, in the Gallup poll sense. But empirical science is not just about a good showing: An experiment must not just fool most of the experimentalists most of the time! If the performance-capacity of the machine must be indistinguishable from that of the human being, it must be totally indistinguishable, not just indistinguishable more often than not. Moreover, some of the problems that I have raised for T2 – the kinds of verbal exchanges that draw heavily on sensorimotor experience – are not even likely to give a good showing if the candidate is only a digital computer, regardless of how rich a database it is given in advance.

♠ Ford, Glymour, and Hayes: Again, a simple point that has often been misunderstood since.

3.4 Digital Computers♥

The idea behind digital computers may be explained by saying that these machines are intended to carry out any operations which could be done by a human computer.♠ The human computer is supposed to be following fixed rules♣; he has no authority to deviate from them in any detail. We may suppose that these rules are supplied in a book, which is altered whenever he is put on to a new job. He has also an unlimited supply of paper on which he does his calculations. He may also do his multiplications and additions on a "desk machine", but this is not important.

If we use the above explanation as a definition we shall be in danger of circularity of argument. We avoid this by giving an outline of the means by which the desired effect is achieved. A digital computer can usually be regarded as consisting of three parts:

(i) Store
(ii) Executive unit
(iii) Control

The store is a store of information, and corresponds to the human computer's paper, whether this is the paper on which he does his calculations or that on which his book of rules is printed. In so far as the human computer does calculations in his head a part of the store will correspond to his memory.

The executive unit is the part which carries out the various individual operations involved in a calculation. What these individual operations are will vary from machine to machine. Usually fairly lengthy operations can be done such as

♥ Saygin: Turing's treatment here and in the next section is one of the most concise, but clear explanations of basic theory of computing that exists – I think it could be useful for teaching purposes. Although Turing is one of the fathers of computer science, being a pioneer in a field does not in itself mean that one is able to speak coherently about that field at an introductory level. I think his success is partly due to the fact that he himself is able to see, in a way characteristic of an interdisciplinary scientist, the relations between the abstract notions of computation, different levels of application, behavioural manifestation, and philosophical analysis.

♠ Ford, Glymour, and Hayes: It is often said that computers were invented around 1940, but this claim would have sounded strange at that date. The bare term "computer" then meant a human being who (often aided by an electromechanical calculator) performed computations for a living, or in support of some other intellectual activity such as theoretical physics, astronomy, or code-breaking. Computational skill was highly prized, and to be called a "computer" in 1940 was a professional compliment, as it had been since at least the 1850s.

In fact, the famous astronomer Simon Newcomb wrote a recommendation letter for one of his calculators in which he said, "His mind is more like a mathematical machine than any I have ever encountered," which was high praise indeed. To explain the operation of an electronic computer (the adjective, now seeming redundant, is commonly dropped) in terms of rule-books used by human beings, was therefore a perfectly natural expository device. However, this device can be misleading when read with hindsight, since it can suggest that the "thinking" part of the computer is the part of it which corresponds in this expositional metaphor to the human computer, i.e., the "executive unit" or CPU, which is nowadays simply a piece of etched silicon. A similar misunderstanding is the layman's objection – which Turing mentions later – that computers "can only obey instructions."

♣ Harnad: This goes on to describe what has since become the standard definition of computers as rule-based symbol-manipulating devices (Turing machines).

"Multiply 3540675445 by 7076345687" but in some machines only very simple ones such as "Write down 0" are possible.

We have mentioned that the "book of rules" supplied to the computer is replaced in the machine by a part of the store. It is then called the "table of instructions". It is the duty of the control to see that these instructions are obeyed correctly and in the right order. The control is so constructed that this necessarily happens.

The information in the store is usually broken up into packets of moderately small size. In one machine, for instance, a packet might consist of ten decimal digits. Numbers are assigned to the parts of the store in which the various packets of information are stored, in some systematic manner. A typical instruction might say:

"Add the number stored in position 6809 to that in 4302 and put the result back into the latter storage position."

Needless to say it would not occur in the machine expressed in English. It would more likely be coded in a form such as 6809430217. Here, 17 says which of various possible operations is to be performed on the two numbers. In this case the operation is that described above, viz. "Add the number…" It will be noticed that the instruction takes up ten digits and so forms one packet of information, very conveniently. The control will normally take the instructions to be obeyed in the order of the positions in which they are stored, but occasionally an instruction such as

"Now obey the instruction stored in position 5606, and continue from there" may be encountered, or again

"If position 4505 contains 0 obey next the instruction stored in 6707, otherwise continue straight on."

Instructions of these latter types are very important because they make it possible for a sequence of operations to be replaced over and over again until some condition is fulfilled, but in doing so to obey, not fresh instructions on each repetition, but the same ones over and over again. To take a domestic analogy. Suppose Mother wants Tommy to call at the cobbler's every morning on his way to school to see if her shoes are done, she can ask him afresh every morning. Alternatively she can stick up a notice once and for all in the hall which he will see when he leaves for school and which tells him to call for the shoes, and also to destroy the notice when he comes back if he has the shoes with him.

The reader must accept it as a fact that digital computers can be constructed, and indeed have been constructed, according to the principles we have described, and that they can in fact mimic the actions of a human computer very closely.♠♥

♠ Ford, Glymour, and Hayes: One can almost sense the frustration that Turing may have felt when trying to find a convincing way to persuade a sceptical audience that mechanical computation was indeed possible. Again, all these metaphors about Mother and Tommy seem curiously antiquated to a contemporary sensibility.

♥ Saygin: The analogies here are overly simplified for purposes of explanation. However, I think Turing does believe at some level that most human behaviour is guided by "programs" of the sort that one prepares to make machines perform actions. It is easy to criticize this view, claiming AI has not produced much in terms of intelligent behaviour based on programs of this sort, and that

The book of rules which we have described our human computer as using is of course a convenient fiction. Actual human computers really remember what they have got to do. If one wants to make a machine mimic the behaviour of the human computer in some complex operation one has to ask him how it is done, and then translate the answer into the form of an instruction table.♠ Constructing instruction tables is usually described as "programming". To "programme a machine to carry out the operation A" means to put the appropriate instruction table into the machine so that it will do A.♠

An interesting variant on the idea of a digital computer is a "digital computer with a random element". These have instructions involving the throwing of a die or some equivalent electronic process; one such instruction might for instance be, "Throw the die and put the resulting number into store 1000". Sometimes such a machine is described as having free will (though I would not use this phrase

---

there is much more to being human than just following rules. These latter views are not inconsistent with Turing's thought, and a careful reading of his work will reveal he is also aware that human behaviour is guided by a program much more complex than these analogies suggest, even when random elements are thrown into the picture. It is also likely to be a rather opaque, if not cryptic, program since it will be based on a lifetime of perception, sensation, action, learning and buildup on little innate substrate in a rather experience-driven manner over a long period of time. But it does not follow from the complexity and opacity of the "human behavior program" that runs on the brain that is qualitatively different from a computer program of the sort discussed here. The point here is not to defend what is sometimes called the "computational view of the mind," which, in the light of recent research in cognitive neuroscience, is too symbolic and restricted to account for the level of complexity needed to model human minds – I am pretty sure Turing would not subscribe to that view either. But creating such a program based on ideas from modern cognitive science research and theory (e.g., based on connectionism, dynamical systems theory, embodied cognition and theoretical neuroscience) could be consistent with Turing's views.

♠ FORD, GLYMOUR, AND HAYES: This sentence is prescient. Turing was probably thinking of iterative numerical computations of the kind that human computers did indeed perform, but in fact (with a generous interpretation of "instruction table") this is exactly how "knowledge-based systems" are constructed, which have proven capable of performing many tasks which were not previously considered to lie within the province of human computation, but instead to require other human abilities such as "intuition" or "judgement."

♠ FORD, GLYMOUR, AND HAYES: Again, describing programming as the construction of look-up tables now seems very archaic. We are now much more familiar with programming as centrally concerned with *language*: programs typically manipulate expressions which themselves may be further interpreted as code, and the actual physical machine may be many levels below all this programming, almost invisible to the human user and even to the programmer. What Turing is describing, and what was at the time the only method of programming available, is what we would now call "assembly-code" programming, an activity that only a few specialists ever practice. In most modern computers, virtually every instruction executed by the CPU was generated by some other piece of code rather than written by a human programmer. Writing assembly code requires an intimate knowledge of the inner workings of the computer's hardware. Turing was what would now be called a wizard or a hacker. Given his views on programming technique and hardware design, he would probably be horrified by the wastefulness of modern programming, in which billions of machine cycles are wasted waiting for human typists' fingers to hit the next key.

myself).♣ It is not normally possible to determine from observing a machine whether it has a random element, for a similar effect can be produced by such devices as making the choices depend on the digits of the decimal for π.

Most actual digital computers have only a finite store. There is no theoretical difficulty in the idea of a computer with an unlimited store. Of course, only a finite part can have been used at any one time. Likewise only a finite amount can have been constructed, but we can imagine more and more being added as required. Such computers have special theoretical interest and will be called infinitive capacity computers.

The idea of a digital computer is an old one. Charles Babbage, Lucasian Professor of Mathematics at Cambridge from 1828 to 1839, planned such a machine, called the Analytical Engine, but it was never completed. Although Babbage had all the essential ideas, his machine was not at that time such a very attractive prospect. The speed which would have been available would be definitely faster than a human computer but something like 100 times slower than the Manchester machine, itself one of the slower of the modern machines. The storage was to be purely mechanical, using wheels and cards.

The fact that Babbage's Analytical Engine was to be entirely mechanical will help us to rid ourselves of a superstition. Importance is often attached to the fact that modern digital computers are electrical, and that the nervous system also is electrical.♠ Since Babbage's machine was not electrical, and since all digital computers are in a sense equivalent, we see that this use of electricity cannot be of theoretical importance. Of course, electricity usually comes in where fast signalling is concerned, so that it is not surprising that we find it in both these connections. In the nervous system chemical phenomena are at least as important as electrical. In certain computers the storage system is mainly acoustic. The feature of using electricity is thus seen to be only a very superficial similarity. If we wish to find such similarities we should look rather for mathematical analogies of function.

3.5 Universality of Digital Computers

The digital computers considered in the last section may be classified amongst the "discrete state machines". These are the machines which move by sudden jumps or clicks from one quite definite state to another. These states are sufficiently different

♣ HARNAD: Nor would I. But surely an even more important feature for a Turing Test candidate than a random element or statistical functions would be autonomy in the world – which is something a T3 robot has a good deal more of than a T2 pen pal. The ontic side of free will – namely, whether we ourselves, real human beings, actually have free will – rather exceeds the scope of Turing's paper (Harnad, 1982b). So too does the question of whether a Turing test-passing machine would have any feelings at all (whether free or otherwise; Harnad, 1995). What is clear, though, is that computational rules are not the only ways to "bind" and determine performance: ordinary physical causality can do so too.

♠ FORD, GLYMOUR, AND HAYES: A similar superstition is the view that brains cannot be thought of as computers because they are made of organic material.

for the possibility of confusion between them to be ignored. Strictly speaking there are no such machines. Everything really moves continuously. But there are many kinds of machine which can profitably be *thought* of as being discrete-state machines. For instance, in considering the switches for a lighting system it is a convenient fiction that each switch must be definitely on or definitely off. There must be intermediate positions, but for most purposes we can forget about them. As an example of a discrete-state machine we might consider a wheel which clicks round through 120° once a second, but may be stopped by a lever which can be operated from outside; in addition a lamp is to light in one of the positions of the wheel. This machine could be described abstractly as follows. The internal state of the machine (which is described by the position of the wheel) may be q_1, q_2, or q_3. There is an input signal i_0 or i_1 (position of lever). The internal state at any moment is determined by the last state and input signal according to the table

		Last State		
		q_1	q_2	q_3
Input	i_0	q_2	q_3	q_1
	i_1	q_2	q_3	q_1

The output signals, the only externally visible indication of the internal state (the light) are described by the table

State	q_1	q_2	q_3
Output	o_0	o_0	o_1

This example is typical of discrete-state machines. They can be described by such tables provided they have only a finite number of possible states.

It will seem that given the initial state of the machine and the input signals it is always possible to predict all future states.♣ This is reminiscent of Laplace's view that from the complete state of the universe at one moment of time, as described by the positions and velocities of all particles, it should be possible to predict all future states. The prediction which we are considering is, however, rather nearer to practicability than that considered by Laplace. The system of the "universe as a whole" is such that quite small errors in the initial conditions can have an overwhelming effect at a later time.♠ The displacement of a single electron by a

♣ Harnad: The points about determinism are probably red herrings. The only relevant property is performance capacity. Whether either the human or the machine is completely predictable is irrelevant. (Both many-body physics and complexity theory suggest that neither causal determinacy nor following rules guarantee predictability in practise – and this is without even invoking the arcana of quantum theory.)

♠ Ford, Glymour, and Hayes: In more modern terminology, the universe is in some sense *chaotic*. Chaos theory had not been developed when Turing was writing, of course.

billionth of a centimetre at one moment might make the difference between a man being killed by an avalanche a year later, or escaping. It is an essential property of the mechanical systems which we have called "discrete state machines" that this phenomenon does not occur. Even when we consider the actual physical machines instead of the idealized machines, reasonably accurate knowledge of the state at one moment yields reasonably accurate knowledge any number of steps later.

As we have mentioned, digital computers fall within the class of discrete-state machines. But the number of states of which such a machine is capable is usually enormously large. For instance, the number for the machine now working at Manchester is about $2^{165,000}$, i.e., about $10^{50,000}$. Compare this with our example of the clicking wheel described above, which had three states. It is not difficult to see why the number of states should be so immense. The computer includes a store corresponding to the paper used by a human computer. It must be possible to write into the store any one of the combinations of symbols which might have been written on the paper. For simplicity suppose that only digits from 0 to 9 are used as symbols. Variations in handwriting are ignored. Suppose the computer is allowed 100 sheets of paper each containing 50 lines each with room for 30 digits. Then the number of states is $10^{100 \times 50 \times 30}$, i.e., $10^{150,000}$. This is about the number of states of three Manchester machines put together. The logarithm to the base two of the number of states is usually called the "storage capacity" of the machine. Thus the Manchester machine has a storage capacity of about 165,000 and the wheel machine of our example about 1.6. If two machines are put together their capacities must be added to obtain the capacity of the resultant machine. This leads to the possibility of statements such as "The Manchester machine contains 64 magnetic tracks each with a capacity of 2,560, eight electronic tubes with a capacity of 1,280. Miscellaneous storage amounts to about 300 making a total of 174,380."♠♥

Given the table corresponding to a discrete-state machine it is possible to predict what it will do. There is no reason why this calculation should not be carried out by means of a digital computer. Provided it could be carried out sufficiently quickly the digital computer could mimic the behaviour of any discrete-state machine. The imitation game could then be played with the machine in question (as B) and the mimicking digital computer (as A) and the interrogator would be unable to distinguish them.♠ Of course, the digital computer must have an adequate storage

♠ Ford, Glymour, and Hayes: It is hard to compare this accurately with modern machines, but a typical laptop computer may have an active memory capacity of approximately 10|9, and a hard disc capacity of perhaps a hundred times more. Of course, not all of this huge capacity may be being used in a way that Turing would have thought sensible.

♥ Saygin: Revisiting the question of whether Turing was proposing the game as a real operational definition or test. It seems highly unlikely to me that a man proposing a thought experiment would spend such time, space and energy to explain not only what he means by "thinking" but also exactly what kind of machine a digital computer is.

♠ Ford, Glymour, and Hayes: Here, Turing seems to be using the term "imitation game" in a very generic sense.

capacity as well as working sufficiently fast. Moreover, it must be programmed afresh for each new machine which it is desired to mimic.♥

This special property of digital computers, that they can mimic any discrete-state machine, is described by saying that they are *universal* machines. The existence of machines with this property has the important consequence that, considerations of speed apart, it is unnecessary to design various new machines to do various computing processes. They can all be done with one digital computer, suitably programmed for each case. It will be seen that as a consequence of this all digital computers are in a sense equivalent.♣

We may now consider again the point raised at the end of §3. It was suggested tentatively that the question, "Can machines think?" should be replaced by "Are there imaginable digital computers which would do well in the imitation game?" If we wish we can make this superficially more general and ask "Are there discrete-state machines which would do well?" But in view of the universality property we see that either of these questions is equivalent to this, "Let us fix our attention on one particular digital computer C. Is it true that by modifying this computer to have an adequate storage, suitably increasing its speed of action, and providing it with an appropriate program, C can

♥ SAYGIN: Turing reminds us of the imitation game here. He is trying to emphasize the universality aspect of the discrete-state machine, but he does so using the original indistinguishability test we started with. There is an interesting twist: What is the interrogator looking for in this instantiation of the game? Which one is the machine? Which one is the mimicking digital computer? What would you even ask in order to differentiate between the two? It does not make much sense, unless Turing means they will play the gender-based imitation game. He never says we change the game into anything other than the gender-based game anyway. It may sound silly or pointless to compare how well two entities imitate a woman in a teletype conversation, but as I will elaborate below, tests and experiments often construct situations that do not have direct equivalents in real life (i.e., they do not always have high ecological validity).

♣ HARNAD: All true, but all irrelevant to the question of whether a digital computer alone could pass T2, let alone T3. The fact that eyes and legs can be simulated by a computer does not mean a computer can see or walk (even when it is simulating seeing and walking). So much for T3. But even just for T2, the question is whether simulations alone can give the T2 candidate the capacity to verbalize and converse about the real world indistinguishably from a T3 candidate with autonomous sensorimotor experience in the real world.

(I think yet another piece of unnoticed equivocation by Turing – and many others – arises from the fact that thinking is not directly observable, which helps us imagine that computers think. But even without having to invoke the other-minds problem (Harnad, 1991), one needs to remind oneself that a universal computer is only formally universal: it can describe just about any physical system, and simulate it in symbolic code, but in doing so, it does not capture all of its properties: Exactly as a computer-simulated airplane cannot really do what a plane does (i.e., fly in the real world), a computer-simulated robot cannot really do what a real robot does (act in the real world) – hence there is no reason to believe it is really thinking. A real robot may not really be thinking either, but that does require invoking the other-minds problem, whereas the virtual robot is already disqualified for exactly the same reason as the virtual plane: both fail to meet the Turing Test criterion itself, which is real performance capacity, not merely something formally equivalent to it!).

be made to play satisfactorily the part of A in the imitation game, the part of B being taken by a man?”♠♥

♠ FORD, GLYMOUR, AND HAYES: Turing implicitly uses what has become known as the Church-Turing thesis: The computable functions are all and only those computable by a Universal Turing Machine. The idea that one can make a given computer act like any other just by reprogramming it, given enough processor speed and memory capacity, is now a familiar idea in our culture; but persuading his audience of its reasonableness was probably one of Turing’s most difficult tasks of exposition in 1951.

♥ SAYGIN: Notice that the woman has disappeared from the game altogether. But the objectives of A, B, and C remain unaltered; at least Turing does not explicitly state any change. To be precise, what we have is a digital computer and a man both trying to convince an interrogator that they are the real woman.

Why the fuss about the woman, the man, and the replacement? Turing does not seem the type of person who would beat around the bush for no reason. What is going on here?

One could say the gender-based imitation game is merely an analogy, serving the purpose of making the paper easier to understand. But in a paper that starts with the sentence, “Can machines think?” would something like “Let us take a machine and a man and see if the machine can convince interrogators that it is a human being via teletype conversations” be really much harder to process or understand? Or maybe Turing was simply careless and forgot to clarify that we are no longer talking about the gender-based imitation game. Given the level of detail and precision in Turing’s writing (see Sections 4 and 5 of this paper), this is unlikely to be the explanation. Also bear in mind Turing is a successful mathematician, a discipline characterized by precision of definition and rigor in argument and generalization, which would make it unlikely that he is being sloppy.

Here is my explanation for the quirks of the game – I cannot guarantee that this is what Turing intended, but I think it is consistent with the way he thinks and writes. Neither the man in the gender-based imitation game nor any kind of machine is a woman. Furthermore what Turing proposes is essentially to compare the machine’s success against that of the man – not to look at whether it actually “beats” the woman. The man and the machine are measured in terms of their respective performances and their performances are comparable because they are both simulating something which they are not. Even though it is regarded as obscure by many, the imitation game could be a carefully planned experimental design. It provides a fair basis for comparison: the woman (either as a participant in the game or as a concept) acts as a neutral point so that the two imposters can be assessed in how well they perform the imitation. In other words, Turing gives us a question that is to be examined via a carefully defined task, an experimental group (digital computers) and a control group (men). This setup looks more like an operational definition given in terms of an experimental design than anything else.

It might seem that we are a long way from such relatively minor methodological points being relevant. But at least two studies have shown that people’s judgments of computers’ conversational performance are substantially influenced by whether or not they know in advance that their conversational partners may be machines. In the 1970s, a group of scientists devised an electronic interviewing environment where experienced psychiatrists corresponded with both real-life paranoid patients and computer programs simulating paranoid behaviour through teletype. The judges were not told that some of the interviewees could be computer programs. Details can be found in Colby et al. (1972), but to summarize, the finding was that the psychiatric judges did not do better than chance guessing at identifying the computers from the human patients.

In a more recent study, we carried out an experiment to examine possible relationships between pragmatic violations and imitation game performance, using real excerpts from human–computer conversations (Saygin and Cicekli, 2002). Due to the design of the experiment, some subjects made pragmatic judgments on a set of conversations without being told there were computers involved

3.6 Contrary Views on the Main Question

We may now consider the ground to have been cleared and we are ready to proceed to the debate on our question, "Can machines think?" and the variant of it quoted at the end of the last section. We cannot altogether abandon the original form of the problem, for opinions will differ as to the appropriateness of the substitution and we must at least listen to what has to be said in this connexion.

It will simplify matters for the reader if I explain first my own beliefs in the matter. Consider first the more accurate form of the question. I believe that in about 50 years' time it will be possible, to program computers, with a storage capacity of about 10^9, to make them play the imitation game so well that an average interrogator will not have more than 70% chance of making the right identification after 5 min of questioning.♠♣♥ The original

---

before they were told about the imitation game and asked to evaluate the computers' performance in the same conversations, while other subjects knew computers' involvement from the outset. We noted that even something seemingly trivial like having read the conversations only once without any bias prior to being asked to make decisions regarding the computers' behaviour had a differential effect on people's judgments. In particular, the analysis revealed that when people were faced with anomalies in the conversations, those who knew about computers' involvement tended to automatically think these were indicative of the conversational partner's identity (i.e., by the fact that it is a machine). On the other hand, unbiased subjects tried to work through the problematic exchanges in the same way they would in a pragmatically confusing conversation between humans. Now note that the gender-based imitation game is immune to the bias that knowledge of computer participation may bring. It allows the interrogators to work out pragmatic violations (and in general, exercise their naive psychology) the way they normally do; therefore, this design allows us to give the digital computers a fairer shot at performing well.

In sum, the gender-based imitation game is a good experimental design. It provides an operational definition (i.e., a larger question is replaced by a task we can evaluate). It is controlled; the task is simulating something both the experimental and control subjects are not. Furthermore, any bias the interrogator (which may be thought of as a measurement device) brings in will be based on gender expectations, which will tend not to affect the two groups differentially. Viewed in this light, the quirky imitation game may well be one of the few ways to fairly and experimentally assess machine thought.

♠ FORD, GLYMOUR, AND HAYES: Turing was right about the memory capacity of modern computers, but it is widely claimed that he was wrong in his Turing Test prediction: here we are, 50 years later, and where are the passers of his imitation game? However, notice that Turing says that it will be *possible*. That question is still moot: maybe it is possible. Certainly, computers have already performed many tasks that were previously thought of as requiring human sagacity of some kind. But in any case, very few contemporary AI researchers are seriously trying to build a machine to play Turing's imitation game. Instead they are concerned with exploring the computational machinery of intelligence itself, whether in humans, dogs, computers, or aliens. The scientific aim of AI research is to understand intelligence as computation, and its engineering aim is to build machines that surpass or extend human mental abilities in some useful way. Trying to imitate a human conversation (however "intellectual" it may be) contributes little to either ambition. Progress in AI is not measured by checking fidelity to a human conversationalist. And yet many critics of AI are complaining of a lack of progress toward this old ambition. But perhaps we should forgive the critics, as even many AI textbooks still offer the Turing Test as AI's ultimate goal, which seems akin to starting a textbook on aeronautical engineering with an explanation that the goal of the field is to make machines that fly so exactly like pigeons that they can even fool other pigeons (Ford and Hayes, 1998).

question, "Can machines think?" I believe to be too meaningless to deserve discussion.♣ Nevertheless, I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of

---

This is of course a huge area of controversy, far larger than we have space here to survey, but one point may be worth making. In making this 50-year prediction, Turing may have meant that a determined 50-year effort devoted to this single aim could succeed, a kind of imitation-game Manhattan project, or a fivefold expansion of the decade of national effort it took to get a human being to the moon and back. Certainly, given his wartime experiences of large-scale government-sponsored projects, this interpretation is not implausible; and later in the paper he suggests explicitly that it would be a 3,000-man-year project.

♣ HARNAD (p. 41): No doubt this party-game/Gallup poll criterion can be met by today's computer programs – but that remains as meaningless a demographic fact today as it was when predicted 50 years ago. Like any other science, cognitive science is not the art of fooling most of the people for some or most of the time! The candidate must really have the generic performance capacity of a real human being – capacity that is totally indistinguishable from that of a real human being to any real human being (for a lifetime, if need be!). No tricks: real performance capacity.

♥ SAYGIN (p. 41): More than 50 years have gone by since Turing wrote these often-quoted words, yet we are nowhere near "the goal." How could Turing, a man with such great vision and intellect, so grossly underestimate the time it would take to tackle the problems he left behind? I grant it that Turing underestimated either how hard the task at hand is, or how long it takes to carry out such a task. But I wonder sometimes if that is the whole story. Could he, in addition, have overestimated how hard future researchers would work at the problems? I think the latter has played more of a role than is commonly considered in the fact that we have a gaping hole between Turing's expectations and the current state of AI. Think about it: Can we really say we followed Turing's advice, gave it our all and it did not work? Or did we try shortcuts and little hacks and cheats and gave up in pursuit of "useful" AI when they did not work? The point here is not to criticize AI researchers for working on this or that topic. I only want to note that we do not know how much closer we would have been at developing AI systems that can communicate using natural language had we actually pursued it as a serious, full-time goal. Turing's tone in this paper leads me to think that the future he envisioned is based on scientists, philosophers, and programmers working hard and wholeheartedly towards the goal, patiently overcoming obstacles and making steady progress. What really happened in the AI arena was a buzz, a wave of optimism with many researchers believing that successful AI was right around the corner, finding the whole endeavor challenging but "cool," and wanting to make it work and make it work fast. However, when the problem proved too difficult to yield fruit soon, there was an ensuing burnout, which soon led to a lack of serious interest in endeavors such as the Turing Test. Some AI researchers even went as far as outwardly refusing to work on the Turing Test, defending that it belongs in history books rather than current research agendas, indeed calling it "harmful for AI" (Hayes and Ford, 1995).

♣ HARNAD: It is not meaningless, it is merely indecisive: What we mean by "think" is, on the one hand, what thinking creatures can do and how they can do it, and, on the other hand, what it feels-like to think. What thinkers can do is captured by the Turing Test. A theory of how they do it is provided by how our man-made machine does it. (If there are several different successful machines, it is a matter of normal inference-to-the-best-theory.) So far, nothing is meaningless. Now we ask: Do the successful candidates really feel, as we do when we think? This question is not meaningless; it is merely unanswerable – in any other way than by being the candidate. It is the familiar old other-minds problem (Harnad, 1991).

machines thinking without expecting to be contradicted.♠♣ I believe further that no useful purpose is served by concealing these beliefs. The popular view that scientists proceed inexorably from well-established fact to well-established fact, never being influenced by any improved conjecture, is quite mistaken. Provided it is made clear which are proved facts and which are conjectures, no harm can result. Conjectures are of great importance since they suggest useful lines of research.♠♣

I now proceed to consider opinions opposed to my own.

3.6.1 The Theological Objection

Thinking is a function of man's immortal soul.♣ God has given an immortal soul to every man and woman, but not to any other animal or to machines. Hence no animal or machine can think.

♠ Ford, Glymour, and Hayes: That prediction has in a sense been vindicated: the idea of thinking machines has indeed entered popular culture, and is widely discussed as either a present reality or an imminent one. What is more interesting, however, and what Turing apparently did not foresee, is the emergence of a kind of linguistic creepage, where the boundaries of "real thinking" are redrawn so as to exclude whatever it is that machines become able to do. When electronic computers were new, the ability to perform mental arithmetic rapidly and accurately was widely admired as a human mental ability; now it is "merely" mechanical. Now that a computer has beaten the world chess champion, skill at chess is becoming perceived as "merely" mechanical. This gradual but irresistible cultural shift in meaning also bears on the utility of the imitation game: One has to ask, to which generation does the judge belong? Behaviour that someone of Turing's generation would have found convincing may completely fail to impress someone who grew up with talking teddy bears.

♣ Harnad: Yes, but only at a cost of demoting "thinking" to meaning only "information processing" rather than what you or I do when we think, and what that feels-like.

♠ Ford, Glymour, and Hayes: One can make out a reasonable case that this paper, and its bold conjectures, played a central role in the emergence of AI and cognitive science in the 1960s and 1970s.

♣ Harnad: This is mistaken. Yes, science proceeds by a series of better approximations, from empirical theory to theory. But the theory here would be the actual design of a successful Turing Test candidate, not the conjecture that computation (or anything else) will eventually do the trick. Turing is confusing formal conjectures (such as that the Turing machine and its equivalents capture all future notions and instances of what we mean by "computation" – the "Church/Turing Thesis") and empirical hypotheses, such as that thinking is just computation. Surely the Turing Test is not a license for saying that we are explaining thinking better and better as our candidates fool more and more people longer and longer. On the other hand, something else that sounds superficially similar to this could be said about scaling up the Turing Test empirically by designing a candidate that can do more and more of what we can do. And Turing testing certainly provides a methodology for such cumulative theory-building and theory-testing in cognitive science.

♣ Harnad: The real theological objection is not so much that the soul is immortal but that it is immaterial. This view also has non-theological support from the mind/body problem: no one – theologian, philosopher, or scientist – has even the faintest hint of an idea of how mental states

I am unable to accept any part of this, but will attempt to reply in theological terms. I should find the argument more convincing if animals were classed with men, for there is a greater difference, to my mind, between the typical animate and the inanimate than there is between man and the other animals.♣ The arbitrary character of the orthodox view becomes clearer if we consider how it might appear to a member of some other religious community. How do Christians regard the Moslem view that women have no souls?♠ But let us leave this point aside and return to the main argument. It appears to me that the argument quoted above implies a serious restriction of the omnipotence of the Almighty. It is admitted that there are certain things that He cannot do such as making one equal to two, but should we not believe that He has freedom to confer a soul on an elephant if He sees fit? We might expect that He would only exercise this power in conjunction with a mutation which provided the elephant with an appropriately improved brain to minister to the needs of this soul. An argument of exactly similar form may be made for the case of machines. It may seem different because it is more difficult to "swallow". But this really only means that we think it would be less likely that He would consider the circumstances suitable for conferring a soul. The circumstances in question are discussed in the rest of this paper. In attempting to construct such machines we should not be irreverently usurping His power of creating souls, any more than we are in the procreation of children: rather we are, in either case, instruments of His will be providing mansions for the souls that He creates.

However, this is mere speculation. I am not very impressed with theological arguments whatever they may be used to support. Such arguments have often been found unsatisfactory in the past. In the time of Galileo it was argued that the texts, "And the sun stood still… and hasted not to go down about a whole day" (Joshua x. 13) and "He laid the foundations of the earth, that it should not move at any time" (Psalm cv. 5) were an adequate refutation of the Copernican theory. With our present knowledge such an argument appears futile. When that knowledge was not available it made a quite different impression.♠

---

can be material states (or, as I prefer to put it, how functional states can be felt states). This problem has been dubbed "hard" (Chalmers in Shear, 1997). It may be even worse: it may be insoluble (Harnad, 2001). But this is no objection to Turing Testing which, even if it will not explain how thinkers can feel, does explain how they can do what they can do.

♣ HARNAD: Yes, and this is why the other-minds problem comes into its own in doing Turing testing of machines rather than in doing mind reading of our own species and other animals. ("Animate" is a weasel word, though, for vitalists are probably also animists; Harnad, 1994a.)

♠ FORD, GLYMOUR, AND HAYES: Turing's source for this view is unknown. The contrary opinion is given in the Qu'ran.

♠ FORD, GLYMOUR, AND HAYES: The last three sentences are a bit odd. We only acquired our present knowledge because many people (Galileo himself, Bruno before him, Kepler, Newton, etc.) already found the argument futile.

3.6.2 The "Heads in the Sand" Objection

"The consequences of machines thinking would be too dreadful. Let us hope and believe that they cannot do so."

This argument is seldom expressed quite so openly as in the form above. But it affects most of us who think about it at all. We like to believe that Man is in some subtle way superior to the rest of creation. It is best if he can be shown to be *necessarily* superior, for then there is no danger of him losing his commanding position. The popularity of the theological argument is clearly connected with this feeling. It is likely to be quite strong in intellectual people, since they value the power of thinking more highly than others, and are more inclined to base their belief in the superiority of Man on this power.

I do not think that this argument is sufficiently substantial to require refutation. Consolation would be more appropriate: perhaps this should be sought in the transmigration of souls.♠

3.6.3 The Mathematical Objection

There are a number of results of mathematical logic which can be used to show that there are limitations to the powers of discrete-state machines. The best known of these results is known as Gödel's theorem (1931), and shows that in any sufficiently powerful logical system statements can be formulated which can neither be proved nor disproved within the system, unless possibly the system itself is inconsistent. There are other, in some respects similar, results due to Church (1936), Kleene (1935), Rosser, and Turing (1937). The latter result is the most convenient to

---

♠ Ford, Glymour, and Hayes: Turing raises an issue still central in our own time: the limitation of scientific inquiry by religious dogma, and in particular by the doctrine of souls. The fundamental religious objection to embryonic stem cell research is that when a human sperm cell and an ovum form an embryonic cell, the cell is "ensouled," it supernaturally acquires a soul. In this subsection, irritation or exasperation seems to have overwhelmed Turing's usual ingenuity in argument. While many current advocates of "Heads in the Sand" may be utterly thoughtless, there is a history of arguments for the position, all of which Turing ignores. William James, in *The Will to Believe*, argued roughly as follows: we should not believe that human intelligence has a purely biological, chemical, and physical explanation, for if we did so believe, we would conclude there is no basis for moral assessment; the world would be a worse place if we believed there is no basis for moral assessment, and it is rational not to act to bring about the worse case. The argument is in the spirit of Pascal's Wager. Pascal, the Turing of the 17th century, argued that one should act so as to cause oneself to believe in God, because the expected payoff of believing is infinitely greater than the expected payoff of not believing. We think neither James' argument nor Pascal's is sound, but the arguments deserve at least as much consideration as others Turing does respond to.

consider, since it refers directly to machines, whereas the others can only be used in a comparatively indirect argument: for instance, if Gödel's theorem is to be used we need in addition to have some means of describing logical systems in terms of machines, and machines in terms of logical systems. The result in question refers to a type of machine which is essentially a digital computer with an infinite capacity. It states that there are certain things that such a machine cannot do. If it is rigged up to give answers to questions as in the imitation game, there will be some questions to which it will either give a wrong answer, or fail to give an answer at all however much time is allowed for a reply. There may, of course, be many such questions, and questions which cannot be answered by one machine may be satisfactorily answered by another. We are of course, supposing for the present that the questions are of the kind to which an answer "Yes" or "No" is appropriate, rather than questions such as "What do you think of Picasso?" The questions that we know the machines must fail on are of this type, "Consider the machine specified as follows.... Will this machine ever answer "Yes" to any question?" The dots are to be replaced by a description of some machine in a standard form, which could be something like that used in §5. When the machine described bears a certain comparatively simple relation to the machine which is under interrogation, it can be shown that the answer is either wrong or not forthcoming. This is the mathematical result: it is argued that it proves a disability of machines to which the human intellect is not subject.

The short answer to this argument is that although it is established that there are limitations to the powers of any particular machine, it has only been stated, without any sort of proof, that no such limitations apply to the human intellect.♣♥ But I do not think this view can be dismissed quite so lightly. Whenever one of these machines is asked the appropriate critical question, and gives a definite answer, we know that this answer must be wrong, and this gives us a certain feeling of superiority. Is this feeling illusory? It is no doubt quite genuine, but I do not think too much importance should be attached to it. We too often give wrong answers to questions ourselves to be justified in being very pleased at such evidence of fallibility on the part of the machines. Further, our superiority can only be felt on such an occasion in relation to the one machine over which we have scored our petty triumph. There would be no question of triumphing simultaneously over *all* machines. In short, then, there might be men cleverer than any

---

♣ HARNAD: Gödel's theorem shows that there are statements in arithmetic that are true, and we know are true, but their truth cannot be computed. Some have interpreted this as implying that "knowing" (which is just a species of "thinking") cannot be just computation. Turing replies that maybe the human mind has similar limits, but it seems to me it would have been enough to point out that "knowing" is not the same as "proving". Gödel shows the truth is unprovable, not that it is unknowable. There are far better reasons for believing that thinking is not computation.

♥ SAYGIN: I do not see this issue discussed much anywhere, but I think it is a profound idea. How do we know that the human brain-computer would "halt" given the description of another human brain-computer and asked what it would reply to a given input?

given machine, but then again there might be other machines cleverer again, and so on.♠

Those who hold to the mathematical argument would, I think, mostly be willing to accept the imitation game as a basis for discussion. Those who believe in the two previous objections would probably not be interested in any criteria.♠

3.6.4 The Argument from Consciousness

This argument is very well expressed in *Professor Jefferson's* Lister Oration for 1949, from which I quote. "Not until a machine can write a sonnet or compose a concerto because of thoughts and emotions felt, and not by the chance fall of symbols, could we agree that machine equals brain♣ – that is, not only write it but know that it had written it. No mechanism could feel (and not merely artificially signal, an easy contrivance) pleasure at its successes, grief when its valves fuse, be

---

♠ FORD, GLYMOUR, AND HAYES: It is interesting that the two men who might be said to have provided the intellectual machinery of this objection, Gödel and Turing, came to opposite philosophical conclusions. Turing's conclusion was that since human thinkers are equivalent in inferential power to machines, there must be truths that they – we – are incapable of proving. Gödel started with the assumption that there were no mathematical truths that human beings could not, in principle, grasp, and concluded that human beings could not be machines. Penrose, who has a similarly grand view of the power of human thought, cites Gödel with approval.

The argument can be further illustrated by considering the fact that many people believe themselves to be consistent: It follows from the Gödel result mentioned that no consistent system of a reasonable deductive power can conclusively establish its own consistency. Turing concluded that human beings have inconsistent beliefs, a view that Gödel and Penrose apparently reject.

It is hard to see how this disagreement could be resolved conclusively, since the central issue could not be determined empirically; there is no finite amount of information that could conclusively establish that human beings, or indeed anything else in the universe, are capable of such all-encompassing deductive powers. Further, if the inferential abilities of humans are in fact bounded by Turing computability, a human could not even reliably converge to the truth about whether a system (including humans) is or not a computer from its responses to inputs.

♠ FORD, GLYMOUR, AND HAYES: Turing accurately predicted and then succinctly rebutted the many subsequent resurrections of the mathematical argument, including much of what Roger Penrose has written against the computational conception of mind. This debate has now taken place a number of times; the "mathematical objection" seems to get resuscitated every 15 years or so. For a more thorough analysis of the technical issues involved in this debate see (Laforte, Hayes, and Ford, 1998).

♣ HARNAD: This standard argument against the Turing Test (repeated countless times in almost exactly the same way until the present day) is merely a restatement of the other-minds problem: THERE IS NO WAY TO KNOW WHETHER EITHER HUMANS OR MACHINES DO WHAT THEY DO BECAUSE THEY FEEL like it – or whether they feel anything at all, for that matter. But there is a lot to be known from identifying what can and cannot generate the capacity to do what humans can do. (The limits of symbol-manipulation [computation] are another matter, and one that can be settled empirically, based on what sorts of machine can and cannot pass the Turing Test; Harnad, 2003.)

warmed by flattery, be made miserable by its mistakes, be charmed by sex, be angry or depressed when it cannot get what it wants."♠

This argument appears to be a denial of the validity of our test. According to the most extreme form of this view the only way by which one could be sure that machine thinks is to *be* the machine and to feel oneself thinking. One could then describe these feelings to the world, but of course, no one would be justified in taking any notice. Likewise according to this view the only way to know that a *man* thinks is to be that particular man. It is in fact the solipsist point of view.♠ It may be the most logical view to hold but it makes communication of ideas difficult.♣ A is liable to believe "A thinks but B does not" whilst B believes "B thinks but A does not". Instead of arguing continually over this point it is usual to have the polite convention that everyone thinks.

I am sure that Professor Jefferson does not wish to adopt the extreme and solipsist point of view. Probably he would be quite willing to accept the imitation game as a test.♠

♠ Ford, Glymour, and Hayes: Professor Jefferson's view is a special case of the view argued subsequently by Keith Gunderson, that to be intelligent, a thing must not only do some of what we do intelligently (e.g., converse or imitate), but must do *all* of what we do, and must do it *as* we do. Essentially, this position demands that any intelligent machine actually *be* human or, more precisely, it requires that any intelligent machine have a humanoid phenomenology: it must have the rather elusive quality that being it *feels like* being human. Of course, any such criterion rejects the "behavioural" quality of any test such as the one Turing proposes. Unfortunately, at least until we develop a theory of phenomenology linked adequately to the rest of science, it rejects *any* objective test of any kind.

♠ Ford, Glymour, and Hayes: This is a slight oversimplification. It is possible to hold this view rationally without adopting solipsism. In fact, the current mainstream view of consciousness, aptly described by David Chalmers in "The Conscious Mind" as "the hard problem," is that the presence of such an "inner view" is indeed characteristic of consciousness, and that any kind of behaviour might, in principle, be produced by a "zombie" which has no such inner life; and that therefore, no such behavioural criterion can be taken to be definitive evidence for consciousness. Of course, this is not in itself an objection to the presence of thought itself, as Chalmers himself is at pains to point out, since a zombie may indeed be thinking without being conscious of thinking. (The distinctions being applied in this area have become much more delicate than they were when Turing was writing.)

♣ Harnad: Turing is dead wrong here. This is not solipsism (i.e., not the belief that only I exist and all else is my dream). It is merely the other-minds problem (Harnad, 1991); and it is correct, but irrelevant – or rather put into perspective by the Turing Test: there is no one else we can know has a mind but our own private selves, yet we are not worried about the minds of our fellow-human beings, because they behave just like us and we know how to mind read their behaviour. By the same token, we have no more or less reason to worry about the minds of anything else that behaves just like us – so much so that we cannot tell them apart from other human beings. Nor is it relevant what stuff they are made out of, since our successful mind reading of other human beings has nothing to do with what stuff they are made out of. It is based only on what they do.

♠ Ford, Glymour, and Hayes: This is the second time that Turing assumes confidently that an intellectual opponent would "probably" be willing to accept his imitation game as a valid test. This optimism seems misplaced if it is supposed to indicate a willingness to accede to Turing's philosophical position; what he may have meant, however, is that if faced with an actual machine which passed the test, Jefferson would probably agree that it was, in fact, thinking intelligently.

The game (with the player B omitted) is frequently used in practice under the name of viva voce to discover whether some one really understands something or has "learnt it parrot fashion". Let us listen in to a part of such a *viva voce*[♥]:

Interrogator:	In the first line of your sonnet which reads "Shall I compare thee to a summer's day", would not "a spring day" do as well or better?
Witness:	It wouldn't scan.
Interrogator:	How about "a winter's day" That would scan all right.
Witness:	Yes, but nobody wants to be compared to a winter's day.
Interrogator:	Would you say Mr. Pickwick reminded you of Christmas?
Witness:	In a way.
Interrogator:	Yet Christmas is a winter's day, and I do not think Mr. Pickwick would mind the comparison.
Witness:	I don't think you're serious. By a winter's day one means a typical winter's day, rather than a special one like Christmas.

And so on. What would Professor Jefferson say if the sonnet-writing machine was able to answer like this in the viva voce? I do not know whether he would regard the machine as "merely artificially signalling" these answers, but if the answers were as satisfactory and sustained as in the above passage I do not think he would describe it as "an easy contrivance". This phrase is, I think, intended to cover such devices as the inclusion in the machine of a record of someone reading a sonnet, with appropriate switching to turn it on from time to time.

In short then, I think that most of those who support the argument from consciousness could be persuaded to abandon it rather than be forced into the solipsist position. They will then probably be willing to accept our test.[♠]

I do not wish to give the impression that I think there is no mystery about consciousness. There is, for instance, something of a paradox connected with any attempt to localize it.[♠] But I do not think these mysteries necessarily need to be solved before we can answer the question with which we are concerned in this paper.

[♥] SAYGIN: One final comment about the gender-based game: The game actually ended up being ecologically valid after all! Rest assured that at this very moment a variant is being played in many Internet chat rooms across the world, with participants trying to guess each others' gender (often with the hope of forming romantic relations) based on "teletype" connections.

[♠] FORD, GLYMOUR, AND HAYES: Perhaps Professor Jefferson need not be driven to solipsism at all. He might reply to Turing with an argument from similarity: "I am composed of the same kind of tissue and cells as any other human. When a human feels pain, from a burning finger, say, we know there is a course of nerve signals from the digit to the brain, and as neuroscience advances we will be able to follow the trace in more detail. It is alike in every human as far as we can tell, and so also in me. But I, Jefferson, know I *feel* pain. Since other humans are composed and function as I do, I can reasonably infer that they feel pain in like circumstances. But I have no such assurance for the digital computer."

[♠] FORD, GLYMOUR, AND HAYES: Indeed there is. For examples, see Daniel Dennett's delightful essay "Where Am I?"

3.6.5 Arguments from Various Disabilities

These arguments take the form, "I grant you that you can make machines do all the things you have mentioned but you will never be able to make one to do X". Numerous features X are suggested in this connexion. I offer a selection:

Be kind, resourceful, beautiful, friendly, have initiative, have a sense of humour, tell right from wrong, make mistakes, fall in love, enjoy strawberries and cream, make some one fall in love with it, learn from experience, use words properly, be the subject of its own thought, have as much diversity of behaviour as a man, do something really new. (Some of these disabilities are given special consideration as indicated [on the next few pages]).♣

No support is usually offered for these statements. I believe they are mostly founded on the principle of scientific induction. A man has seen thousands of machines in his lifetime. From what he sees of them he draws a number of general conclusions. They are ugly, each is designed for a very limited purpose, when required for a minutely different purpose they are useless, the variety of behaviour of any one of them is very small, etc. Naturally he concludes that these are necessary properties of machines in general. Many of these limitations are associated with the very small storage capacity of most machines. (I am assuming that the idea of storage capacity is extended in some way to cover machines other than discrete-state machines. The exact definition does not matter as no mathematical accuracy is claimed in the present discussion.) A few years ago, when very little had been heard of digital computers, it was possible to elicit much incredulity concerning them, if one mentioned their properties without describing their construction. That was presumably due to a similar application of the principle of scientific induction. These applications of the principle are of course, largely unconscious. When a burnt child fears the fire and shows that he fears it by avoiding it, I should say that he was applying scientific induction. (I could, of course, also describe his behaviour in many other ways.) The works and customs of mankind do not seem to be very suitable material to which to apply scientific induction. A very large part of space-time must be investigated, if reliable results are to be obtained. Otherwise we may (as most English children do) decide that everybody speaks English, and that it is silly to learn French.♠

There are, however, special remarks to be made about many of the disabilities that have been mentioned. The inability to enjoy strawberries and cream may have struck the reader as frivolous. Possibly a machine might be made to enjoy this

♣ Harnad: Turing rightly dismisses this sort of scepticism (which I have dubbed "Granny Objections" by pointing out that these are empirical questions about what computers (and other kinds of machines) will eventually be shown to be able to do. The performance items on the list, that is. The mental states (feelings), on the other hand, are moot because of the other-minds problem.

♠ Ford, Glymour, and Hayes: Could it be that the French actually converse by secretly passing notes in English back and forth?

delicious dish, but any attempt to make one do so would be idiotic. What is important about this disability is that it contributes to some of the other disabilities, e.g., to the difficulty of the same kind of friendliness occurring between man and machine as between white man and white man, or between black man and black man.♠

The claim that "machines cannot make mistakes" seems a curious one. One is tempted to retort, "Are they any the worse for that?" But let us adopt a more sympathetic attitude, and try to see what is really meant. I think this criticism can be explained in terms of the imitation game. It is claimed that the interrogator could distinguish the machine from the man simply by setting them a number of problems in arithmetic. The machine would be unmasked because of its deadly accuracy. The reply to this is simple. The machine (programmed for playing the game) would not attempt to give the *right* answers to the arithmetic problems. It would deliberately introduce mistakes in a manner calculated to confuse the interrogator.♠ A mechanical fault would probably show itself through an unsuitable decision as to what sort of a mistake to make in the arithmetic. Even this interpretation of the criticism is not sufficiently sympathetic. But we cannot afford the space to go into it much further. It seems to me that this criticism depends on a confusion between two kinds of mistake. We may call them "errors of functioning" and "errors of conclusion". Errors of functioning are due to some mechanical or electrical fault which causes the machine to behave otherwise than it was designed to do. In philosophical discussions one likes to ignore the possibility of such errors; one is therefore discussing "abstract machines". These abstract machines are mathematical fictions rather than physical objects. By definition they are incapable of errors of functioning. In this sense we can truly say that "machines can never make mistakes".♥ Errors of conclusion can only arise when some meaning is attached to the output signals from the machine. The machine might, for instance, type out mathematical equations, or sentences in English. When a false proposition is typed we say that the machine has committed an error of conclusion. There is clearly no reason at all for saying that a machine cannot make this kind of mistake. It might do nothing but type out repeatedly "0 = 1". To take a less perverse example, it might have some method for drawing

---

♠ Ford, Glymour, and Hayes: It is hard to know whether it is Turing's own racism in evidence in the preceding sentence, or merely a realistic acknowledgement of the state of race relations in England and her colonies in 1950 (and alas, often since). Given the post-WW2 history of racial relations in the USA, this racial reference may convey an unfortunate impression, but one needs to view it both from a 50-year perspective and a transatlantic shift. Turing was probably making a point about cross-*cultural* difficulties of communication.

♠ Ford, Glymour, and Hayes: This point is obvious, but seems to undermine the value of imitation game to research in computer science. As Turing seems to realize, playing the imitation game – in any version – is really an exercise in mendacity. When one bears this in mind, it is hardly surprising that relatively little effort has in fact been expended in seriously trying to succeed at such a very silly and pointless goal, when there are so many more interesting and useful applications available for computer technology and AI.

♥ Saygin: In a similar vein, can we say neurons can or cannot make mistakes?

conclusions by scientific induction. We must expect such a method to lead occasionally to erroneous results.♠♥

The claim that a machine cannot be the subject of its own thought can of course only be answered if it can be shown that the machine has *some* thought with *some* subject matter. Nevertheless, "the subject matter of a machine's operations" does seem to mean something, at least to the people who deal with it. If, for instance, the machine was trying to find a solution of the equation $x^2 - 40x - 11 = 0$ one would be tempted to describe this equation as part of the machine's subject matter at that moment. In this sort of sense a machine undoubtedly can be its own subject matter. It may be used to help in making up its own programs, or to predict the effect of alterations in its own structure. By observing the results of its own behaviour it can modify its own programs so as to achieve some purpose more effectively. These are possibilities of the near future, rather than Utopian dreams.♠♥

The criticism that a machine cannot have much diversity of behaviour is just a way of saying that it cannot have much storage capacity. Until fairly recently a storage capacity of even a thousand digits was very rare. ♠

♠ Ford, Glymour, and Hayes: Such programs have been written. The most famous examples are probably Lenat's AM, which learned a number of mathematical concepts, and Simon's BACON program. Inductive learning is now a generally useful technique in many application areas. The same basic techniques are used in televisions to guess likely programs to record, to rapidly detect likely credit card fraud, and in industrial control applications.

♥ Saygin: Computers are now routinely used in several real-world applications that are best addressed by inductive learning from large amounts of data. A lot of solicitation we receive (e.g., credit card and loan offers) is based on computations carried out by computer programs that try to predict our interests and spending habits. They make mistakes – otherwise I would not be getting all this junk information about how I should finance my "second home."

♠ Ford, Glymour, and Hayes: This was not merely a conjecture by Turing. In some sense it was obvious that a machine's interactions with the environment could alter the machine's program, and, equally, that a machine could have a representation of its own program and use that representation as an object of computation – i.e., make inferences about it. Optimizing compilers have been doing this kind of thing for decades. It is amusing to recall that one of the papers presented at one of the first AI meetings ever held, in 1956, was about the design of a Fortran complier.

♥ Saygin: Turing is correct about this prediction. The field of machine learning is one of the most fruitful lines of research in AI. Furthermore, computer programs that modify themselves are also used (e.g., many types of genetic algorithm and neural network systems).

♠ Ford, Glymour, and Hayes: Now that machines have gigantic capacities, it is sometimes objected that they still are rather unenterprising in their behaviour: they do not strike out for new-ground, form new concepts, or behave unpredictably. This criticism misses the fact that most computer programs are *designed* to be unenterprising because they are more useful that way. It is not hard to make a laptop computer extremely unpredictable in its behaviour, but – like a servant with attention-deficit disorder – that also makes it much less useful. What Turing seems to have in mind is the objection that computers do not seem to adaptively violate their own behavioural regularities in appropriate circumstances, a capacity that Sterrett has suggested is a mark of genuine intelligence. There is, however, no argument that computers cannot be designed to do as much.

The criticisms that we are considering here are often disguised forms of the argument from consciousness. Usually if one maintains that a machine *can* do one of these things, and describes the kind of method that the machine could use, one will not make much of an impression. It is thought that the method (whatever it may be, for it must be mechanical) is really rather base. Compare the parentheses in Jefferson's statement quoted on page 47.♠

3.6.6 Lady Lovelace's Objection

Our most detailed information of Babbage's Analytical Engine comes from a memoir by Lady Lovelace (1842). In it she states, "The Analytical Engine has no pretensions to *originate* anything. It can do *whatever we know how to order it* to perform" (her italics). This statement is quoted by *Hartree* (1949) who adds: "This does not imply that it may not be possible to construct electronic equipment which will "think for itself", or in which, in biological terms, one could set up a conditioned reflex, which would serve as a basis for "learning". Whether this is possible in principle or not is a stimulating and exciting question, suggested by some of these recent developments. But it did not seem that the machines constructed or projected at the time had this property".

I am in thorough agreement with Hartree over this. It will be noticed that he does not assert that the machines in question had not got the property, but rather that the evidence available to Lady Lovelace did not encourage her to believe that they had it. It is quite possible that the machines in question had in a sense got this property. For suppose that some discrete-state machine has the property. The Analytical Engine was a universal digital computer, so that, if its storage capacity and speed were adequate, it could by suitable programming be made to mimic the machine in question. Probably this argument did not occur to the Countess or to Babbage.♠ In any case there was no obligation on them to claim all that could be claimed.

This whole question will be considered again under the heading of learning machines.

A variant of Lady Lovelace's objection states that a machine can "never do anything really new".♣ This may be parried for a moment with the saw, "There is

---

♠ Ford, Glymour, and Hayes: Exactly. As many have noted, if we know how it works, we are reluctant to call it intelligent.

♠ Ford, Glymour, and Hayes: To be fair, it is not a very good argument in any case, as the analytical engine had intrinsic speed limitations due to its mechanical construction. Turing here slips a little too quickly between theoretical computability and practical implementability.

♣ Harnad: This is one of the many Granny objections. The correct reply is that (1) all causal systems are describable by formal rules (this is the equivalent of the Church/Turing Thesis), including ourselves; (2) we know from complexity theory as well as statistical mechanics that the fact that a system's performance is governed by rules does not mean we can predict everything it does; (3) it is not clear that anyone or anything has "originated" anything new since the Big Bang.

nothing new under the sun". Who can be certain that "original work" that he has done was not simply the growth of the seed planted in him by teaching, or the effect of following well-known general principles.♠ A better variant of the objection says that a machine can never "take us by surprise". This statement is a more direct challenge and can be met directly. Machines take me by surprise with great frequency. This is largely because I do not do sufficient calculation to decide what to expect them to do, or rather because, although I do a calculation, I do it in a hurried, slipshod fashion, taking risks. Perhaps I say to myself, "I suppose the voltage here ought to be the same as there: anyway let's assume it is". Naturally I am often wrong, and the result is a surprise for me for by the time the experiment is done, these assumptions have been forgotten. These admissions lay me open to lectures on the subject of my vicious ways, but do not throw any doubt on my credibility when I testify to the surprises I experience.

I do not expect this reply to silence my critic. He will probably say that such surprises are due to some creative mental act on my part, and reflect no credit on the machine. This leads us back to the argument from consciousness, and far from the idea of surprise. It is a line of argument we must consider closed, but it is perhaps worth remarking that the appreciation of something as surprising requires as much of a "creative mental act" whether the surprising event originates from a man, a book, a machine, or anything else.

The view that machines cannot give rise to surprises is due, I believe, to a fallacy to which philosophers and mathematicians are particularly subject. This is the assumption that as soon as a fact is presented to a mind all consequences of that fact spring into the mind simultaneously with it. It is a very useful assumption under many circumstances, but one too easily forgets that it is false. A natural

♠ Ford, Glymour, and Hayes: Turing's point applies to many philosophical cavils about machine learning. John Norton, at the University of Pittsburgh, once described a machine that learned Newton's laws from empirical regularities of the solar system such as Kepler's laws. The procedure used simple heuristics that could have been applied in other domains, but perhaps not always successfully. The same could be said of the discovery procedures proposed by Pat Langley, Herb Simon, and their collaborators in their *Scientific Discovery*, 1985, and elsewhere. A common objection is that these programs do not really discover anything, because the context, data, and framework are all kludged – built in by the programmer. Newton and a few others (e.g., Einstein) showed enormous flexibility and inventiveness for constructing new mathematical representations framing families of possible theories, and for inventing heuristics for inference to theories so represented. But even they started with an enormous background of tools they did not invent (as even Newton acknowledged: "I have stood on the shoulders of giants."). Einstein, for example, learned electrodynamics as an undergraduate, and his textbook emphasized the key puzzle behind the special theory of relativity, the induction of current by relative motion of conductor and magnetic field. When he turned to attempts to extend relativity to gravitation he learned differential geometry and field theory and used these tools in a heuristic search, over 8 years and many failed attempts, for a satisfactory theory. The mystification of Einstein by some notable historians notwithstanding, it is not implausible to think a program could simulate Einstein's search for the general theory of relativity.

consequence of doing so is that one then assumes that there is no virtue in the mere working out of consequences from data and general principles.♠♣

3.6.7 *Argument from Continuity in the Nervous System*

The nervous system is certainly not a discrete-state machine. A small error in the information about the size of a nervous impulse impinging on a neuron, may make a large difference to the size of the outgoing impulse. It may be argued that, this being so, one cannot expect to be able to mimic the behaviour of the nervous system with a discrete-state system.♣

It is true that a discrete-state machine must be different from a continuous machine. But if we adhere to the conditions of the imitation game, the interrogator will not be able to take any advantage of this difference. The situation can be made clearer if we consider some other simpler continuous machine. A differential analyser will do very well. (A differential analyser is a certain kind of machine not of the discrete-state type used for some kinds of calculation.) Some of these provide their answers in a typed form, and so are suitable for taking part in the game. It would not be possible for a digital computer to predict exactly what answers the differential analyser would give to a problem, but it would be quite capable of giving the right sort of answer. For instance, if asked to give the value of π (actually about 3.1416) it would be reasonable to choose at random between the values 3.12, 3.13, 3.14, 3.15, 3.16 with the probabilities of 0.05, 0.15, 0.55, 0.19, 0.06 (say). Under these circumstances it would be very difficult for the interrogator to distinguish the differential analyser from the digital computer.

♠ Ford, Glymour, and Hayes: This tendency is evident in several methodological critics of AI. For example, both Jerry Fodor and Hilary Putnam, interviewed in "Speaking Minds" (Baumgartner and Payr, 1996), seem to feel that the "engineering details" of the actual mechanisms of mechanical thinking are of no real interest (Hayes and Ford, 1997).

♣ Harnad: Turing is quite right to point out that knowing something is true does not mean knowing everything it entails; this is especially true of mathematical conjectures, theorems, and axioms.

But I think Lady Lovelace's preoccupation with freedom from rules and novelty is even more superficial than this. It takes our introspective ignorance about the causal basis of our performance capacities at face value, as if that ignorance demonstrated that our capacities are actually sui generis acts of our psychokinetic will – rather than being merely the empirical evidence of our functional ignorance, for future reverse-engineering (cognitive science) to remedy.

♣ Harnad: According to the Church/Turing Thesis, there is almost nothing that a computer cannot simulate, to as close an approximation as desired, including the brain. But, as noted, there is no reason computers should be the only machines eligible for Turing testing. Robots can have analogue components as well. Any dynamical causal system is eligible, as long as it delivers the performance capacity.

3.6.8 The Argument from Informality of Behaviour

It is not possible to produce a set of rules purporting to describe what a man should do in every conceivable set of circumstances.♣ One might for instance, have a rule that one is to stop when one sees a red traffic light, and to go if one sees a green one, but what if by some fault both appear together? One may perhaps decide that it is safest to stop. But some further difficulty may well arise from this decision later. To attempt to provide rules of conduct to cover every eventuality, even those arising from traffic lights, appears to be impossible. With all this I agree.

From this it is argued that we cannot be machines. I shall try to reproduce the argument, but I fear I shall hardly do it justice. It seems to run something like this. "If each man had a definite set of rules of conduct by which he regulated his life he would be no better than a machine. But there are no such rules, so men cannot be machines." The undistributed middle is glaring. I do not think the argument is ever put quite like this, but I believe this is the argument used nevertheless. There may however be a certain confusion between "rules of conduct" and "laws of behaviour" to cloud the issue. By "rules of conduct" I mean precepts such as Stop if you see red lights", on which one can act, and of which one can be conscious. By "laws of behaviour" I mean laws of nature as applied to a man's body such as "if you pinch him he will squeak".♠ If we substitute "laws of behaviour which regulate his life" for "laws of conduct by which he regulates his life" in the argument quoted the undistributed middle is no longer insuperable. For we believe that it is not only true that being regulated by laws of behaviour implies being some sort of machine (though not necessarily a discrete-state machine), but that conversely being such a machine implies being regulated by such laws. However, we cannot so easily convince ourselves of the absence of complete laws of behaviour as of complete rules of conduct. The only way we know of for finding such laws is scientific observation, and we certainly know of no circumstances under which we could say, "We have searched enough. There are no such laws."

♣ HARNAD: First, the successful Turing Test candidate need not be just computational (rule based); all the arguments for T3 robots and their need of real-world sensorimotor capacities, mechanisms, and experience suggest that more is required in a successful candidate than just computation. The impossibility of second-guessing a set of rules that predicts every contingency in advance is probably also behind the "Frame Problem" in (AI) (Harnad, 1993). But it will still be true, because of the Church/Turing Thesis, that the successful hybrid computational/dynamic T3 robot will still be computer-simulable in principle – a virtual robot in a virtual world. So the rule-based system can describe what a T3 robot would do under all contingencies; that simulation would simply not be a T3 robot, any more than its virtual world would be the real world.

♠ FORD, GLYMOUR, AND HAYES: This dichotomy seems, with hindsight, to omit a number of intermediate possibilities, in particular "laws" or "rules" which are purely psychological, but of which we are unconscious.

We can demonstrate more forcibly that any such statement would be unjustified; for suppose, we could be sure of finding such laws if they existed. Then given a discrete-state machine it should certainly be possible to discover by observation sufficient about it to predict its future behaviour, and this within a reasonable time, say a thousand years. But this does not seem to be the case. I have set up on the Manchester computer a small program using only 1,000 units of storage, whereby the machine supplied with one 16-figure number replies with another within 2 s. I would defy anyone to learn from these replies sufficient about the program to be able to predict any replies to untried values.♠

3.6.9 The Argument from Extra-sensory Perception

I assume that the reader is familiar with the idea of extrasensory perception, and the meaning of the four items of it, viz. telepathy, clairvoyance, precognition, and psychokinesis. These disturbing phenomena seem to deny all our usual scientific ideas. How we should like to discredit them! Unfortunately the statistical evidence, at least for telepathy, is overwhelming.♣ It is very difficult to rearrange one's ideas so as to fit these new facts in. Once one has accepted them it does not seem a very big step to believe in ghosts and bogies. The idea that our bodies move simply according to the known laws of physics, together with some others not yet discovered but somewhat similar, would be one of the first to go.

This argument is to my mind quite a strong one. One can say in reply that many scientific theories seem to remain workable in practice, in spite of clashing with E.S.P.; that in fact one can get along very nicely if one forgets about it. This is rather cold comfort, and one fears that thinking is just the kind of phenomenon where E.S.P. may be especially relevant.

A more specific argument based on E.S.P. might run as follows: "Let us play the imitation game, using as witnesses a man who is good as a telepathic receiver, and a digital computer. The interrogator can ask such questions as 'What suit does the

---

♠ Ford, Glymour, and Hayes: This seems to be a similar point to that made earlier about prediction and complexity. Turing has testified that machines often surprise him: Here, he seems to be saying, he has a small machine that will surprise *you*. More seriously, however, Turing's general point here seems to be well supported by the last 50 years of cognitive science. Indeed, there are many "laws of behavior" which seem to apply to the workings of our minds, and of which we are quite unconscious. The many fringe areas of consciousness revealed by studies such as those made popular by the writings of such authors as Vilayanur Ramachandran and Oliver Sacks are also eloquent testimonials to the imperfect nature of our own introspections.

♣ Harnad: It is a pity that at the end Turing reveals his credulousness about these dubious phenomena, for if psychokinesis (mind over matter) were genuinely possible, then ordinary matter/energy engineering would not be enough to generate a thinking mind; and if telepathy (true mind-reading) were genuinely possible, then that would definitely trump the Turing Test.

card in my right hand belong to?' The man by telepathy or clairvoyance gives the right answer 130 times out of 400 cards. The machine can only guess at random, and perhaps gets 104 right, so the interrogator makes the right identification." There is an interesting possibility which opens here. Suppose the digital computer contains a random number generator. Then it will be natural to use this to decide what answer to give. But then the random number generator will be subject to the psychokinetic powers of the interrogator. Perhaps this psychokinesis might cause the machine to guess right more often than would be expected on a probability calculation, so that the interrogator might still be unable to make the right identification. On the other hand, he might be able to guess right without any questioning, by clairvoyance. With E.S.P. anything may happen.

If telepathy is admitted it will be necessary to tighten our test up. The situation could be regarded as analogous to that which would occur if the interrogator were talking to himself and one of the competitors was listening with his ear to the wall. To put the competitors into a "telepathy-proof room" would satisfy all requirements.♠

3.7 Learning Machines♣

The reader will have anticipated that I have no very convincing arguments of a positive nature to support my views. If I had I should not have taken such pains to point out the fallacies in contrary views. Such evidence as I have I shall now give.

Let us return for a moment to Lady Lovelace's objection, which stated that the machine can only do what we tell it to do. One could say that a man can "inject" an idea into the machine, and that it will respond to a certain extent and then drop into quiescence, like a piano string struck by a hammer. Another simile would be an atomic pile of less than critical size: an injected idea is to correspond to a neutron entering the pile from without. Each such neutron will cause a certain disturbance which eventually dies away. If, however, the size of the pile is sufficiently increased,

---

♠ Ford, Glymour, and Hayes: This section reads strangely now, but the conviction that the statistical evidence for telepathy was "overwhelming" was not uncommon at the time. (Compare, e.g., Michael Scriven's similar conclusion only a few years later.) But that conviction was seriously shaken when J. B. Rhine's successor, who tried unsuccessfully to guarantee the reality of extrasensory perception by using a computer to randomly generate targets and score subjects' hits and misses, was found to have jimmied the computer to produce faked positive results. There is a less anecdotal argument: even the tiniest quantum effects, when real, can be promulgated through multipliers to reliably produce big effects (Turing hinted at an example earlier). But no ESP phenomenon, by anyone, has ever been so multiplied (Glymour, 1987).

♣ Harnad: Turing successfully anticipates machine learning, developmental modelling and evolutionary modelling in this prescient section.

the disturbance caused by such an incoming neutron will very likely go on and on increasing until the whole pile is destroyed. Is there a corresponding phenomenon for minds, and is there one for machines? There does seem to be one for the human mind. The majority of them seem to be "sub-critical", i.e., to correspond in this analogy to piles of subcritical size. An idea presented to such a mind will on average give rise to less than one idea in reply. A smallish proportion is supercritical. An idea presented to such a mind may give rise to a whole "theory" consisting of secondary, tertiary, and more remote ideas. Animals minds seem to be very definitely subcritical. Adhering to this analogy we ask, "Can a machine be made to be super-critical?"

The "skin of an onion" analogy is also helpful. In considering the functions of the mind or the brain we find certain operations which we can explain in purely mechanical terms. This we say does not correspond to the real mind: it is a sort of skin which we must strip off if we are to find the real mind. But then in what remains we find a further skin to be stripped off, and so on. Proceeding in this way do we ever come to the "real" mind, or do we eventually come to the skin which has nothing in it? In the latter case the whole mind is mechanical. (It would not be a discrete-state machine however. We have discussed this.)

These last two paragraphs do not claim to be convincing arguments. They should rather be described as "recitations tending to produce belief".♠

The only really satisfactory support that can be given for the view expressed at the beginning of §6, will be that provided by waiting for the end of the century and then doing the experiment described. But what can we say in the meantime? What steps should be taken now if the experiment is to be successful?

As I have explained, the problem is mainly one of programming. Advances in engineering will have to be made too, but it seems unlikely that these will not be adequate for the requirements. Estimates of the storage capacity of the brain vary from 10^{10} to 10^{15} binary digits. I incline to the lower values and believe that only a very small fraction is used for the higher types of thinking.♠ Most of it is probably used for the retention of visual impressions. I should be surprised if more than 10^9 were required for satisfactory playing of the imitation game, at any rate against a blind man. (Note: The capacity of the *Encyclopaedia Britannica*, 11th edition, is 2×10^9.) A storage capacity of 10^7 would be a very practicable possibility even by present techniques. It is probably not necessary to increase the speed of operations of the machines at all.♠ Parts of modern machines which can be regarded as

♠ Ford, Glymour, and Hayes: What Daniel Dennett later referred to as "intuition pumps." It seems that both sides in these debates are often reduced to pumping.

♠ Ford, Glymour, and Hayes: These estimates were based on very little neurological data. The current view is that it may be hard to express the storage capacity of the human brain in such simple terms.

♠ Ford, Glymour, and Hayes: That view now seems unrealistic. It may be that Turing underestimated the computational costs involved in running realistically reliable software simulations. At the time of writing, no one had even attempted to write a program of the complexity of a modern operating system.

analogues of nerve cells work about a thousand times faster than the latter.♠ This should provide a "margin of safety" which could cover losses of speed arising in many ways. Our problem then is to find out how to program these machines to play the game. At my present rate of working I produce about a thousand digits of program a day, so that about 60 workers, working steadily through the 50 years might accomplish the job, if nothing went into the waste-paper basket.♠ Some more expeditious method seems desirable.

In the process of trying to imitate an adult human mind we are bound to think a good deal about the process which has brought it to the state that it is in. We may notice three components:

(a) The initial state of the mind, say at birth
(b) The education to which it has been subjected
(c) Other experience, not to be described as education, to which it has been subjected

Instead of trying to produce a program to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child-brain is something like a notebook as one buys it from the stationer's. Rather little mechanism, and lots of blank sheets.♠ (Mechanism and writing are from our point of view almost synonymous.) Our hope is that there is so little mechanism in the child-brain that something like it can be easily programmed. The amount of

♠ Ford, Glymour, and Hayes: Again, this comment seems naïve in hindsight, although it was one commonly made at that time. Neurons, we now realize, are far more complex than the simple switch-like components that they were once thought to be. Moreover, as the neurologist Valentino Braitenberg has observed, the average connectivity of the mammalian cortex is so high that it would be impossible to physically assemble a brain even if one were given all the neurons and a connection diagram. This does not argue against Turing's basic point, but it does suggest that the complexity estimates might need to be revised.

♠ Ford, Glymour, and Hayes: Turing's optimism about the management of large-scale software engineering projects now seems incredible.

♠ Ford, Glymour, and Hayes: This remark also seems now to be naive, although again it reflects a commonly held view at the time Turing was writing. The range of innate knowledge available to neonates, or occurring automatically in the course of maturation without the application of learning mechanisms, has been the subject of intense research and debate in developmental psychology in recent years. Nativists argue that knowledge of biological distinctions, native physics, and elements of "folk" psychology are innate; non-nativists argue that while some perceptual categorizations and dispositions to imitate may be innate, more sophisticated knowledge is produced by rapid learning mechanisms, so that the internal mental life of even very young children consists of private, and perhaps subconscious, theory construction and revision. Nativist arguments turn on experimental demonstrations of knowledge at ever earlier ages, with the tacit premise that children can not have learned specific ranges of knowledge with the requisite speed. But there has been very little research on learning mechanisms in early childhood. For a discussion of folk psychology, see (Gopnik and Meltzoff, 1996), and for a discussion of hypotheses about early learning mechanisms see (Glymour, 2001).

work in the education we can assume, as a first approximation, to be much the same as for the human child.♥

We have thus divided our problem into two parts. The child-program and the education process. These two remain very closely connected. We cannot expect to find a good child-machine at the first attempt. One must experiment with teaching one such machine and see how well it learns. One can then try another and see if it is better or worse. There is an obvious connection between this process and evolution, by the identifications

Structure of the child machine = Hereditary material
Changes of the child machine = Mutations
Natural selection = Judgment of the experimenter

One may hope, however, that this process will be more expeditious than evolution. The survival of the fittest is a slow method for measuring advantages. The experimenter, by the exercise of intelligence, should be able to speed it up. Equally important is the fact that he is not restricted to random mutations. If he can trace a cause for some weakness he can probably think of the kind of mutation which will improve it.

It will not be possible to apply exactly the same teaching process to the machine as to a normal child. It will not, for instance, be provided with legs, so that it could not be asked to go out and fill the coal scuttle. Possibly it might not have eyes.♥ But however, well these deficiencies might be overcome by clever engineering, one could not send the creature to school without the other children making excessive fun of it. It must be given some tuition. We need not be too concerned about the legs, eyes, etc. The example of Miss *Helen Keller* shows that education can take place provided that communication in both directions between teacher and pupil can take place by some means or other.

♥ SAYGIN: Most neurons can be thought of as having "very little mechanism." The problem is, there is just a lot of that little mechanism in the brain, running in parallel. The way digital computers prefer to process information is fast and serial. The way neurons do it is slower but massively parallel. Almost anything that is worth talking about in terms of "thought" arises out of some mysterious interaction of all these little mechanical events, the sum not just bigger, but somehow also different than the parts. Shall we model the mechanism literally and hope intelligence emerges, or shall we try to figure out meaningful "chunks" in their operation and encapsulate them to represent them the way digital computers prefer it? The former becomes computationally intractable for large numbers of neurons. The latter is prone to errors of interpretation and representation at several levels. Both approaches are being tried in current research, but it is unlikely that either approach alone will work for modeling a substantial component of human thought or behaviour.

♥ SAYGIN: To me, this is the only place in this paper where Turing is clearly wrong. Perceiving, sensing, and acting upon the environment is what knowledge is built upon (e.g., Barsalou, 2000). It might be the case that having some type of sensor and body with which to experience and act upon objects and events is necessary to be able to play the imitation game well (e.g., as argued by Harnad, 1990).

We normally associate punishments and rewards with the teaching process. Some simple child-machines can be constructed or programmed on this sort of principle. The machine has to be so constructed that events which shortly preceded the occurrence of a punishment-signal are unlikely to be repeated, whereas a reward-signal increased the probability of repetition of the events which led up to it. These definitions do not presuppose any feelings on the part of the machine.♥ I have done some experiments with one such child-machine, and succeeded in teaching it a few things, but the teaching method was too unorthodox for the experiment to be considered really successful.

The use of punishments and rewards can at best be a part of the teaching process. Roughly speaking, if the teacher has no other means of communicating to the pupil, the amount of information which can reach him does not exceed the total number of rewards and punishments applied. By the time a child has learnt to repeat "Casabianca" he would probably feel very sore indeed, if the text could only be discovered by a "Twenty Questions" technique, every "NO" taking the form of a blow. It is necessary therefore to have some other "unemotional" channels of communication. If these are available it is possible to teach a machine by punishments and rewards to obey orders given in some language, e.g., a symbolic language. These orders are to be transmitted through the "unemotional" channels. The use of this language will diminish greatly the number of punishments and rewards required.♠

Opinions may vary as to the complexity which is suitable in the child machine. One might try to make it as simple as possible consistently with the general principles.♠ Alternatively one might have a complete system of logical inference "built in". In the latter case the store would be largely occupied with definitions and propositions. The propositions would have various kinds of status, e.g., well-established facts, conjectures, mathematically proved theorems, statements given by an authority, expressions having the logical form of proposition but not belief-value. Certain propositions may be described as "imperatives". The machine

♥ SAYGIN: Interesting point… many philosophers of mind talk about beliefs, desires, and emotions as complex and "human" traits. However, emotional systems are one of the better understood systems in neuroscience and are not all that complex or opaque compared with many others. If our emotions, more or less, boil down to levels of a handful of chemicals, it may not be all that far-fetched to just say that a computer "feels" those emotions it is programmed to feel.

♠ FORD, GLYMOUR, AND HAYES: This passage seems oddly out of place. One wonders if Turing here accidentally strayed into reminiscence of his own schooldays for a while.

♠ FORD, GLYMOUR, AND HAYES: The basic idea of making a simple "child machine" which can learn about its own environment was a common trope in early AI. It is probably safe to say that the optimism expressed here has not been borne out in practice. The learning process gone through during a human childhood is extremely complex and still not well understood, but it is certainly not a simple matter of assembling conditioned reflexes under the influence of positive and negative feedbacks. Even language learning, for example, seems to involve very intricate built-in processes that are supplied by genetics.

should be so constructed that as soon as an imperative is classed as "well- established" the appropriate action automatically takes place. To illustrate this, suppose the teacher says to the machine, "Do your homework now". This may cause "Teacher says 'Do your homework now'" to be included amongst the well-established facts. Another such fact might be, "Everything that teacher says is true". Combining these may eventually lead to the imperative, "Do your homework now," being included amongst the well-established facts, and this, by the construction of the machine, will mean that the homework actually gets started, but the effect is very satisfactory. The processes of inference used by the machine need not be such as would satisfy the most exacting logicians. There might for instance be no hierarchy of types. But this need not mean that type fallacies will occur, any more than we are bound to fall over unfenced cliffs. Suitable imperatives (expressed *within* the systems, not forming part of the rules *of* the system) such as "Do not use a class unless it is a subclass of one which has been mentioned by teacher" can have a similar effect to "Do not go too near the edge".

The imperatives that can be obeyed by a machine that has no limbs are bound to be of a rather intellectual character, as in the example (doing homework) given above. Important amongst such imperatives will be ones which regulate the order in which the rules of the logical system concerned are to be applied. For at each stage when one is using a logical system, there is a very large number of alternative steps, any of which one is permitted to apply, so far as obedience to the rules of the logical system is concerned. These choices make the difference between a brilliant and a footling reasoner, not the difference between a sound and a fallacious one. Propositions leading to imperatives of this kind might be "When Socrates is mentioned, use the syllogism in Barbara" or "If one method has been proved to be quicker than another, do not use the slower method". Some of these may be "given by authority", but others may be produced by the machine itself, e.g., by scientific induction.♠

The idea of a learning machine may appear paradoxical to some readers. How can the rules of operation of the machine change? They should describe completely how the machine will react whatever its history might be, whatever changes it might undergo. The rules are thus quite time-invariant. This is quite true. The

♠ Ford, Glymour, and Hayes: This remarkable passage clearly outlines the research program that has been referred to as "good old-fashioned AI" (GOFAI) by John Haugeland; less derisively, it is often called "mainstream AI": the use of an explicit knowledge representation formalism which is manipulated by reasoning engines and linked to actions which are triggered by particular inference patterns. The details differ, but the same basic paradigm underlies McCarthy's logical reasoner-based research program, all of theorem-proving and computational logic, the production-system-based style of psychological modelling pioneered and developed by Simon and Newell, together with its successors, and most work in AI planning and natural language comprehension (Russell and Norvig, 2002). While the adequacy of this paradigm as a basic model for cognitive science is still controversial, the overall success of this research program is beyond dispute; most of these ideas are now part of mainstream computer science and engineering.

explanation of the paradox is that the rules which get changed in the learning process are of a rather less pretentious kind, claiming only an ephemeral validity. The reader may draw a parallel with the Constitution of the USA.

An important feature of a learning machine is that its teacher will often be very largely ignorant of quite what is going on inside, although he may still be able to some extent to predict his pupil's behaviour. This should apply most strongly to the later education of a machine arising from a child-machine of well-tried design (or program). This is in clear contrast with normal procedure when using a machine to do computations: one's object is then to have a clear mental picture of the state of the machine at each moment in the computation. This object can only be achieved with a struggle. The view that "the machine can only do what we know how to order it to do", appears strange in face of this. Most of the programs which we can put into the machine will result in its doing something that we cannot make sense of at all, or which we regard as completely random behaviour. Intelligent behaviour presumably consists in a departure from the completely disciplined behaviour involved in computation, but a rather slight one, which does not give rise to random behaviour, or to pointless repetitive loops. Another important result of preparing our machine for its part in the imitation game by a process of teaching and learning is that "human fallibility" is likely to be omitted in a rather natural way, i.e., without special "coaching". (The reader should reconcile this with the point of view [developed early in this essay].) Processes that are learnt do not produce a 100% certainty of result; if they did they could not be unlearnt.

It is probably wise to include a random element in a learning machine (see p. 438). A random element is rather useful when we are searching for a solution of some problem. Suppose, for instance, we wanted to find a number between 50 and 200 which was equal to the square of the sum of its digits, we might start at 51 then try 52 and go on until we got a number that worked. Alternatively we might choose numbers at random until we got a good one. This method has the advantage that it is unnecessary to keep track of the values that have been tried, but the disadvantage that one may try the same one twice, but this is not very important if there are several solutions. The systematic method has the disadvantage that there may be an enormous block without any solutions in the region which has to be investigated first. Now the learning process may be regarded as a search for a form of behaviour which will satisfy the teacher (or some other criterion). Since there is probably a very large number of satisfactory solutions the random method seems to be better than the systematic.♠ It should be noticed that it is used in the analogous process of evolution. But there the systematic method is not possible. How could one keep track of the different genetical combinations that had been tried, so as to avoid trying them again?

We may hope that machines will eventually compete with men in all purely intellectual fields. But which are the best ones to start with? Even this is a difficult

♠ Ford, Glymour, and Hayes: Indeed, so-called Markov methods which search large spaces starting from random points have proven extremely successful in many applications, and are now routinely used in AI and search processes more generally.

decision. Many people think that a very abstract activity, like the playing of chess, would be best.[♠] It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English.[♥] This process could follow the normal teaching of a child. Things would be pointed out and named, etc. Again I do not know what the right answer is, but I think both approaches should be tried.[♠][♥]

We can only see a short distance ahead, but we can see plenty there that needs to be done.[♠]

[♠] Ford, Glymour, and Hayes: As is well-known, early AI work followed Turing's suggestion. Checkers (draughts) was an early success in machine learning, when Samuels' program rapidly learned to outplay its creator. Since then, of course, Deep Blue has become the world's best chess player, and many personal computers can outplay most human amateurs. However, one can ask whether this effort has been as meaningful as its early enthusiasts felt it was. Certainly it seems to have been partly responsible for a rather unhealthy attitude towards AI in which it is assumed that the only purpose of making machines with intellectual capacity is to somehow beat human beings at some game or other. Turing's imitation game itself also has this character, of course. It would be a pity if this emphasis on needless competition were to obscure the far more useful, and we think ultimately far more interesting, goal of making machines which extend, rather than replace, human cognitive and intellectual powers. Most applied AI is in fact devoted to making systems that can be best described as servants, aids, or even cognitive prosthetics to humans, rather than artificial competitors. In beating Kasparov, Deep Blue was not attacking humanity. In fact, a better way to characterize the situation is to say that Deep Blue is a tool with which anyone, even a child, could be world chess champion. The winner, ultimately, is the person moving the chess pieces, no matter what kind of machine he or she is using.

[♥] Saygin: I have always found it extremely interesting that Turing mentions "chess" and "buying the best sense organs money can buy and teach it to understand and speak English." The former is a completely disembodied task and computers have come to perform it rather well. The latter is done effortlessly by infants across the world but has proven to be very difficult to model on computers. But, we did not follow Turing's advice, we did not buy the best sense organs and let the machine acquire English – or any other language for that matter.

[♠] Ford, Glymour, and Hayes: This is one of the most original sections of a very original paper, and perhaps the part that has drawn the least positive response. Turing proposes nothing less than a behavioural simulacrum of human cognitive development, motor activity aside. To the best of our knowledge, nothing like it has been attempted, and the computational theories that might be applied to the task are only beginning to emerge. Until very recently, developmental psychologists did not take up Turing's implicit challenge to describe algorithms by which children learn the myriad features and regularities of the everyday world. That is beginning to change, especially with the work of Alison Gopnik and her collaborators on young children's procedures for learning causal relationships (Gopnik and Meltzoff, 1997). But one might wonder: if we could program a computer to develop the knowledge of the world, the capacity for recognition, classification, categorization, prediction, learning, and control exhibited in the course of development by a normal human child; if we could do that, would we need the imitation game to convince ourselves that such a marvellous computer thinks?

[♥] Saygin: Again, so is nature. For instance, the gist of the idea of "eye" has been around in evolution in even very simple early organisms.

[♠] Ford, Glymour, and Hayes: Again, a call to arms! It seems clear that Turing hoped his bold conjecture would motivate research and interest. Certainly Turing could see further ahead than most.

Chapter 4
Commentary on Turing's "Computing Machinery and Intelligence"

John Lucas

Turing's aim was to refute claims that aspects of human intelligence were in some mysterious way superior to the Artificial Intelligence (AI) that Turing machines might be programmed to manifest. He sought to do this by proposing a conversational test to distinguish human from AI, a test which, he claimed, would, by the end of the 20th century, fail to work. And, it must be admitted, it often does fail – but not because machines are so intelligent, but because humans, many of them at least, are so wooden. The underlying question is about the limits of "algorithmic intelligence"', whether all reasoning is in accordance with some rule or other – whether, that is, to be reasonable is to be acting in accordance with a rule – or whether some exercises of reason go beyond anything covered by antecedent rules. But whether or not this is so, there are many people, bureaucrats, legal clerks, accountants, who are entirely rule-governed, and take care never to do or say anything unless it is in accordance with the rule-book. Turing's Test would classify them with the artificial algorithmic intelligences, not because they were artificial, but because their responses were mechanical.

It is a distinction we are familiar with in ordinary social life. Often we find ourselves having entirely predictable conversations with people who invariably say the correct thing and utter conventional opinions and manifest standard responses; but occasionally we meet someone who has interesting ideas and says things which we had not thought of but which we immediately recognize as right and fitting. Turing parries this variant of Lady Lovelace's objection [p. 450] (p. 21) {p. 56} by suggesting that "There is nothing new under the sun", and that all his thoughts are really unoriginal. But the objection lacks force, as Turing himself admits: "I do not expect this reply to silence my critic. He will say that <they>…are due to some creative mental act…" [p. 451] (pp. 21–22) {p. 57}. But the crucial point is that we do make the distinction, whether or not we sometimes misapply it. We distinguish conversation with Turing's critic, who has a mind of his own, and, when we introduce a topic, can "go on" making fresh apposite points from conversation with someone who produces only programmed responses with nothing individual or original about them. We have the concept of nonalgorithmic intelligence.

Merton College, Oxford

R. Epstein et al. (eds.), *Parsing the Turing Test.*

Turing says that the argument from creativity leads back to the argument from consciousness, which he considered closed, since those who support it are committed, whether they realize it or not, to solipsism. It was a point easily made in 1950 against the background of the then dominant Verificationist theory of meaning. But meaning is not constituted by the method of verification. Many understand Fermat's Last Theorem, though few can fathom Andrew Wiles' proof. The tests of whether a person is conscious are one thing, what it means to say of a person is another. Meaning is a matter not of tests, but of entailment patterns, of what follows from the ascription or is inconsistent with it. It would be inconsistent of me to say that you were in great pain, and go on to assert that you were as happy as happy can be; rather, I should show sympathy, and not expect you to be able to think hard about peripheral matters. The nightmarish case of a person paralyzed by curare, yet conscious while an operation is performed under an ineffective anesthetic shows how different the concept of consciousness is from the criteria for its ascription. It is generally characteristic of consciousness and mental concepts that, though we often have good grounds for ascribing them, our ascriptions are subject to subsequent withdrawal. It is the same with truth. We often have good grounds for holding that something is true, and quite often are right in doing so, but, apart from some empty tautologies, live with the perpetual possibility of being wrong. This shows that Turing's Test is much less definitive than he thought. Its logic is not the simple, clear logic of deductive argument, but the messier "dialectical" logic of prima facie arguments and counterarguments, of objections and rebuttals, inconclusive arguments, and conclusions subject to "other things being equal" clauses, and the possibility of our having later to emend them. It does not follow that Turing's Test is not good, but it does follow that its application is more difficult, and may involve wider considerations than a simple exchange of conversational gambits.

One feature of consciousness is that a conscious being can be the subject of its own thought. Turing complains that no evidence was offered for this claim, but it seems true, and I think that it opens the door, when we come to think about our own rationality, to certain sorts of reflexive thought and self-referring argument of great importance.

Turing is dismissive of the "Heads in the Sand" objection, when the consequences of mechanism are considered and found to be too dreadful. But although we have to be prepared to discover that things are as they are and their consequences will be what they will be, there are good reasons for being chary of throwing over established modes of thought too easily. They may have much going for them, and often have been tried over many generations, and found to be reliable. In particular, we should be chary of throwing over the idea of rationality itself. If some theory has as a consequence that we cannot trust our intimations of rationality, then we may well be skeptical of the reasoning that leads us to adopt that theory. It is a very general test of a metaphysical system: what account does it give of itself? Does it cut the ground from underneath the considerations that might incline us to accept it? On an autobiographical note, it was considerations of this sort that first led me to think about the self-referential paradoxes of reductive accounts of human reasoning, and ultimately to Gödel's theorem as encapsulating the principle of self-reference in a rigorous way.

Turing allows that there are limitations to algorithmic intelligence, but resists the conclusion that human intelligence is therefore superior. Although Gödel and Turing proved their own theorems, each using principles of inference that went beyond those laid down for the system they were studying, it might be that each was actually an instantiation of some stronger system of algorithmic reasoning. After all, once some original move has been recognized as a right one, it becomes possible to encapsulate it in some definitely formulated rule. It has often happened in the history of the creative arts. Novelty in music, in painting, in literature, is first recognized as original, then generally accepted and copied, and then systematized and standardized, and finally becomes vieux jeu. So seeming novelty in human intelligence might be algorithmic in some wider system after all; even if not already algorithmic, there would be some machine that could be built incorporating the apparently novel move. So "our superiority can only be felt on such an occasion in relation to the one machine over which we have secured our petty triumph. There can be no question of triumphing simultaneously over all machines. In short, then, there might be men cleverer than any given machine, but then there might be other machines cleverer again, and so on." [p. 445] (p. 16) {p. 52}.

These objections were ones I found it difficult to overcome when I was thinking out my "Minds, Machines and Gödel" (Lucas, 1961). I overcame the first by considering the purported mechanical model of the human's own mind and I neutralized the second by following the "and so on" up into the transfinite. Douglas Hofstadter (1979) is not sure whether the foray into the transfinite secures or refutes my argument, and opines that it refutes it because of the Church-Kleene theorem that "There is no recursively related notation system which gives a name to every constructive ordinal", which means in the case of Turing's contest between an algorithmic machine and a human mind "that no algorithmic method can tell how to apply the method of Gödel to all possible kinds of formal system". But the absence of such an algorithmic method is crippling only to an algorithmic intelligence. Only if the human mind were an algorithmic intelligence would it be unable to keep up the pressure as the contest ascended through ever higher transfinite ordinals. If the mind can understand Gödel's theorem, as it seems it can, then it will be able to apply it in novel circumstances not covered by any rule-book, and so outgun an algorithmic machine however ordinarily complex its Gödelizing operator is.

References

Hofstadter, D., 1979, *Gödel, Escher, Bach: An Eternal Golden Braid*, Basic Books, New York, pp. 475–476.

Lucas, J., 1961, Minds, machines and Gödel, *Philosophy* XXXVI: 112–127; reprinted in 1963, *The Modeling of Mind*, K. M. Sayre and F. J. Crosson, eds., Notre Dame Press, pp. 269–270; and 1954, *Minds and Machines*, Anderson, A. R., ed., Prentice-Hall, pp. 43–59.

Turing, A. M., 1950, Computing machinery and intelligence, *Mind* **59**: 433–460; reprinted in Anderson, A. R., 1964, *Minds and Machines*, Englewood Cliffs, pp. 4–30; Page references to this version are in parentheses, thus (p. 16.); also in 1990, *The Philosophy of Artificial Intelligence*, M. Boden, ed., Oxford University Press; Page references to this version are in round brackets, thus {p. 52}.

Part II
The Ongoing Philosophical Debate

Chapter 5
The Turing Test

Mapping and Navigating the Debate

Robert E. Horn

Abstract The structure of the Turing Test debates has been diagrammed into seven large posters containing over 800 major claims, rebuttals, and counterrebuttals. This "mapping" of the debates is explained and discussed.

Keywords Argumentation maps, can computers think, debates, Turing Test

5.1 Background to the Debate

The Turing debate, as to whether computers will ever be able to think, is one of the great philosophical debates of recent times. It focuses on what it means to be a human being. It lies at the foundation of cognitive science. It has great practical consequences to society and what our communities shall become (Kurzweil, 1999; Moravec, 1999). The Turing debate is – as yet – unresolved.

The debate was initiated by Alan Turing in his article in *Mind* (Turing, 1950). Turing said, "I believe that at the end of the century [i.e., 2000] … one will be able to speak of machines thinking without expecting to be contradicted." Turing's article has unleashed five decades of debate about the many aspects of this proposition. Over 800 major "moves" in the argument later, we are still debating the subject. I say 800 major moves in the argument because that is the number we came up with after diagramming the important claims, rebuttals, and counterrebuttals of this argument.

5.2 The Problem of Comprehending a Large Debate

A debate this large and sprawling, carried on by over 400 scholars, researchers, and scientists worldwide from at least ten academic disciplines, is difficult for the human mind to comprehend. It is a gigantic knowledge management problem. But,

Stanford University and Saybrook Graduate School

R. Epstein et al. (eds.), *Parsing the Turing Test*.

over the centuries humans have created a whole visual language of large diagrams and maps to help us see the structure, organization, and dynamics of complex objects of study (Horn, 1998b). When we must master the complex intricacies of debates such as those concerned with Turing's claim, we must have knowledge maps to help us navigate over the landscape of these arguments. We need to be able to individuate the major claims from the oceans of prose that surround them. We need to isolate the principal data, experiences, or other grounds that support the arguments. Above all, we need to be able to know if the claims offered by each protagonist have been rebutted, perhaps utterly destroyed, by subsequent moves in the debate. We need to know who these protagonists are and from what worldview they are presenting their arguments. All this needs to have some kind of graphical structure – some map-like presentation – so that we can scan and recognize areas of investigation as easily as using a roadmap. We need to take advantage of the capacity of visual tools to articulate the important relationships in the debate in ways that prose simply cannot do.

If we do not have a map, our debates will become increasingly lost in detours and we are left spinning our wheels in muddy ditches; our journey along the road of philosophy will be slow and frustrating; the foundations of the new cognitive science will be shaky; our view of our technology future will be clouded.

5.3 Our Approach: Argumentation Mapping

Recently, our team published an argumentation mapping approach to the Turing debate (Horn, 1998a). This set of maps is part of a larger series that is intended to create visual navigational tools for intellectual history. The "maps" are large diagrams, the core of which connects claims, rebuttals, and counterrebuttals together so that a person unfamiliar with the debates has a convenient way of seeing the structure of the debates as well as the detailed arguments. Figure 5.1 presents an example of one of the seven argumentation maps in the series about the Turing Test debates discussed in this book. Together, the seven maps (each of them 3 × 4 ft in size) provide a complete framework and context for the Turing Test debates. Figures 5.3–5.9 provide close-up views of the details of the mapping of arguments taken from the maps.

5.3.1 Subarguments Show Major "Issue Areas" of the Debates

We divided – analytically and visually – the debates into some 70 subarguments, listed in Fig. 5.2. Each provided an easily recognizable region of the maps.

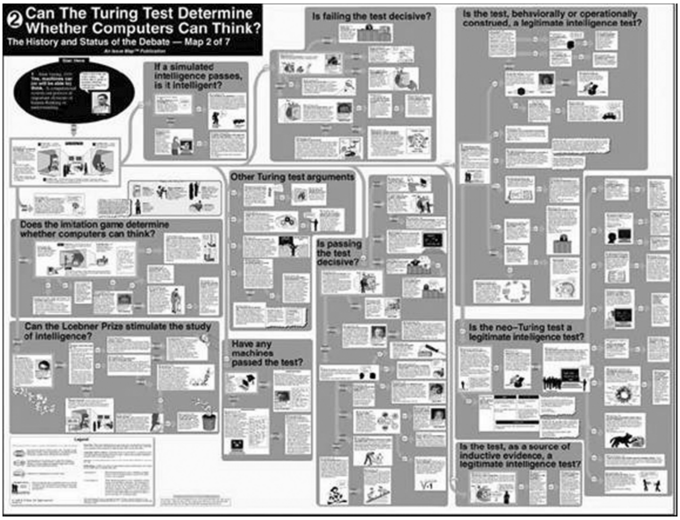

Fig. 5.1 Can the Turing Test determine whether computers can think?

5.4 Parts of the Debate Covered in this Chapter

Our maps provide a way of looking at the debate as it stood in late 1997 when our final prepublication work was done. As with any geographical map, the territory changes over time. Thus our maps may be slightly behind the times, although we believe that the major topography of the debates remains.

The discussion in this chapter will cover the following topics from Map 2:

- Is the imitation game adequate?
- Is failing the Turing Test decisive?
- Is passing the Turing Test decisive?
- Have any machines passed the Turing Test?

5.5 Questions not Covered in this Chapter

Other topics on Map 2 cannot be covered in this chapter because of space limitations:

- Is the Turing Test, behaviorally or operationally construed, a legitimate intelligence test?
- Is the Turing Test, as a source of inductive evidence, a legitimate intelligence test?
- Is the neo-Turing Test a legitimate intelligence test?
- Can the Loebner Prize stimulate the study of intelligence?

Fig. 2 The subarguments of the basic question "Can Computers Think?"

Over 70 questions on specific issues are addressed on the 7 Maps. The argumentation maps summarize the following issues

Map 1: Can computers think?
Can computers have free will?
Can computers have emotions?
Can computers be creative?
Can computers understand arithmetic?
Can computers draw analogies?
Can computers be persons?
Is the brain a computer?
Can computers reason scientifically?
Are computers inherently disabled?
Should we pretend that computers will never be able to think?
Does God prohibit computers from thinking?

Map 2: Can the Turing test determine whether computers can think?
Is failing the test decisive?
Is passing the test decisive?
If a simulated intelligence passes, is it intelligent?
Have any machines passed the test?
Is the test, behaviorally or operationally construed, a legitimate intelligence test?
Is the test, as a source of inductive evidence, a legitimate intelligence test?
Is the neo-Turing test a legitimate intelligence test?
Does the imitation game determine whether computers can think?
Can the Loebner Prize stimulate the study of intelligence?
Other Turing test arguments

Map 3: Can physical symbol systems think?
Does thinking require a body?
Is the relation between hardware and software similar to that between human brains and minds?
Can physical symbol systems learn as humans do?
Can the elements of thinking be represented in discrete symbolic form?
Can symbolic representations account for human thinking?
Does the situated action paradigm show that computers can't think?
Can physical symbol systems think dialectically?
Can a symbolic knowledge base represent human understanding?
Do humans use rules as physical symbol systems do?
Does mental processing rely on heuristic search?
Do physical symbol systems play chess as humans do?
Other physical symbol systems arguments

Figure 2. The subarguments of the basic question "Can Computers Think?"
Reproduced by permission of MacroVU Press.
www.macrovu.com

Map 4: Can Chinese Rooms think?
Do humans, unlike computers, have intrinsic intentionality?
Is biological naturalism valid?
Can computers cross the syntax–semantics barrier?
Can learning machines cross the syntax–semantics barrier?
Can brain simulators think?
Can robots think?
Can a combination robot/brain simulator think?
Can the Chinese Room, considered as a total system, think?
Do Chinese Rooms instantiate programs?
Can an internalized Chinese Room think?
Can translations occur between the internalized Chinese Room and the internalizing English speaker?
Can computers have the right causal powers?
Is strong AI a valid category?
Other Chinese Room arguments

Map 5, Part 1: Can connectionist networks think?
Are connectionist networks like human neural networks?
Do connectionist networks follow rules?
Are connectionist networks vulnerable to the arguments against physical symbol systems?
Does the subsymbolic paradigm offer a valid account of connectionism?
Can connectionist networks exhibit systematicity?
Other connectionist arguments

Map 5, Part 2: Can computers think in images?
Can images be realistically represented in computer arrays?
Can computers represent the analogue properties of images?
Can computers recognize Gestalts?
Are images less fundamental than propositions?
Is image psychology a valid approach to mental processing?
Are images quasi-pictorial representations?
Other imagery arguments

Map 6: Do computers have to be conscious to think?
Can computers be conscious?
Is consciousness necessary for thought?
Is the consciousness requirement solipsistic?
Can higher-order representations produce consciousness?
Can functional states generate consciousness?
Does physicalism show that computers can be conscious?
Does the connection principle show that consciousness is necessary for thought?

Map 7: Are thinking computers mathematically possible?
Is mechanistic philosophy valid?
Does Gödel's theorem show that machines can't think?
Does Gödel's theorem show that machines can't be conscious?
Do mathematical theorems like Gödel's show that computers are intrinsically limited?
Does Gödel's theorem show that mathematical insight is non-algorithmic?
Can automata think?
Is the Lucas argument dialectical?
Can improved machines beat the Lucas argument?
Is the use of consistency in the Lucas argument problematic?
Other Lucas arguments

Figure 2. The subarguments of the basic question "C an Computers Think?"

Fig. 5.2 The subarguments of the basic question, “Can computers think?”

5.6 Wider Debates Covered in Other Maps

Because of the limited parameters of this book, we also cannot review herein the wider debates that Turing’s claim has provoked. These revolve around the von Neuman (Map 3) and Connectionist (Map 5) computer architectures, the clashes about consciousness and computing (Map 6), and the brilliant debate about whether, at bottom, thought is somehow visual and, hence, in order to think machines must be visual (Map 5). This chapter also cannot cover in detail the extraordinary 10-year combat about the Chinese Room (Map 4) that asks if syntax alone can produce semantics. Each of these sub-debates average close to 100 major "moves" each and their own history and structure. The reader who is interested in such sub-questions as those concerned with the Gödel or Penrose debates, the Chinese Room arguments, or the Heidiggerian perspectives advanced by Dreyfus will have to consult Maps 7, 4, and 3 respectively. No doubt, many of these issues will be debated in this book. It can be examined in the seven maps whether the authors address the previous major moves in the argument. These wider issues, of course, contradict Turing's denial that people might achieve wide agreement about how thinking might be assessed by the use of his test.

5.7 Important Possibilities that Argumentation Mapping Provides

This chapter will thus provide an overview of the structure of a central part of the argument and, in addition, provide a way for the reader of this book to begin to answer several important questions:

- What *new* lines of argument have developed since our series of maps was published and how do these new debates fit into the overall structure of the debate so far?
- What lines of argument have been *extended* by new rebuttals, counterrebuttals, or evidence? (The argumentation maps, in addition to displaying the intellectual history of the sidebars, provide an easy way of seeing where the debates have stopped or paused. You can read the last rebuttal in each thread of the argument on the right-hand side of each thread.)
- What new ways of *framing* the arguments have been offered since then?
- What claims or rebuttals have *not* been replied to? (And what can we infer about *that*?)

5.8 Understanding the Turing Test Itself

Let us now investigate in detail the structure of the arguments by examining some of the core questions. To do this, one must first understand the test itself. Turing starts by envisioning a guessing game played by a woman and a man as described in Fig. 5.3. In our argumentation mapping approach, if something can be explained by a combination of words and pictures better than by words alone, we provide a visual illustration.

This imitation game sets up the framework for the test. But Turing is interested in computers, so he replaces one of these humans with a computer and makes his claim that this test will provide an operational definition of "thinking" (see Fig. 5.4).

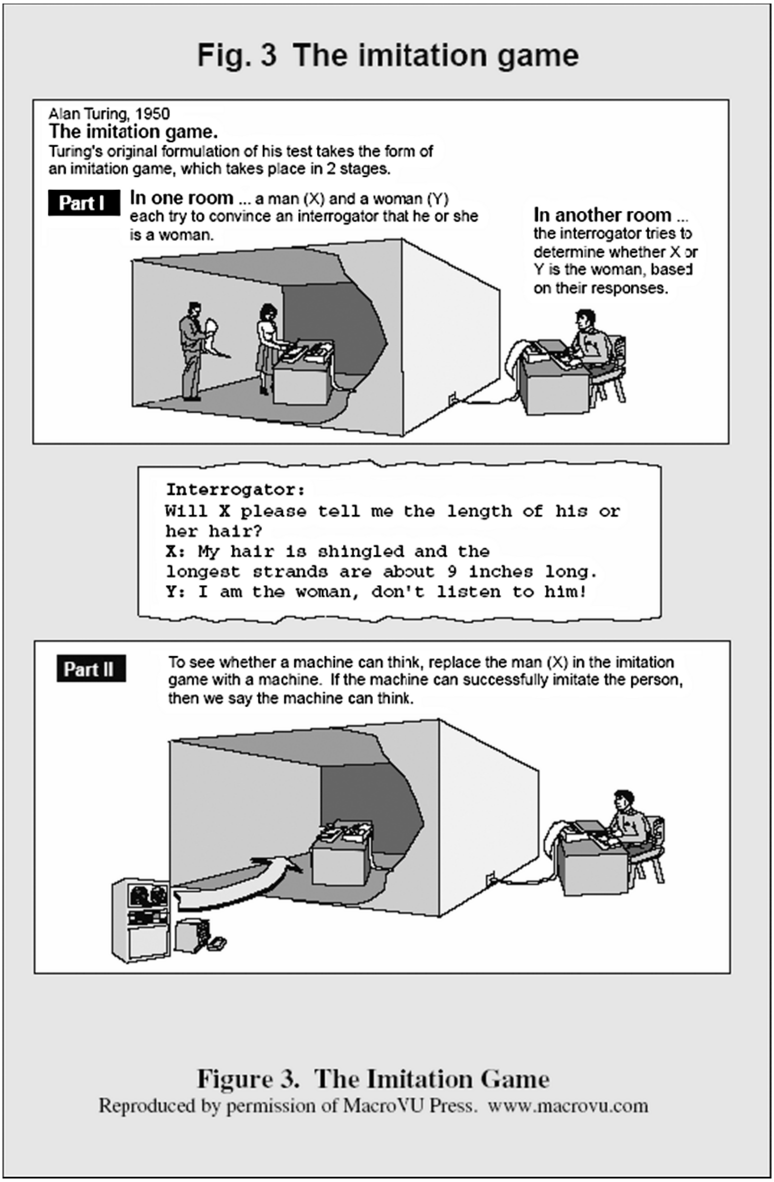

Fig. 5.3 The imitation game

This is the central situation about which so much has been debated and about which this book revolves. Because there are so many interpretations of the test based on so many worldviews, the debates have gone off in many directions. It can be argued that the test will never be conclusive because test results always have to

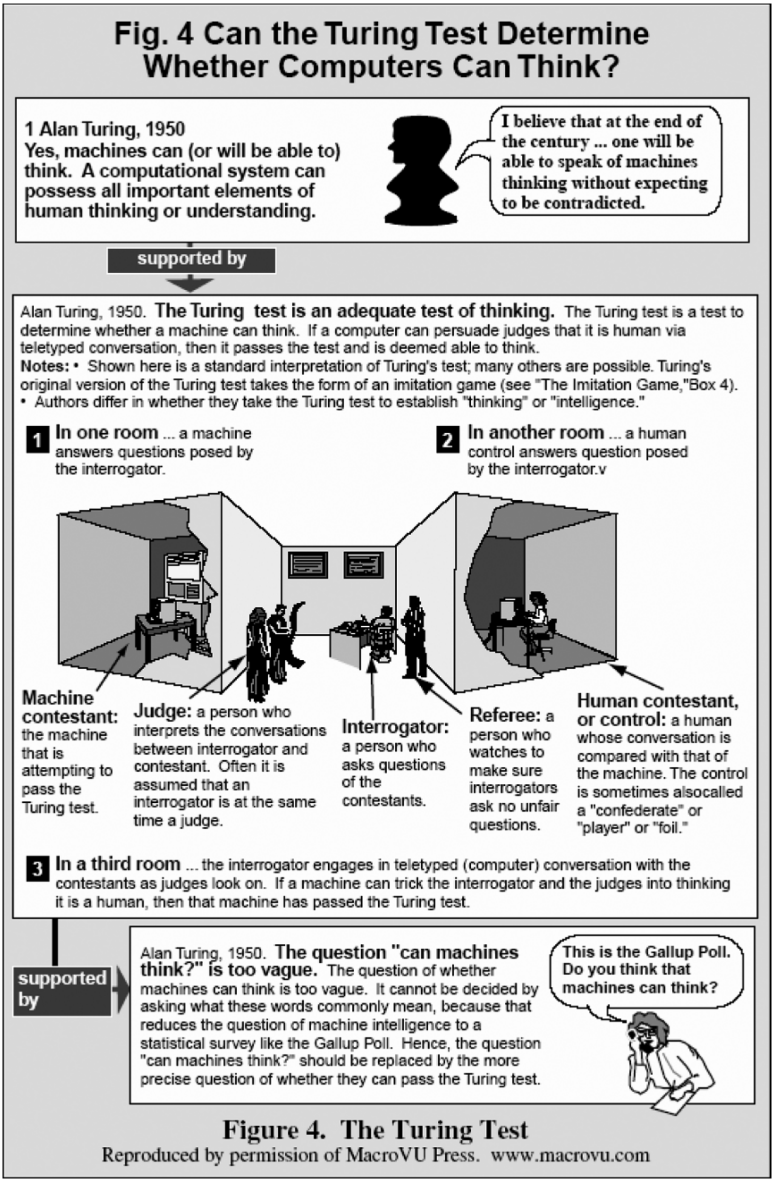

Figure 4. The Turing Test

Reproduced by permission of MacroVU Press. www.macrovu.com

Fig. 5.4 Can the Turing Test determine whether computers can think?

be interpreted and interpretations always are subject to further debate. This could be argued, but is not an argument that appears on our argumentation maps because at the time of publication, we did not find it argued explicitly by any of the protagonists!

5.8.1 *Does the Imitation Game Help Determine Whether Computers Can Think?*

The first debate is on the structure of the debate itself, on the imitation game (see Fig. 5.5).

How are we to assess this path of the debate?

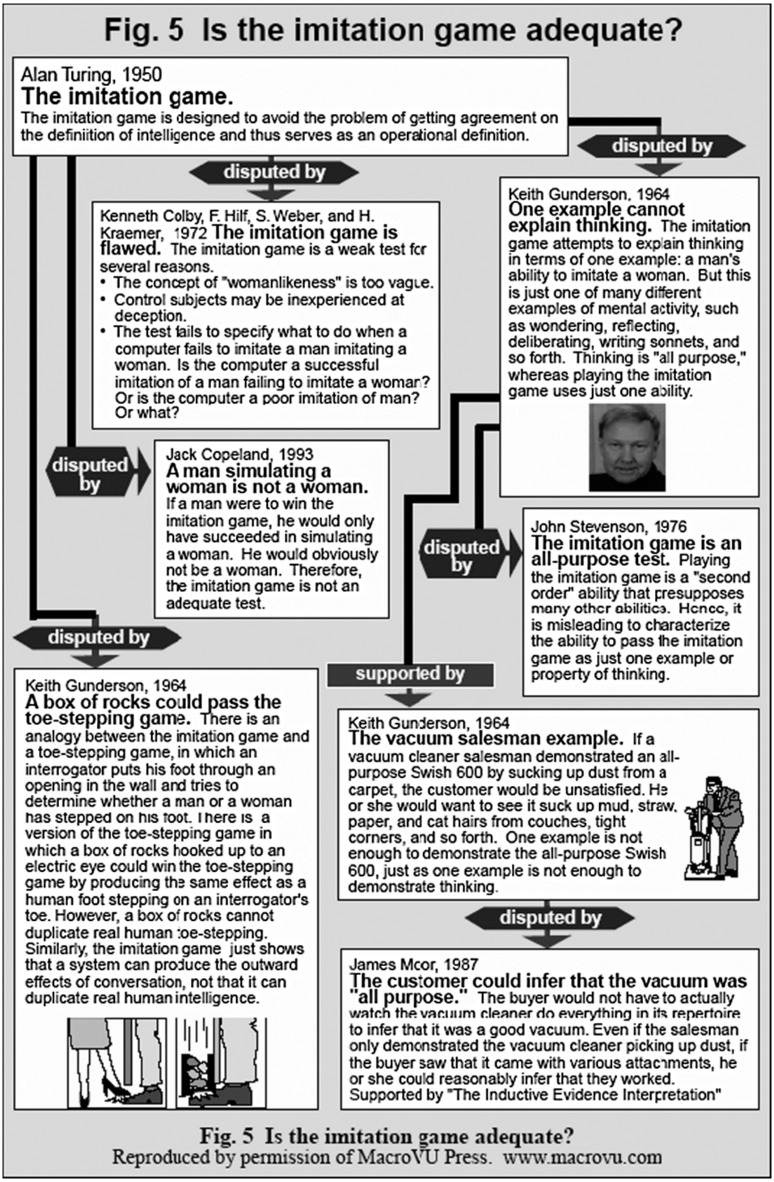

Fig. 5 Is the imitation game adequate?
Reproduced by permission of MacroVU Press. www.macrovu.com

Fig. 5.5 Is the imitation game adequate?

5.8.1.1 Assessment of the Debate So Far

The map tells us immediately that there are provocative and unanswered critiques. How? One can read the unanswered arguments from the end of the diagram paths. Why are they unanswered? Perhaps it is because others have evaluated the claims as not important enough to waste time on. After all, several of the claims are quite old. Perhaps they have simply been missed because they appear in a book or article now out of print. Or, perhaps, someone has answered these claims and our research did not find their rebuttals. Whatever the case, one of the benefits of the mapping approach is to identify unanswered rebuttals. They offer the opportunity for entry into the debate at very precise points. And they point to potentially key aspects of the debate: If these claims go unanswered, is the attempt to base the computer version of the imitation game also ungrounded?

5.8.2 After Turing's Claim, How Has the Debate Proceeded?

The two most important questions about the adequacy of the Turing Test are: "Is failing the test decisive?" and "Is passing the test decisive?" If the test is not adequate in these respects, the rest of the debate about the test is limited. In the remainder of this chapter, I will present some sections of our maps on the subsequent 50 years of the debate.

5.8.3 Is Failing the Test Decisive?

The answer to this question is important. It is directed at the adequacy of the test itself. Ned Block's 1981 claim is that failing the test is *not* decisive. It is possible to fail the Turing Test for intelligence, he says, and still be an intelligent being. This has provoked a series of arguments showed in Fig. 5.6.

5.8.3.1 Assessment of the Debate So Far

It would seem that the three claims offered by Ned Block and Jack Copeland would have been answered (Block, 1981; Copeland, 1993). Perhaps they have. But it is surprising that we did not find these rebuttals in our extensive search of the literature. Block claims judges may discriminate too well. Overly discerning or chauvinistic judges might fail intelligent machines solely because of their machine-like behavior. Copeland points out that intelligent machines could fail the test by acting nonhuman, by being psychologically unsophisticated (though intelligent), or simply by being bored with the proceedings. Copeland also says chimpanzees, dolphins, and prelinguistic infants can think, but would fail the Turing Test. If a thinking animal could fail the test, then presumably a thinking machine could too.

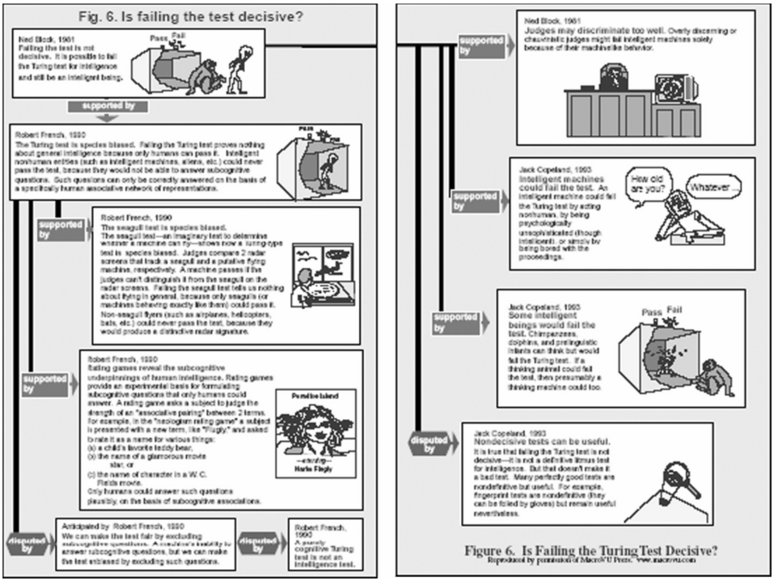

Fig. 5.6 Is failing the test decisive?

There are some qualifications that could answer these objections. In the next rounds of the debate, who will make them? Will they hold up? This shows how argumentation maps are extensible in quite a straightforward fashion. They afford the community of those interested in the debates a framework for joining in and contributing. We are currently working on software that would permit such collaborative argumentation on the web. When this facility is available, participants will be able to offer additions to the maps as well as get an up-to-date status of the debates.

5.8.4 Is Passing the Test Decisive?

This question is the obverse of the previous question. Block starts the debate asserting that passing the test is not decisive. Even if a computer were to pass the Turing Test, he says, this would not justify the conclusion that it was thinking intelligently. This side of the debate has produced a very lively and extensive exchange (see Fig. 5.7).

5.8.4.1 Assessment of the Debate So Far

Most of the debate so far has focused on whether the test is too narrow and proponents of the test have rallied to its defense. Note that here is where the arguments summarized on Map 2 are linked to many other parts of the *Can Computers Think?* series. For example, here is the link to John Searle's famous Chinese Room argument (see Fig. 5.8).

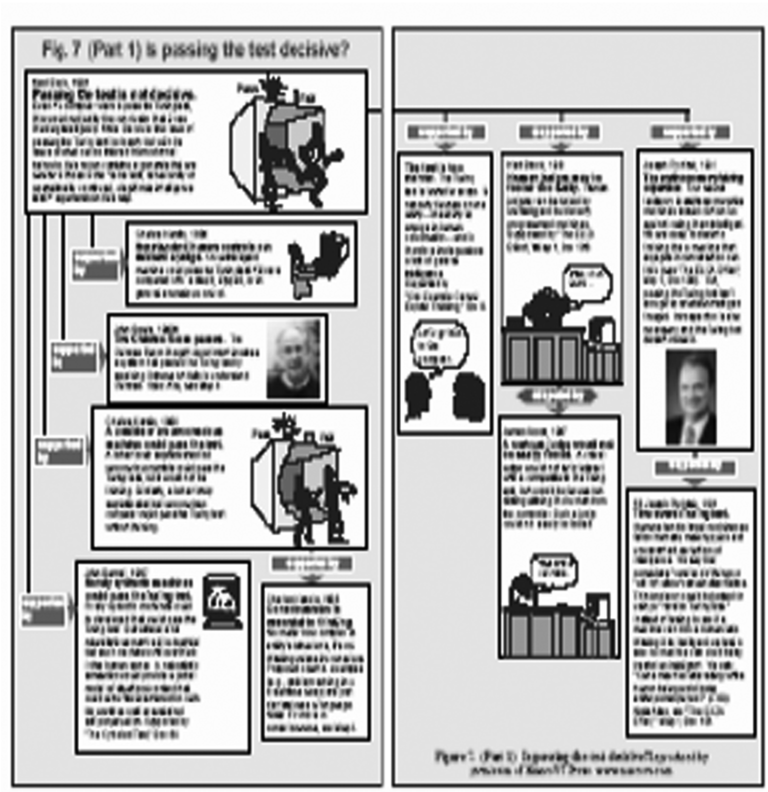

Fig. 5.7.1 (Part 1) Is passing the test decisive?

5.8.4.2 Assessment of the Debate So Far

The Chinese Room argument itself produced such an extensive and prolonged debate that we have devoted one entire map (over 100 moves) to that debate (Map 4). Participants on neither side of the Chinese Room debate concede that the other side has made decisive arguments. What are we to think about that? Why is the debate stalled? Is something missing? Is it only that protagonists will stick by their assumptions?

5.8.5 Have Any Computers Passed the Test?

It is apparent that Turing's Test has not been passed by any computers yet. But we have provided a framework for future debate about it. The two software programs presented on the map simply indicate how future moves in the debate will be treated (see Fig. 5.9). This issue area is where the tests conducted annually with real software will swell the debate.

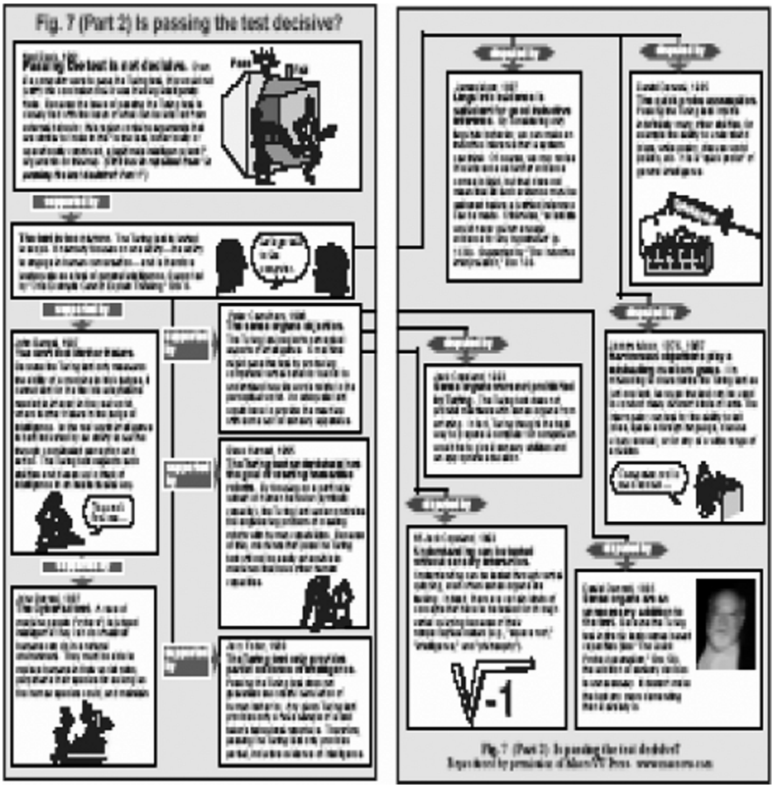

Fig. 5.7.2 (Part 2) Is passing the test decisive?

5.8.5.1 Assessment of the Debate So Far

One of the exciting aspects of following the Turing Test debates is Kurzweil's (1999) claim that supercomputers will reach the capacity of the human brain in 2010.

5.9 Conclusions

5.9.1 Benefits of Argumentation Maps

To sum up, the argumentation maps have a number of benefits for a debate as extensive as that about the Turing Test. Argumentation maps:

- Identify clearly the major supporting evidence for the claims and major attacks upon them in a way that makes easy comparison and evaluation of arguments

Fig. 5.8 The Chinese Room

- Enable participants to separate the claims and rebuttals from the process of evaluating (or giving weight to) them
- Show at a glance the location of unanswered claims or rebuttals (making the discussion more sharply focused) and enable the discussion to grow (instead of starting over again from the beginning each time the group gets together or a new book is published)
- Provide a framework to enable the group of inquirers to go easily to deeper levels of detail to investigate their differences and agreements
- Integrate arguments from all points of view and supporting data and can show the worldviews from which participants argue

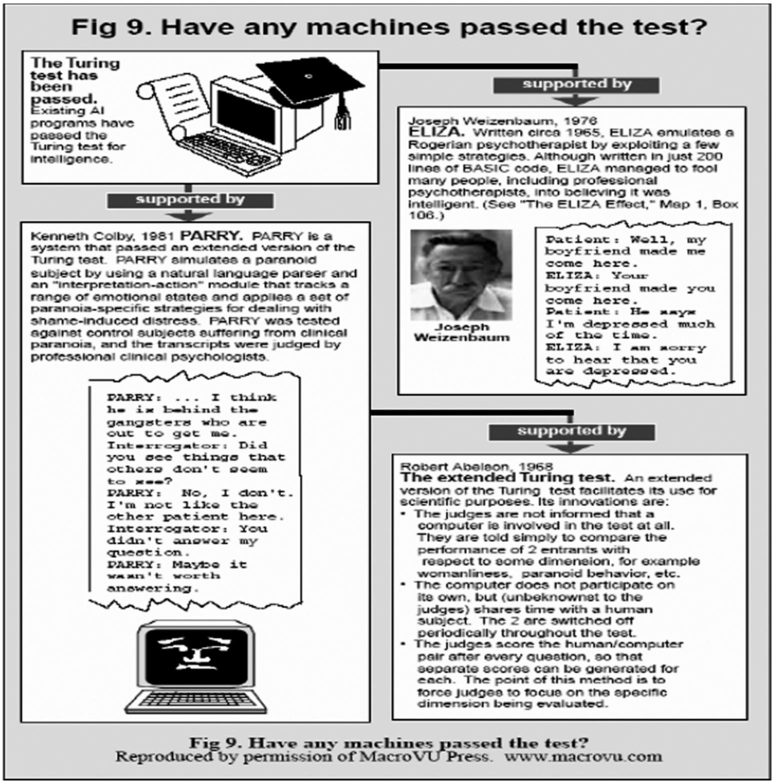

Fig. 5.9 Have any computers passed the test?

For this reason, we look forward to more progress in clarifying the issues and aiding us all to navigate the debate, and eventually, perhaps, a decisive conclusion to these debates.

Notes

Our argumentation mapping project began in the mid-1980s. The work I had done on the Information Mapping® method of structured writing and analysis is a successful attempt to carefully delineate a taxonomy and a methodology for relatively stable subject matter. (Horn, 1989, 1992a, b, 1993, 1995) Relatively stable subject matter is that which does not change much and about which there is little dispute. Information Mapping's method is widely used in industry and government for writing documentation, training, procedures and policies, and reports. Information Mapping is a registered trademark of Information Mapping, Inc. (See www.infor-map.com.) The argumentation mapping extends these ideas to disputed discourse.

Acknowledgments I want to salute and thank the members of my team – Jeff Yoshimi, Mark Deering (of the University of California, Irvine) and Russ McBride (University of California, Berkeley) – without whose dedicated effort and creative thought these maps would not be what they are today. I also want to thank the publishers, MacroVU, Inc. and the Lexington Institute, for their generous support of the *Can Computers Think?* Project.

References

Horn, R. E., 1989, *Mapping Hypertext: Analysis, Linkage, and Display of Knowledge for the Next Generation of On-line Text and Graphics*, The Lexington Institute, Lexington, MA.

Horn, R. E., 1992a, *How High Can it Fly – Examining the Evidence on Information Mapping's Method of High Performance Communication*, The Lexington Institute, Lexington, MA.

Horn, R. E., 1992b, Clarifying two controversies about information mapping's method, *Educational and Training Technology International* **2**(29): 109–117; http://www.stanford.edu/~rhorn/ClarifyingTwo.html.

Horn, R. E., 1993, Structured writing at twenty five, *Performance and Instruction*, February 1993, pp. 11–17; http://www.stanford.edu/~rhorn/Horn-StWritingAt25.html.

Horn, R. E., 1995, Structured writing as a paradigm, in: *Instructional Development: State of the Art*, Romiszowski, A. and Dills, C., eds., Educational Technology Publications, Englewood Cliffs, NJ; http://www.stanford.edu/~rhorn/HornStWrAsParadigm.html.

Horn, R. E., 1998a, *Mapping Great Debates: Can Computers Think?* A series of seven maps and handbook, MacroVU, Bainbridge Island, WA.

Horn, R. E., 1998b, *Visual Language: Global Communication for the 21st Century*, MacroVU, Bainbridge Island, WA.

Kurzweil, R., 1999, *The Age of Spiritual Machines: Where Computers Exceed Human Intelligence*, Viking, New York.

Moravec, H., 1999, *Robot: Mere Machines to Transcendent Mind*. Oxford University Press, New York.

Turing, A., 1950, Computing machinery and intelligence, *Mind* **59**: 434–460.

References Cited in the Figures

Abelson, R. P., 1968, Simulation of social behavior, in: *The Handbook of Social Psychology*, 2nd edn., G. Lindzey and E. Aronson, eds., Addison Wesley, Reading, MA, pp. 274–356.

Barresi, J., 1987, Prospects for the Cyberiad: certain limits on human self-knowledge in the cybernetic age, *Journal for the Theory of Social Behavior* **1**: 19–46.

Block, N., 1981, Psychologism and behaviorism, *The Philosophical Review* **40**(1): 5–43.

Carleton, L., 1984, Programs, language, understanding, and Searle, *Synthese* **59**: 219–30.

Carruthers, P., 1986, *Introducing Persons*, Croom Helm, London.

Chisholm, R., 1957, *Perceiving: A Philosophical Study*, Cornell University Press, Ithaca, NY.

Colby, K. M., 1981, Modeling a paranoid mind, *Behavioral and Brain Sciences* **4**: 515–560.

Colby, K. M., Hilf, F., Weber, S., and Kraemer, H., 1972, Turing-like indistinguishability tests for validation of a computer simulation of paranoid processes, *Artificial Intelligence* **3**: 199–221.

Copeland, J., 1993, *Artificial Intelligence: A Philosophical Introduction*, Blackwell, Oxford.

Dennett, D. C., 1985, Can machines think? in: *How We Know*, M. Shafto, ed., Harper & Row, San Francisco, CA, pp. 121–145.

Fodor, J. A., 1968, The logic of simulation, in: *Psychological Explanation: An Introduction to the Philosophy of Psychology*, Random House, New York, pp.121–52.

French, R. M., 1990, Subcognition and the limits of the Turing Test, *Mind* **99**(393): 53–64.
Gunderson, K., 1964, The imitation game, in: *Minds and Machines*, A. R. Anderson, ed., Prentice-Hall, Englewood Cliffs, NJ, pp. 60–71.
Harnad, S., 1995, Does mind piggyback on robotic and symbolic capacity? in: *The Mind, The Brain, and Complex Adaptive Systems*, H. Morowitz and J. Singer, eds., Proceedings vol. 22 in Santa Fe Institute studies in the sciences of complexity, Addison Wesley, Reading, MA, pp. 203–220.
Karelis, C., 1986, Reflections on the Turing Test, *Journal for the Theory of Social Behavior* **16**(2): 161–172.
Kurzweil, R., 1999, *The Age of Spiritual Machines: Where Computers Exceed Human Intelligence*, Viking, New York.
Moor, J. H., 1976, An analysis of the Turing Test, *Philosophical Studies* **30**: 249–257.
Moor, J. H., 1987, Turing Test, in: *The Encyclopedia of Artificial Intelligence*, S. C. Shapiro and D. Eckroth, eds., Wiley, New York, Vol. 2, pp. 1126–1130.
Moravec, H., 1999, *Robot: Mere Machines to Transcendent Mind*, Oxford University Press, New York.
Putnam, H., 1975a, Brains and behavior, in: *Mind, Language, and Reality: Philosophical Papers*, Cambridge University Press, Cambridge, Vol. 2, pp. 325–361.
Rychlak, J., 1991, *Artificial Intelligence and Human Reason: A Teleological Critique*, Columbia University Press, New York.
Ryle, G., 1949, *The Concept of Mind*, Barnes & Noble, New York.
Searle, J. R., 1980b, Minds, brains, and programs (with peer commentary and author's response), *Behavioral and Brain Sciences* **3**: 417–457.
Stevenson, J., 1976, On the imitation game, *Philosophia* **6**: 131–133.
Turing, A. M., 1950, Computing machinery and intelligence, *Mind* **59**: 433–460.
Weizenbaum, J., 1976, *Computer Power and Human Reason*, W. H. Freeman, San Francisco, CA.

Chapter 6
If I Were Judge

Selmer Bringsjord

Abstract: I have spent a lot of time through the years attacking the Turing Test and its variants (e.g., Harnad's Total Turing Test). As far as I am concerned, my attacks have been lethal, but of course not everyone agrees. At any rate, in the present paper I shift gears: I pretend that the Turing Test is valid, put on the table a proposition designed to capture this validity, and then slip into the shoes of the judge, determined to deliver a correct verdict as to which contestant is the machine, and which the woman. My strategies for separating mind from machine may well reveal some dizzying new-millennium challenges for Artificial Intelligence.

Keywords Artificial Intelligence, Turing Test

6.1 Introduction

I have spent a lot of time through the years attacking the Turing Test and its variants. For example, the Turing Test and *many* variants (e.g., the *Total* Turing Test and the *Total Total* Turing Test) are overthrown in "Could, how could we tell if, and why should-androids have inner lives?" (Bringsjord, 1995). I have also proposed complete replacements for the Turing Test, in "Creativity, the Turing Test, and the (better) Lovelace test" (Bringsjord et al., 2001). As another example, I have recently carefully refined, extended, and defended Searle's (1980) Chinese Room Argument against the Turing Test (Bringsjord, 1992; Bringsjord and Noel, 2002). As far as I am concerned, these attacks have been lethal, but not everyone agrees (at least not yet). At any rate, in the present paper I shift gears: I pretend that the Turing Test is valid, put on the table a proposition designed to capture this validity, and then slip into the shoes of the judge, determined to deliver a correct verdict as to which contestant is the machine, and which is the woman. My strategies for

Rensselaer Polytechnic Institute

separating mind from machine may well reveal some dizzying new-millennium challenges for Artificial Intelligence (AI).

6.2 Validity of the Turing Test in Declarative Form

The basic architecture of the Turing Test will be familiar to all readers; it is given in Turing's (1950) famous *Mind* paper. A judge must attempt to determine which of two sequestered agents is a machine, and which is a woman. The judge can interact with the agents only via (to modernize things a bit) typed e-mail. Many, many variations on the Turing Test have been suggested. In fact, rather long ago, in my *What Robots Can and Can't Be* (1992), I defined the Turing Test Sequence, which assumes tests ranging from those less demanding than Turing Test (e.g., judges cannot know anything about AI, and can only ask questions about a small, determinate domain), to those that – like Kugel's (1990) – require contestants to have a capacity for infinitary processing. (When I introduced the sequence, I also asserted that, sooner or later, a machine can be built by us to pass any and every test in it.) Many of the variations in this sequence have been offered to supplant the original Turing Test, which even to fans of "Strong" AI seems to be afflicted by a certain myopia (e.g., the Turing Test ignores sensorimotor behavior in favor of the purely linguistic variety). However, I now assume for the present paper that the original Turing Test is in fact valid.

Now let us get a little bit more precise about what it means to say that the Turing Test is valid. One possibility is:

(TT_1) For every computer *c*, if *c* passes Turing Test, then *c* is conscious.

It may strike you as odd, if not flatly wrong, that we have turned the focus upon consciousness. Someone might object, specifically, as follows: 'Turing presents his test as a test for "intelligence" or "thought." Your interpretation of Turing Test cheats over toward precisely what Turing sought to dodge: phenomenal consciousness and qualia.' But a careful reading of Turing's (1950) paper supports my interpretation. Specifically, I draw your attention to the section therein entitled '(4) The Argument from Consciousness.' The argument from consciousness is one Turing takes to be well-expressed in Professor Jefferson's Lister Oration of 1949:

> Not until a machine can write a sonnet or compose a concerto because of thoughts and emotions felt, and not by the chance fall of symbols, could we agree that machine equals brain – that is, not only write it, but know that it had written it. No mechanism could feel (and not merely artificially signal, an easy contrivance) pleasure at its successes, grief when its valves fuse, be warmed by flattery, be made miserable by mistakes, be charmed by sex, be angry or depressed when it cannot get what it wants (Turing, 1950).

What is Turing's response? I am afraid he gives an absolutely dreadful rejoinder; it amounts to:

> If one refuses to agree that passing Turing Test (when, say, sonnets are requested by the judge) implies consciousness, one must accept solipsism. Since solipsism is false, one cannot refuse in the manner indicated by Jefferson.

Though you and I no doubt reject solipsism, I cannot for the life of me imagine where Turing gets his first premise. To take a guess, perhaps the underlying idea is that if one rejects the notion that *c*'s passing implies that *c* is conscious, one will have no solution (or candidate solution) to the problem of other minds – save for solipsism. But since the literature is filled with proposed solutions to the problem of other minds that make no appeal to the Turing Test, if this is Turing's underlying idea, it is a fatally flawed one. At any rate, Turing's argument is beside the point we have by now made, which is that Turing clearly holds that if a computer (or robot) passes his test (in part by producing sonnets, etc.), it follows that it is conscious.

However, I propose in the present paper to assume a version of Turing's claim for the Turing Test that insulates him from all the sorts of attacks that naturally arise if (TT_1) is the target. (The thesis claiming that passing Turing Test ensures consciousness is vulnerable to attacks based on thought-experiments in which passing occurs, but subjective awareness is absent. Searle's (1980) CRA is an example of such an attack.) Specifically, I propose something like:

(TT_2) For every computer *c*, if *c* passes Turing Test, then *c* is as intelligent as human persons.

Notice how charitable a move toward such a cashing out of "the Turing Test is valid" is. (TT_2) makes no reference (at least no overt reference) to invisible mental properties, which would surely gladden Turing's empiricist heart. On the other hand, there is a severe defect in (TT_2): It classifies a Turing Test-passing computer as human-level intelligent, but it does not stipulate that the human contestant is smart! To concretize this point, suppose that a machine competes alongside a 2-year old in the confines of the Turing Test. We can probably safely assume that some computer today, or at least in the near future, could leave me in the dark as to which room houses little, just-learning-to-talk Johnny. What this reveals is that propositions like (TT_2) leave concealed a fact that needs to be uncovered: viz., that there are *a lot* of candidate human contestants.

I see this point as another version of one made by the thinker who proposed Turing Test before (!) Turing: Descartes.[1] Here is the relevant passage:

> If there were machines which bore a resemblance to our body and imitated our actions as far as it was morally possible to do so, we should always have two very certain tests by which to recognize that, for all that, they were not real men. The first is that they could never use speech or other signs as we do when placing our thoughts on record for the benefit of others. For we can easily understand a machine's being constituted so that it can utter

[1] Actually, Descartes proposed a test that is much more demanding than the Turing Test (Descartes, 1911), but I do not explain and defend this herein. In a nutshell, if you read the passage very carefully, you will see that Descartes' test is passed only if the computer has the capacity to answer arbitrary questions. A machine which has a set of stored chunks of text that happen to perfectly fit the queries given it during a Turing test would not pass Descartes' test – even though it would pass Turing's.

> words, and even emit some responses to action on it of a corporeal kind, which brings about a change in its organs; for instance, if it is touched in a particular part it may ask what we wish to say to it; if in another part it may exclaim that it is being hurt, and so on. But it never happens that it arranges its speech in various ways, in order to reply appropriately to everything that may be said in its presence, as even the lowest type of man can do. And the second difference is, that although machines can perform certain things as well as or perhaps better than any of us can do, they infallibly fall short in others, by which means we may discover that they did not act from knowledge, but only for the disposition of their organs. For while reason is a universal instrument which can serve for all contingencies, these organs have need of some special adaptation for every particular action. From this it follows that it is morally impossible that there should be sufficient diversity in any machine to allow it to act in all the events of life in the same way as our reason causes us to act. (Descartes 1911)

To operationalize Descartes' point about the diversity of human cognition, and the challenge this poses to a machine, I will say that passing the Turing Test is actually at least a ternary relation *Pass* taking as arguments a computer c, a human player h, and a judge j. By hypothesis, I am the judge; let us denote me by b_j. The proposition itself then is:

$$(\mathrm{TT}_3)\ \forall c\,\forall h(Pass(c, h, b_j) \rightarrow \text{then } c \text{ is as intelligent as human persons})$$

Quantification over contestants has considerable logical payoff. I say this because (TT_2) is vacuously true, and surely that's an unwanted consequence. (TT_2) is vacuously true because for every computer c today, it is simply false that c can pass the Turing Test when stacked against all people. Therefore (TT_2)'s antecedent is false. But by the standard semantics of first-order logic, this makes the entire proposition true. I have discussed this situation (Bringsjord, 1995). And yet there is a problem with (TT_3): I should not be allowed to pick a *particular* individual. We really should be talking about picking from a *class* of humans. It should be obvious why this is so. If I could pick an individual, and if the competing computer does not know this individual inside and out, I have only to ask a few "private" questions to prevail. Turing did not envisage a situation wherein a machine would be trying to impersonate a specific human being. Here is the solution where C ranges over classes of human persons:

$$(\mathrm{TT}_4)\ \forall c\ \forall h(h \in C \rightarrow (Pass(c, h, b_j) \rightarrow \text{then } c \text{ is as intelligent as human persons}))$$

(TT_4) needs further refinement. The problem is that it makes no reference to how long the test is to last. Following what I did in Bringsjord (1995), let us let τ denote the length of time the Turing Test is played for; we can follow this notation here. We can expand the key relation to take four arguments, and we can quantify over intervals. So we have:

$$(\mathrm{TT}_5)\ \forall c\ \forall h\ \forall \tau(h \in C \rightarrow (Pass(c, h, b_j, \tau) \rightarrow \text{then } c \text{ is as intelligent as human persons}))$$

Now (TT_5) can be used to anchor an adversarial relationship between me as judge and computer as competitor. That is, it is now under my control as to what length of time to opt for, and what sort of human to select (= what class C to pick the human from) as my other interlocutor. For reasons already cited, it would not be a particularly clever strategy for me to request that h be instantiated to a 2-year-old, and that τ be set to one minute. So, what *would* be an intelligent pick? To that we now turn.

6.3 My Strategies

To qualify as a Turing Test-passer, I will require that the class C of humans against which c is matched be the *union* of a number of classes. Each of the following four strategies (standardized tests of mental ability, tests for "irrationality", requests that certain paradoxes be solved, and tests for literary creativity) is associated with at least one subclass within this union.

6.3.1 Standardized Tests as a First Hurdle

My first strategy would be to require the computer c to excel on all established, standardized tests – tests of intelligence, creativity, spatial reasoning, and so on. This strategy relates to a form of AI invented by me and Bettina Schimanski; we refer to it as Psychometric AI (Bringsjord and Schimanski, 2003; Bringsjord and Zenzen, 2003). Together, the two of us are attempting to build a robot capable of reaching high performance on *all* standardized tests; this robot is PERI, who "lives" in the Rensselaer AI and Reasoning Lab. PERI is shown in Fig. 6.1. I have a pretty good idea of how demanding it is to pass the hurdle of reaching such performance because of these efforts. In the present essay, I say but a few words about this challenge.

I would first insist the c take a "broad" IQ test, on which it would need to score as high as the best-performing humans in order to remain a candidate for passing Turing Test. What could possibly be a more obvious strategy? After all, by (TT_5), the Turing Test is about *intelligence*, and we have many established broad tests of intelligence. But what is meant by a "broad" IQ test? This question is best answered by turning to an example of a "narrow" intelligence test. The example of a narrow IQ test that I give here is one that Bettina Schimanski and I have built for an artificial agent to crack: Raven's (1962) Progressive Matrices (RPM). An example of the type of problem that appears in RPM is shown in Fig. 6.2, which is taken from Carpenter et al. (1990). The query in each RPM problem is implicit, and invariant: viz., pick the option that preserves the vertical and horizontal patterns. Obviously, the items in question do not relate to general (declarative) knowledge, common sense, or the ability to communicate in natural language.

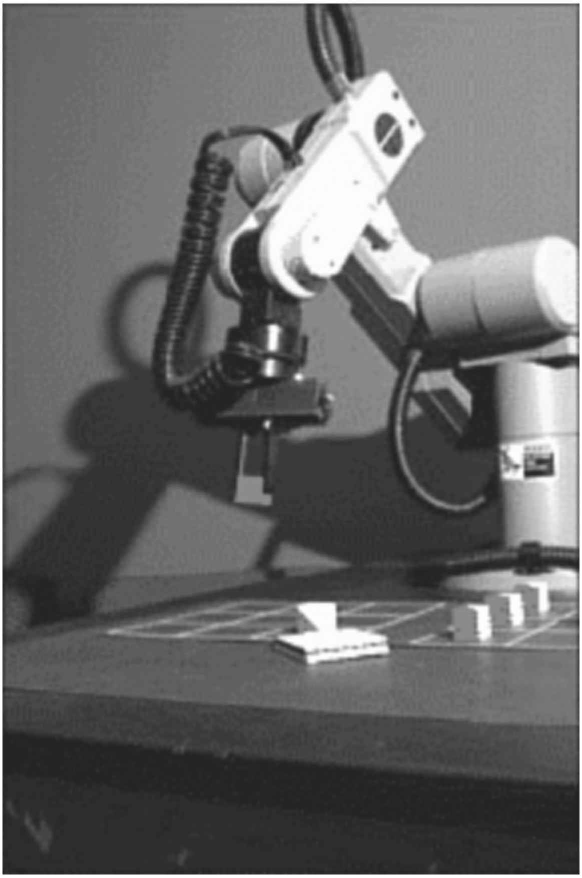

Fig. 6.1 PERI working on the block design puzzle

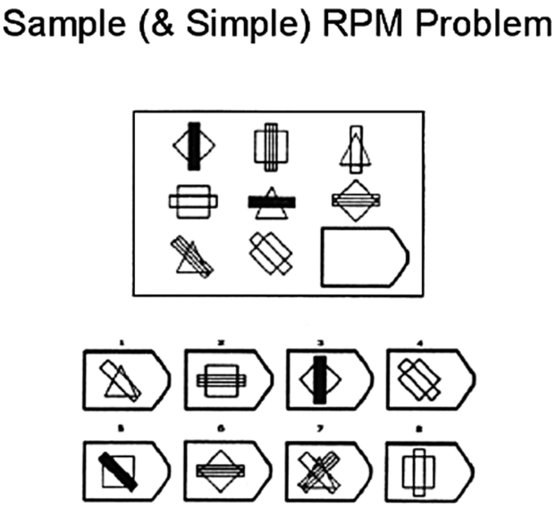

Fig. 6.2 A simple RPM problem “cracked” by a Bringsjord-created intelligent agent

An example of a broad intelligence test is the WAIS (Wechsler Adult Intelligence Scale, available from Psychological Corporation). One of the subtests on the WAIS is block design, in which test-takers must assemble cubes whose sides have different colored patterns to make a larger pattern given as a goal state. (Figure 6.1 shows PERI successfully completing a block design puzzle.) But the WAIS also contains some harder subtests. For example, there is a subtest in which, in conversation, subjects are asked questions designed to determine whether or not they can reason in "common-sense" fashion. For example, subjects might be asked to explain why the tires on automobiles are made of rubber, instead of, say, wood. Another subtest on the WAIS, picture completion, requires that coherent stories be assembled from snapshots of various situations. To our knowledge, no present-day AI system can correctly answer arbitrary questions of this sort, or solve these kinds of narrative-related problems. Hopefully this gives you a tolerably clear sense of the distinction between "narrow" and "broad" in this context, and perhaps you also appreciate that it would be no small feat for a computer to solve these sorts of problems.

I would give the would-be Turing Test-passer not just intelligence tests, but, as I said, *all* established, standardized tests of mental ability. For example, I would subject c to tests of creativity. These tests, in the context of building artificial agents, are discussed by Bringsjord and Ferrucci (2000). On the assumption that a would-be Turing Test-passer clears the first hurdle by matching brilliant humans on all tests of mental ability, I move to the second.

6.3.2 Irrationality

In the next hurdle, I would pick as my class of humans those with college educations, but no advanced training in formal reasoning. I would give the computer c problems like this one[2]:

Problem 1

1. If there is a king in the hand, then there is an ace, or else if there is not a king in the hand, then there is an ace.
2. There is a king in the hand.

Given these premises, what can one infer?

Almost certainly your own verdict is this: One can infer that there is an ace in the hand. Your verdict seems correct, even perhaps *obviously* correct, and yet a little

[2] Problem 1 (or, actually, Illusion 1) is from "How to make the impossible seem probable" (Johnson-Laird and Savary, 1995). Variations are presented and discussed in "Rules and illusions: a critical study of Rips's" (Johnson-Laird, 1997a).

logic suffices to show that not only are you wrong, but that in fact what you can infer is that there *isn't* an ace in the hand![3]

To see this, note that "or else" is to be understood as exclusive disjunction,[4] so (using obvious symbolization) the two premises become

$$(1') \ ((K \rightarrow A) \vee (\neg K \rightarrow A)) \wedge \neg((K \rightarrow A) \wedge (\neg K \rightarrow A))$$

$$(2') \ K$$

Figure 6.3 shows a proof in the standard first-order Fitch-style system F, constructed in HYPERPROOF (Barwise and Etchemendy, 1994), that demonstrates that from these two given one can correctly conclude $\neg A$.

$((K \rightarrow A) \vee (\neg K \rightarrow A)) \wedge \neg((K \rightarrow A) \wedge (\neg K \rightarrow A))$	✓ Given
K	✓ Given
$\neg((K \rightarrow A) \wedge (\neg K \rightarrow A))$	✓ ∧ Elim
$\neg(K \rightarrow A) \vee \neg(\neg K \rightarrow A)$	✓ Taut Con
$\neg(K \rightarrow A)$	✓ Assume
$K \wedge \neg A$	✓ Taut Con
$\neg A$	✓ ∧ Elim
$\neg(\neg K \rightarrow A)$	✓ Assume
$\neg K \wedge \neg A$	✓ Taut Con
$\neg A$	✓ ∧ Elim
$\neg A$	✓ ∨ Elim

Fig. 6.3 A proof that there is no ace in the hand in F

[3] You should not feel bad about succumbing to Illusion 1; after all, you have a lot of company. Johnson-Laird has recently reported that

> "Only one person among the many distinguished cognitive scientists to whom we have given [Illusion 1] got the right answer; and we have observed it in public lectures – several hundred individuals from Stockholm to Seattle have drawn it, and no one has ever offered any other conclusion" (Johnson-Laird, 1997b).

> Time and time again, in public lectures, I have replicated Johnson-Laird's numbers – presented in (Johnson-Laird and Savary, 1995) – *among those not formally trained in logic*.

[4] Even when you make the exclusive disjunction explicit, the results are the same. For example, you still have an illusion if you use

Illusion 1'

(1") If there is a king in the hand then there is an ace, or if there is not a king in the hand then there is an ace, but not both.
(2") There is a king in the hand.

Given these premises, what can you infer?

While psychologists of reasoning create problems like this in order to carry out experiments on humans, AI researchers might be interested in programming computers that *generate* such illusions. Indeed, this is exactly my interest, and I have elsewhere discussed how it is that I came to work on an algorithm able to generate this problem:

Problem 2

3. The following three assertions are either all true or all false:
   - If Billy is happy, Doreen is happy
   - If Doreen is happy, Frank is as well
   - If Frank is happy, so is Emma
4. The following assertion is definitely true: Billy is happy.

Can it be inferred from (3) and (4) that Emma is happy?

Most human subjects answer "Yes", but get the problem wrong – because their *reasons* for answering with an affirmative are incorrect. They say "Yes" because they notice that since Billy is happy, if the three conditionals are true, one can "chain" through them to arrive at the conclusion that Emma is happy. But this is only part of the story, and the other part has been ignored: viz., that it could be that all three conditionals are false. Some subjects realize that there are two cases to consider (conditionals all true, conditionals all false), and because they believe that when the conditionals are all false one cannot prove that Emma is happy, they respond with "No". But this response is also wrong. The correct response is "Yes", because *in both cases it can be proved that Emma is happy*. This can be shown using propositional logic; the proof, once again constructed in HYPERPROOF, is shown in Fig. 6.4. This proof establishes

$$\{\neg(B \rightarrow D), \neg(D \rightarrow F)\} \mid\!- E$$

Note that the trick is exploiting the inconsistency of the set $\{\neg(B \rightarrow D), \neg(\mathrm{D} \rightarrow F)\}$ in order to get a contradiction. Since everything follows from a contradiction, E can then be derived.

What does all this have to do with the Turing Test? I would expect to unmask many would-be Turing Test-passers with problems like these, for the simple reason that a machine, *ceteris paribus*, would not be fooled. That is, the machine might well parse these problems *correctly*, and would then reason them out in accordance with normatively correct structures from symbolic logic; that is, reason them out essentially as shown in Figs. 6.4 and 6.3. In other words, the machine must be smart enough to appear dull. Note that a computer smart enough to meet this challenge would presumably be capable of some form of meta-reasoning. To *really* test for meta-reasoning I would next see if my digital opponent can solve a paradox or two. Suppose that "generic" paradox P is the derivation of contraction $\varphi \Lambda \neg \varphi$ from set Φ of premises. It is unavoidable that, in trying to solve P, one must reason about this reasoning; that is, it is unavoidable that the attempt to solve P involves meta-reasoning. In addition, P may involve propositions that are themselves very expressive, so that modeling them requires very sophisticated modes of representation and

◈	✓ Given
$((H(b) \rightarrow H(d)) \wedge (H(d) \rightarrow H(f)) \wedge (H(f) \rightarrow H(e))) \vee (\neg(H(b) \rightarrow H(d)) \wedge \neg(H(d) \rightarrow H(f)) \wedge \neg(H(f) \rightarrow H(e)))$	✓ Given
$H(b)$	✓ Given
$(H(b) \rightarrow H(d)) \wedge (H(d) \rightarrow H(f)) \wedge (H(f) \rightarrow H(e))$	✓ Assume
$H(b) \rightarrow H(d)$	✓ $\wedge$ Elim
$H(d)$	✓ $\rightarrow$ Elim
$H(d) \rightarrow H(f)$	✓ $\wedge$ Elim
$H(f)$	✓ $\rightarrow$ Elim
$H(f) \rightarrow H(e)$	✓ $\wedge$ Elim
$H(e)$	✓ $\rightarrow$ Elim
$(\neg(H(b) \rightarrow H(d)) \wedge \neg(H(d) \rightarrow H(f)) \wedge \neg(H(f) \rightarrow H(e)))$	✓ Assume
$\neg(H(b) \rightarrow H(d))$	✓ $\wedge$ Elim
$H(b) \wedge \neg H(d)$	✓ Taut Con
$\neg(H(d) \rightarrow H(f))$	✓ $\wedge$ Elim
$H(d) \wedge \neg H(f)$	✓ Taut Con
$\neg H(e)$	✓ Assume
$H(d) \wedge \neg H(d)$	✓ Taut Con
$H(e)$	✓ $\neg$ Intro
$H(e)$	✓ $\vee$ Elim

Fig. 6.4 A proof that "Emma is happy" in F

reasoning. Finally, many paradoxes are infinitary in nature, which implies that agents who would solve them must be able to, at least in some sense, "grasp" infinitary reasoning. All of this may seem rather vague to you. Fortunately, I have recently presented a paradox that makes my points concrete: this paradox is an "infinitized" version of Yablo's (1993) paradox. I present it now as an excerpt from "The mental eye defense of an infinitized version of Yablo's paradox" (Bringsjord and van Heuveln, 2003), in which a full discussion can be found.

6.3.3 A Paradox for a Computer to Tackle

The paradox runs as follows[5]:
Recall the familiar natural numbers $\mathbf{N} = \{0,1,2\ldots\}$. With each $n \in \mathbf{N}$ associate a sentence as follows, using a truth predicate, T:

[5] I specify an infinitary version of Yablo's paradox, expressed in the "background" logic that allows for meta-proofs regarding infinitary logical systems like $£\omega_1\omega$. This system is presented in encapsulated form in *Mathematical Logic* (Ebbinghaus et al., 1984), from which the student interested in infinitary logic can move to *Languages with Expressions of Infinite Length* (Karp, 1964), then to *Model Theory for Infinitary Logic* (Keisler, 1971), and then *Large Infinitary Languages* (Dickmann, 1975).

$$s(0) = \forall k(k > 0 \rightarrow \neg T(s(k)))$$

$$s(1) = \forall k(k > 1 \rightarrow \neg T(s(k)))$$

$$s(2) = \forall k(k > 2 \rightarrow \neg T(s(k)))$$

$$s(3) = \forall k(k > 3 \rightarrow \neg T(s(k)))$$

$$\vdots$$

Expressed with help from the infinitary system $L\omega_1\omega$, we can say that

$$s(0) = \bigwedge_{k>0} \neg\, T(s(k)),\ s(1) = \bigwedge_{k>1} \neg\, T(s(k)),\ s(2) = \bigwedge_{k>2} \neg\, T(s(k))\ldots$$

Next, suppose that $T(s(0))$. From this it follows immediately that $\neg T(s(1))\Lambda\neg T(s(2))\ldots$, which in turn implies by conjunction elimination in £$\omega_1\omega$ that $\neg T(s(1))$. But in addition, if $T(s(0))$ is true, it follows again that $\neg T(s(l))$ $\Lambda\neg T(s(2))\ldots$, and hence that

$$\neg T(s(2))\,\Lambda\,\neg T(s(3))\cdots,$$

which implies that $T(s(1))$. By reduction, we can infer $\neg T(s(0))$. The same indirect proof can be given to show

$$\neg T(s(1)), \neg T(s(2)), \neg T(s(3))\cdots$$

Hence we can infer by the ω-rule

$$\frac{\alpha(1), \alpha(2),\ldots}{\alpha(n)}$$

that

$$(*) \bigwedge_{k\in N} \neg T(s(k))$$

Hence $\neg T$(s(1)), $\neg T$(s(2)), $\neg T$(s(3))…, that is, T(s(0)). But ¬T(s(0)) follows from (*) – contradiction.

I have serious doubts that a computer will ever be able to solve a paradox like this one. (Can you solve it? Or is it a true paradox?) But perhaps you are wondering what a solution would consist of. Well, one possible type of solution for a paradox *P* is to provide a formal theory on which the premises in *P*, Φ, are true, but the contradiction cannot be derived. This kind of solution for the paradox Ihave presented would be remarkable, because the formal theory will in some sense subsume a logical system that in and of itself far exceeds what machines can today (in any sense of the word) understand. It is hard to see how even a future machine would achieve this understanding.

If I assume for the sake of argument that some computer does pass the Turing Test with *C* set to professional logicians and formal philosophers (a group up to the challenge of solving paradoxes), I resort to my final weapon: literary creativity.

6.3.4 Literary Creativity

The idea here is to see if the computer in the Turing Test is capable of producing stories indistinguishable from those produced by accomplished authors of literary fiction. This means that the class *C* of humans against which the computer *c* is matched includes the likes of John Updike, Toni Morrison, Mark Helprin, and so on. I have refined this scenario into what I call the "short short story game", or just S^3G for short. The idea is simple; it is summed up in Fig. 6.5. The computer and human both receive one relatively simple sentence from me, say: "Barnes kept the image to himself, kept the horror locked away as best he could." (For a much better one, see the "loaded" sentence shown in Fig. 6.5.[6]) Both human and machine must now fashion a short short story of no more than 500 words. The machine's objective,

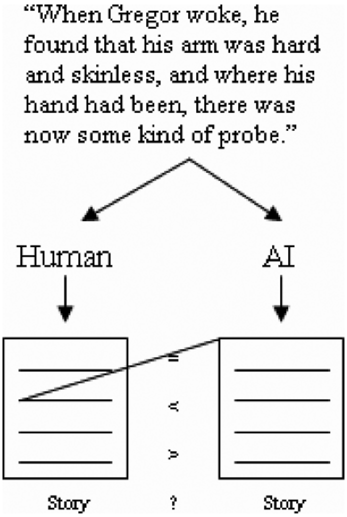

Fig. 6.5 S3G

[6] The actual opening is as follows:

> As Gregor Samsa awoke one morning from uneasy dreams he found himself transformed in his bed into a gigantic insect. He was lying on his hard, as it were armor-plated, back and when he lifted his head a little he could see a dome-like brown belly divided into stiff arched segments on top of which the bed quilt could hardly keep in position and was about to slide off completely. His numerous legs, which were pitifully thin compared to the rest of his bulk, waved helplessly before his eyes. (Kafka, 1948)

of course, is to produce narrative that leaves me in the dark as to whether it is authored by mind or machine. For reasons explained in *Artificial Intelligence and Literary Creativity: Inside the Mind of Brutus, a Storytelling Machine* (Bringsjord and Ferrucci, 2000), I think it will be exceedingly difficult for any computer to match the likes of John Updike. Of course, at 500 words, it may be possible. As the ultimate test, I would as judges allow word length to reach that of a full-length novel (which would of course require that I increase τ considerably).

Acknowledgment I am indebted to my colleague Yingrui Yang, cocreator of mental metalogic theory.

References

Barwise, J. and Etchemendy, J., 1994, *Hyperproof*, CSLI, Stanford, CA.

Bringsjord, S., 1992, *What Robots Can and Can't Be*, Kluwer, Dordrecht, The Netherlands.

Bringsjord, S., 1995, Could, how could we tell if, and why should-androids have inner lives? in: *Android Epistemology*, K. Ford, C. Glymour, and P. Hayes, eds., MIT Press, Cambridge, MA, pp. 93–122.

Bringsjord, S. and Ferrucci, D., 2000, *Artificial Intelligence and Literary Creativity: Inside the Mind of Brutus, a Storytelling Machine*, Lawrence Erlbaum, Mahwah, NJ.

Bringsjord, S. and Noel, R., 2002, Real robots and the missing thought experiment in the Chinese room dialectic, in: *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence*, J. Preston and M. Bishop, eds., Oxford University Press, Oxford, pp. 144–166.

Bringsjord, S. and Schimanski, S., 2003, What is Artificial Intelligence? – Psychometric AI as an answer, *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (IJCAI-03), Morgan Kaufmann, San Francisco, CA, pp. 887–893.

Bringsjord, S. and van Heuveln, B., 2003, The mental eye defense of an infinitized version of Yablo's paradox, *Analysis* **63**(1): 61–70.

Bringsjord, S. and Zenzen, M., 2003, *Superminds: People Harness Hypercomputation, and More*, Kluwer, Dordrecht, The Netherlands.

Bringsjord, S., Ferrucci, D., and Bello, P., 2001, Creativity, the Turing Test, and the (better) Lovelace test, *Minds and Machines* **11**: 3–27.

Carpenter, P., Just, M., and Shell, P., 1990, What one intelligence test measures: a theoretical account of the processing in the Raven progressive matrices test, *Psychological Review* **97**: 404–431.

Descartes, R., 1911, *The Philosophical Works of Descartes*, Vol. 1, translated by Elizabeth S. Haldane and G. R. T. Ross, Cambridge University Press, Cambridge.

Dickmann, M. A., 1975, *Large Infinitary Languages*, North-Holland, Amsterdam, The Netherlands.

Ebbinghaus, H. D., Flum, J., and Thomas, W., 1984, *Mathematical Logic*, Springer, New York.

Johnson-Laird, P., 1997a, Rules and illusions: a critical study of Rips's, *The Psychology of Proof, Minds and Machines* **7**(3): 387–407.

Johnson-Laird, P. N., 1997b, An end to the controversy? A reply to Rips, *Minds and Machines* **7**: 425–432.

Johnson-Laird, P. and Savary, F., 1995, How to make the impossible seem probable, *Proceedings of the 17th Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum, Hillsdale, NJ, pp. 381–384.

Kafka, F., 1948, The metamorphosis, in: *The Penal Colony*, F. Kafka, T. W. Muir, and E. Muir, eds., Schocken Books, New York.

Karp, C., 1964, *Languages with Expressions of Infinite Length*, North-Holland, Amsterdam, The Netherlands.
Keisler, H., 1971, *Model Theory for Infinitary Logic*, North-Holland, Amsterdam, The Netherlands.
Kugel, P., 1990, Is it time to replace Turing's test? Paper presented at *Artificial Intelligence: Emerging Science or Dying Art Form?*, sponsored by AAAI and the State University of New York's program in Philosophy and Computer and Systems Sciences, the University at Binghamton, New York, June 27.
Raven, J. C., 1962, *Advanced Progressive Matrices Set II*, H. K. Lewis, London. Distributed in the USA by The Psychological Corporation, San Antonio, TX.
Searle, J., 1980, Minds, brains and programs, *Behavioral and Brain Sciences* **3**: 417–424; http://members.aol.com/NeoNoetics/MindsBrainsPrograms.html.
Turing, A., 1950, Computing machinery and intelligence, *Mind* **59**(236): 433–460.
Yablo, S., 1993, Paradox without self-reference, *Analysis* **53**: 251–252.

Chapter 7
Turing on the “Imitation Game”

Noam Chomsky

Abstract Turing’s paper has modest objectives. He dismisses the question of whether machines think as “too meaningless to deserve discussion”. His “imitation game”, he suggests, might stimulate inquiry into cognitive function and development of computers and software. His proposals are reminiscent of 17th century tests to investigate “other minds”, but unlike Turing’s, these fall within normal science, on Cartesian assumptions that minds have properties distinct from mechanism, assumptions that collapsed with Newton’s undermining of “the mechanical philosophy”, soon leading to the conclusion that thinking is a property of organized matter, on a par with other properties of the natural world.

Keywords: Cartesian science, computational procedures, Joseph Priestley, organized matter, simulation, thinking

In his justly famous 1950 paper “Computing Machinery and Intelligence”, A. M. Turing formulated what he called “the ‘imitation game’,” later known as “the Turing Test”, a “new form of the question” whether machines can think, designed to focus attention on “the intellectual capacities of a man”. This “new question [is] a worthy one to investigate”, Turing urged, offering several “conjectures” on machine potential that should “suggest useful lines of research”. Human intellectual capacities might be illuminated by pursuit of the task he outlined, which also might advance the welcome prospect “that machines will eventually compete with men in all purely intellectual fields”.

The dual significance of the enterprise – constructing better machines, gaining insight into human intelligence – should no longer be in doubt, if it ever was. There are, however, questions about just where its significance lies, about its antecedents, and about the specific research strategy that Turing proposes.

Massachusetts Institute of Technology

On the matter of significance, Turing expressed his views lucidly and concisely. He begins by proposing "to consider the question, 'Can machines think?,'" but went on to explain that he would not address this question because he believed it "to be too meaningless to deserve discussion", though "at the end of the century", he believed, "the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted". He explained further that for his purposes at least, it would be "absurd" to resolve the issue by determining how the words *machine* and *think* "are commonly used" (a project he conceives much too narrowly, though that is not relevant here).

Turing said nothing more about why he considered the question he posed at the outset – "Can machines think?" – "to be too meaningless to deserve discussion", or why he felt that it would be "absurd" to settle it in terms of "common usage". Perhaps he agreed with Wittgenstein that "We can only say of a human being and what is like one that it thinks"; that is the way the tools are used, and further clarification of their use will not advance the dual purposes of Turing's enterprise. One can choose to use different tools, as Turing suggested might happen in 50 years, but no empirical or conceptual issues arise. It is as if we were to debate whether space shuttles fly or submarines swim. These are idle questions. Similarly, it is idle to ask whether legs take walks or brains plan vacations; or whether robots can murder, act honorably, or worry about the future. Our modes of thought and expression attribute such actions and states to persons, or what we might regard as similar enough to persons. And *person*, as Locke observed, is not a term of natural science but "a forensic term, appropriating actions and their merit; and so belongs only to intelligent agents, capable of a law, and happiness, and misery", as well as accountability for actions, and much else (Locke, 1690). It would be a confusion to seek "empirical evidence" for or against the conclusion that brains or machines understand English or play chess; say, by resort to some performance criterion. That seems a fair rendition of Turing's view.

Of the two "useful lines of research" that Turing contemplated, one – improvement of the capacities of machines – is uncontroversial, and if his imitation game stimulates such research, well and good. The second line of research – investigating "the intellectual capacities of a man" – is a more complex affair, though of a kind that is familiar in the sciences, which commonly use simulation as a guide to understanding. From this point of view, a machine is a kind of theory, to be evaluated by the standard (and obscure) criteria to determine whether the computational procedure provides insight into the topic under investigation: the way humans understand English or play chess, for example. Imitation of some range of phenomena may contribute to this end, or may be beside the point, as in any other domain.

For the reasons that Turing seemed to have in mind, we also learn nothing about whether Jones's brain (where Jones is representative of any human thinker) uses computational procedures for vision, understanding English, solving arithmetic problems, organizing motor action, etc., by observing that, in accord with our ordinary modes of thought and expression, we would not say that a machine carries out the activities, imitating people. Or that matter, by observing that we would not say that Jones himself is performing these actions if he follows instructions that mean nothing to him with input–output relations interpreted by an experimenter as matching human performance of the actions; say, in an "arithmetic room" of the style suggested by John Searle (1980), in which Jones implements an algorithm for long division, perhaps modelled on the algorithm he consciously employs; or a "writing room" in which Jones mechanically carries out instructions that map-coded sound inputs to outputs interpreted as letters in sequence, instructions that might be a close counterpart to an algorithm implemented by Jones's sensorimotor and linguistic systems when he writes down what he hears. No meaningful question is posed as to whether the complex including Jones is doing long division or writing, so there are no answers, whether or not the procedure articulates in an instructive way what the brain is actually doing.

Questions about computational–representational properties of the brain are interesting and seem important; and simulation might advance theoretical understanding. But success in the imitation game in itself tells us nothing about these matters. Perhaps, as Turing believed, the imitation game would provide a stimulus for pursuit of the two "useful lines of research" he advocated; he said little about why this research strategy is preferable to other ways to improve machine capacity and study human intelligence, and his supposition does not seem obvious, apart from some cultural peculiarities that an outside observer might assess with a critical eye.

Turning to antecedents, Turing's imitation game is reminiscent of ideas that were discussed and pursued during what we might call "the first cognitive revolution" of the 17th century, within the context of "the mechanical philosophy", which was based on the conception of matter as inert and governed by principles of contact mechanics. Descartes and his followers attempted to show that the natural world could be incorporated within this framework, including a good part of human perception and action but not workings of the human mind, notably "free will", which "is itself the noblest thing we can have" (Descartes, 1647), Descartes held, and is manifested most strikingly in the ordinary use of language.

The conception raised questions about the existence of other minds: How do we decide whether some creature is a complex mechanism, or is endowed with a mind as well (as we are, we discover in other ways)? To answer this question, experimental tests were proposed to determine whether the creature exhibits properties (mainly language-related) that transcend the limits of mechanism. If it passes the hardest experiments I can devise to test whether it expresses and interprets new thoughts coherently and appropriately as I would, the Cartesians argued, it would be "unreasonable" to doubt that the creature has a mind like mine.

Though similar in some ways to Turing's imitation game, the Cartesian tests for other minds are posed within an entirely different framework. These tests are ordinary science, designed to determine whether some object has a particular property, rather like a litmus test for acidity. The project collapsed when Newton undermined the mechanical worldview, so that the mind/body problem could not even be formulated in Cartesian terms; or any others, so it appears, at least until some new concept of "physical" or "material" is formulated. The natural conclusion, spelled out in the years that followed, is that thinking is a property of organized matter, alongside of other mysterious properties like attraction and repulsion. Thought in humans "is a property of the *nervous system*, or rather of the *brain*", as much "the necessary result of a particular organization [as] sound is the necessary result of a particular concussion of the air" (Priestley, 1777). More cautiously, we may say that people think, not their brains, though their brains provide the mechanisms of thought. As noted, it is a great leap, which often gives rise to pointless questions, to pass from common sense intentional attributions to people, to such attributions to parts of people, and then to other objects.

Throughout the same period, the project of machine simulation was actively pursued, understood as a way to find out something about the world. The great artisan Jacques de Vaucanson did not seek to fool his audience into believing that his mechanical duck was digesting food, but rather to learn something about living things by construction of models, as is standard in the sciences. Turing's intentions seem similar in this regard.

Turing's two "useful lines of research" have proven to be eminently worth pursuing, however one evaluates the research strategy he proposed. Turing's sensible admonitions should also be borne in mind, more seriously than they sometimes have been, in my opinion.

References

Descartes, R., 1647., Letter to Christine of Sweden, 20 November., translated by John Cottingham, Robert Stoothoff, and Dugald Murdoch; reprinted, in: *The Philosophical Writings of René Descartes*, Vol. III (1985), Cambridge University Press, Cambridge, England.

Locke, J., 1690, *An Essay Concerning Human Understanding*, book 2, Chapter 27, Section 26; http://eserver.org/18th/locke-understanding.txt

Priestley, J., 1777, *Disquisitions Relating to Matter and Spirit*, J. Johnson, London, England, pp. 27–28; http://dewey.library.upenn.edu/sceti/printedbooksNew/index.cfm?textID= priestley_disq_sel

Searle, J. R., 1980, Minds, brains, and programs, *Behavioral and Brain Sciences* **3**: 417–457.

Turing, A. M., 1950, Computing machinery and intelligence, *Mind* **59**(236): 433–460.

Chapter 8
On the Nature of Intelligence

Turing, Church, von Neumann, and the Brain

Paul M. Churchland

Abstract Alan Turing is the consensus patron saint of the classical research program in Artificial Intelligence (AI), and his behavioral test for the possession of conscious intelligence has become his principal legacy in the mind of the academic public. Both takes are mistakes. That test is a dialectical throwaway line even for Turing himself, a tertiary gesture aimed at softening the intellectual resistance to a research program which, in his hands, possessed real substance, both mathematical and theoretical. The wrangling over his celebrated test has deflected attention away from those more substantial achievements, and away from the enduring obligation to construct a substantive theory of what conscious intelligence really *is*, as opposed to an epistemological account of how to tell when you are confronting an instance of it. This essay explores Turing's substantive research program on the nature of intelligence, and argues that the classical AI program is not its best expression, nor even the expression intended by Turing. It then attempts to put the famous Test into its proper, and much reduced, perspective.

Keywords Learning, neural computation, Turing, von Neumann

8.1 The Classical Approach: Its Historical Background

Alan Turing wanted to know, as we all want to know, what conscious intelligence *is*. An obvious place to start one's inquiry is the presumptive and prototypical instance of the target phenomenon – normal humans – and the endlessly clever and appropriate behaviors they display in response to the endlessly various perceptual circumstances they encounter. Conscious intelligence presents itself, from the outset, as being some sort of *enduring capacity*, possessed by humans and at least some other animals, to generate behavioral outputs that are somehow appropriate, given the prior state of the intelligent system and the sensory input it has just received.

University of California, San Diego

R. Epstein et al. (eds.), *Parsing the Turing Test.*

This modest opening observation, to which all may agree, leaves us confronting two substantial problems. First, how do we specify, in a suitably general way, what the relation of *input–output appropriateness* consists in, a relation that has infinitely many potential instances? That is to say, what is the peculiar behavioral or functional profile that is our explanatory target here? And second, what sort of internal capacity, power, or mechanism within us is actually responsible for *generating* those appropriate outputs, when given the sensory inputs in question? These are both very hard questions. Given the complex and open-ended character of the input–output relation at issue, we are unable, at least at the outset, to do much more than lamely point to it, as "the one displayed by normal humans". Indeed, it may turn out that the only way to provide an illuminating answer to our first question is to specify in detail the internal generating power or mechanism whose nature is queried in our second question. A general appreciation of how that mechanism works, and how it actually generates its remarkable outputs from its sundry inputs (including, occasionally, some null inputs), will automatically give us a finite but general specification of the infinite appropriateness relation at issue.

Now, to a mathematician such as Alan Turing, and at a time of mathematical development such as the middle of the 20th century, this situation has some obvious and intriguing parallels with a large class of similar situations in the domain of computable functions. For a simple example of the relevant parallel, consider the mathematical function familiar to us as "the basic parabola", namely, $y = x^2$. This dictates, for any given "input" value for x, a unique "output" value for y. Indeed, this function can be usefully and accurately seen as an infinite set of ordered pairs, <*x, y* (= x^2)>, as partially listed below.

x,y(=x^2)

<-2, 4>

<-1, 1>

<0, 0>

<1, 1>

<2, 4>

<3, 9>

<4, 16>

<5, 25>

<6, 36>

<7, 49>, etc

One cannot specify this function by writing out the entire list of the ordered pairs that it embraces, for the list would be infinitely long. Indeed, given that it includes the real numbers as well as the rationals, it is nondenumerably infinitely long. How, then, do we get a grip on it? Fortunately, we can specify, in finite compass, a *recursive procedure*[1] for *generating* the right-hand member of any pair given the left-hand member as input.[2] Any schoolchild can execute the relevant procedure, for it involves nothing more than addressing the relevant input number, and then multiplying it by itself to yield the relevant output. The recursive procedures involved in multiplication may take some time to execute, if the numbers involved happen to be very large. But the time taken to complete the procedure is always finite, for the numbers involved are always finite. While the set of ordered pairs indicated above is indeed an infinite set, every left-hand number in every ordered pair it contains is nonetheless a finite number, and thus the right-hand element can always be generated, as indicated, in finite time.

Accordingly, the specification of a recursive procedure for generating, or computing, the output appropriate to any input, given that input, constitutes a *finite* specification of the entire infinite set at issue. It gives us a finite but still-firm grip on an infinite, but sufficiently ordered or well-behaved, abstract reality.

Except for these finite, recursive specifications, any infinite set, such as the entire set of $<x, y>$ pairs for the basic parabola, would be forever confined, beyond our cognitive reach, to Plato's nonphysical heaven.

Since any function whatever is a set of ordered pairs, we can now divide the class of functions into two mutually exclusive subclasses: those that admit a finite but possibly recursive specification, in the manner of our parabolic example, and those that do not. The former are called *computable* functions, and the latter are called *noncomputable* functions. These latter possess insufficient order, structure, rhyme, or reason to admit any specification more compact than a literally infinite list of their ill-behaved elements. They are, in short, utterly unspecifiable by anyone short of God himself.

Turing's basic theoretical suggestion here is that the general input–output relation that characterizes normal human behavior in the world is one instance of a computable function. After all, our behavior in the world displays a *systematic* if complex structure, and the brain is quite evidently a *finite* system. The guess that human conscious intelligence is, in some way or other, a finite computational specification of an infinite set of potential input–output pairs, is at least an intriguing entry point for further research. Specifically, this guess sends us in search of the very computational procedures that the brain presumably employs in generating its behavioral magic.

---

[1] Intuitively, this is any rote procedure for transforming input items into further items, a procedure that contains sub-procedures that can be deployed again and again until some criterion is met, at which point the procedure then halts with the production of an output item. To *recurse*, note, is literally to *write again*. Prototypical examples of recursive procedures are the ones you learned for the addition, subtraction, multiplication, and division of largish numbers.

[2] Strictly, if the input is an irrational number, we can only *approximate* the output, but we can do that arbitrarily closely. I will ignore this qualification henceforth.

There is no hope, of course, that the relevant procedures will be as simple as those deployed to compute the elements of the function $y = x^2$. None. But that is not what drew us, nor Turing, in this computational direction. What drew us was the need to find some finite way of specifying the infinitely populated appropriateness relation discussed earlier, and the need to find a substantive explanation of how its outputs are actually generated from its inputs. The computational suggestion owed to Turing promises to solve both problems at once. If we pursue a research program aimed at recovering the computational procedures actually used by the avowedly physical brain, or procedures functionally equivalent to them, we can hope to provide a finite but nonetheless general specification of the infinite Platonic set that presumably constitutes human rationality, a set that exists, in its entirety, only in Plato's timeless heaven. At the same time, we can hope to provide a nuts-and-bolts account of how (some of) the elements of that infinite set are (occasionally) computed in this, our physical/temporal world. This, as I see it, is Turing's basic and most enduring insight on this issue.

It is not, to be sure, his only insight. Students of the history here will point immediately to Turing's characterization of what is now called a Universal Turing Machine, a physically realizable discrete-state device that he showed to be capable, at least in principle, of implementing any recursive procedure whatsoever. Given Alonzo Church's prior characterization of computable functions as exactly those functions whose input–output pairs are finitely specifiable by some recursive procedure, Turing's demonstration concerning the recursive prowess of his Universal Machine entailed that such a machine was capable, at least in principle, of computing *any computable function whatsoever*. This is often referred to as the Church–Turing Thesis.

This is indeed of extraordinary importance, both for good and for ill. It lies at the heart of the development of the programmable digital computer, and it lay at the heart of the first great wave of research into what quickly came to be called Artificial Intelligence (AI). It is not hard to see why. What we now call a "computer program" is just a sequence of instructions that directs the (universal) computer's hardwired central processor to execute this, that, or some other recursive procedure, so as to generate appropriate outputs from whatever inputs it may receive. A finite machine is thus rendered capable, depending on what recursive procedures we choose to program into it (a comparatively simple matter), of generating the elements of any computable function we might wish. That is to say, one and the same (universal) machine is capable of imitating – or better, capable of temporarily becoming – any "special-purpose" machine we might wish: a spreadsheet calculator, a word processor, a chess player, a flight simulator, a solar system simulator, and so forth. As we all know, that is how things have turned out, and it is entirely wonderful.

More to the point, the Church–Turing thesis also entails that a universal computer – which, plus or minus a finite memory, is what any standard desktop machine amounts to – is also capable, at least in principle, of computing the elements of whatever marvelous function it is that characterizes the input–output profile of a *conscious intelligence*. Hence the rationale for the original or "classical" approach to creating an AI: *find/write the program that, when run on a* universal *computer, will recreate the same input–output profile that characterizes a normal human*. Given only the modest assumption that the input–output function characteristic of conscious intelligence is *not* one of those pathological, unspecifiable, noncomputable functions, the Church–Turing thesis appears to guarantee that the classical research program will succeed. For it guarantees that there exists a finite specification, in the form of a recursive procedure, that will generate any element of the Holy Grail here at issue, namely, the infinitely membered function characteristic of a conscious intelligence. Our only task is to find that procedure, if only by successive approximation, for it is guaranteed to be there.

8.2 The Classical Approach: Its Actual Performance

All of this is strictly correct, and the relevant program of research, now 40 years old, has produced a great many marvelous things. But, curiously, a programmed machine that displays conscious intelligence, or anything very close to it, is not among them. Researchers began, understandably enough, by attempting to recreate only this or that isolated *aspect* of conscious intelligence, such as the ability to construct logical or algebraic proofs, the ability to segregate a perceptual scene into discrete objects, the ability to parse a sentence, the ability to navigate a toy environment, and so forth. This divide-and-conquer strategy was entirely reasonable: let us first figure out how the obvious components of intelligence are, or might be, achieved, and then worry about integrating our partial successes later on.

Early successes were plentiful, and encouraging. A good machine, and a clever program, can produce dazzling novelties. But as the decades unfolded, the attempts to achieve progressively greater faithfulness to the actual perceptual, cognitive, and behavioral capacities of humans and other animals proved to require an almost exponential increase in the processing times and the memory capacities expended. Progress slowed dramatically, and attempts at suitable integration were postponed indefinitely. All of this was darkly curious, of course, because the speed of signal conduction inside an electronic computer is roughly a million times faster than the speed of signal conduction along the filamentary axon of a human or animal neuron. And the evident "clock speeds" of neural computing elements were also a million times behind their electronic brothers. The advantage should lie entirely with the electronic machine.

But it did not. And whatever was going on inside the human brain, it became increasingly implausible to picture it as engaged in the same sort of *furiously* recursive activities known to be heating up the memory registers and central processors of the electronic machines. The Church–Turing thesis notwithstanding, a reluctant but real skepticism began to spread concerning the classical approach described. How could biological brains be achieving so much more than the high-speed computers, when they apparently had so much less to work with? And why was the classical program hitting a brick wall?

The answer has largely to do with the decidedly *special-purpose* computational architecture of biological brains, as we are about to see. But that is only part of the answer. In retrospect, the classical or program-writing research program presents itself as embarrassingly presumptuous in one central respect. It assumed that the Platonic function characteristic of conscious intelligence was more or less available or accessible to us *at the outset*. It assumed that we already knew, at least roughly or in general outline, what the membership of the relevant infinite set of input–output pairs really is. It remained only to reprise or recreate that evident profile by means of suitably recursive procedures. So, put the programmers to work.

But that Platonic function (let us agree with Church and Turing that such a computable function exists, if only in Plato's heaven) is not in the least evident and available to us. It is only dimly grasped by the folk notions that make up our commonsense "Folk Psychology", and those notions address only a small fraction of the full range of cognitive phenomena – that is, the full extent of the infinite Platonic set – in any case. To assume, as the classical approach did assume, that the target function is more or less known by us, and is at least roughly captured by commonsense Folk Psychology, is to wrongly privilege at the outset a narrow, partial, and possibly false conception of our target phenomenon. It also turns our attention away from the various kinds of empirical research that might positively help in finally *identifying* our target function – research, for example, into the computational activities of the visual system, the motor system, the somatosensory system, and the language system. To deliberately turn one's back, as the classical research program did, on *computational neurobiology*, is to engage in the most egregious form of self-blinkering. For whatever the target function might be, and however else it might be finitely specified, we know going in, thanks to Alan Turing, that it is *already finitely specified in every adult human brain*. The adult human brain is already happily engaged, somehow or other, in computing it. If a finite specification of our target function is what we want to get our hands on, the human brain is the uniquely authoritative place to find it!

Unfortunately, there was a widely repeated and highly influential argument *against* this presumption of unique authority. In retrospect, it can be seen to be a very bad one. Looking at the intricacies of the biological brain, ran the argument, is like looking at the intricacies of a computer's hardware: it will not tell you where the *real* action lies, which is in the peculiar program that the hardware happens to be running.

This breathtaking argument makes the unwarranted and question-begging factual assumption that the biological brain is *also* a general-purpose or universal machine, relevantly like a standard digital computer, a machine that acquires a specific cognitive profile only when programmed with specific, and comparatively ephemeral, recursive procedures.

But the brain is no such thing. And it has been known to be no such thing since the very beginning of the classical research program, when John von Neumann, who is primarily responsible for the architecture of today's serial-processing digital computers, published his seminal researches into the similarities, and the differences, that unite and divide standard electronic computers and natural biological brains. Published back in 1958 and entitled *The Computer and the Brain*, that prescient little book concludes that the presumptive computing elements within the brain, namely, the neurons and their synaptic connections, are both *too slow* in their activities, and *too inaccurate* in their representations, ever to sustain the sorts of high-speed and hyperrecursive activities required for the success of a discrete-state serial-processing electronic computer. The brain is not remotely fast enough to perform, one after another, the thousands upon thousands of recursive steps that are as natural as breathing for a standard computer. And the imperfect accuracy with which it would perform each such step would inevitably lead to fatally accumulated and magnified errors of computation, even if it were fast enough to complete them all in real time. In sum, if the brain were indeed a general-purpose digital serial computer, it would be doomed to be both a computational tortoise and a computational dunce.

As von Neumann (1958) correctly perceived, the brain's computational regime appears to compensate for its inescapable incapacity for fast and accurate logical "depth" by exploiting instead an extraordinary logical "breadth". In his own words, "…large and efficient natural automata are likely to be highly *parallel*, while large and efficient artificial automata will tend to be less so, and rather to be *serial*" (emphasis his). The former "… will tend to pick up as many logical (or informational) items as possible *simultaneously*, and process them *simultaneously*" (emphasis mine). This means, he adds, that we must reach out, beyond the neurons, to include all of the brain's 10^{14} *synapses* in our all-up count of the brain's "basic active organs". This means, allow *me* to add, that the biological brain can execute a whalloping 10^{14} elementary computational operations all at once: not in sequence – all at once. And it can do so ten times a second; perhaps even a hundred times a second.

Von Neumann's observations here were completely correct. But he might as well have been whistling in the wind. As was quickly pointed out, the Church–Turing thesis still guarantees that, however the idiosyncratic human brain may manage its own biologically constrained internal affairs, the function it embodies can nevertheless be computed by a suitable program running on a universal computer. And for aspiring Assistant Professors in Computer Science, writing trial programs on a modern machine was much easier than doing empirical research into the brain's microanatomy and computational physiology. Hacking away at skill-specific programs became the normal form for research into AI. Von Neumann's book, effectively written on his deathbed, faded into obscurity, and was not reprinted for almost half a century. Most important of all, a great gulf was rationalized, and then fixed, between the empirical brain sciences and the discipline of "AI". Cross-fertilization fell close to zero. Neither discipline, it was agreed, had anything vital to teach the other. Neuroscience went on to flourish, despite that disconnection. But by the late 1970s, AI had begun to stagnate.

8.3 Turing's Real Legacy

Can we blame any of this on Turing himself? I think not. Despite being co-responsible for the Church–Turing thesis, Turing does not marshal the methodological arguments, scouted earlier, that counsel turning our backs on the biological brain as a fertile route into *identifying*, and subsequently recreating a recursive procedure for *generating*, the infinite function for conscious intelligence. Nor does he pretend that we already know, at least well enough, what that function is. On the contrary, Turing closes his famous paper (Turing, 1950), written in defense of the possibility of machine intelligence, by recommending that we simply *give up* the ambition of writing recursive procedures for adult human intelligence directly. He suggests, instead, that we approach the goal of such procedures in the same way that the biological brain does: via a long period in which it gradually *learns* the set of adult skills and capacities we seek to recreate. On this approach, a successful machine intelligence will acquire its sophisticated skills and capacities not in one fell swoop, from the program-downloading activity of some implausibly omniscient hacker. Rather, it will learn those capacities from its continual interactions with the world in which it has to live, just as brains do.

Guided by Turing, our aim has now shifted from writing the input–output program for an adult human, to writing the program for an *infant* human. The Church–Turing thesis provides the same guarantee, presumably, that a universal machine will once again be capable of computing the relevant computable function. That program, as Turing remarks, will have to be capable of some form of self-modification or self-modulation, if it is to be equal to the task of mastering the vast curriculum that confronts it. (This introduces some nice wrinkles because, like a leopard, a function strictly cannot change its spots: it is the same infinite set of ordered pairs at one time that it is at any other time. But that is no problem in principle. The input and output elements of any of its ordered pairs must now contain subelements, some of which index the current but modifiable state of the computing system. This leads us into the domain of what are called *dynamical systems*.)

Well and good, but on this importantly revised research program, something is dead obvious that was not so obvious on the classical research program that dominated AI from the 1960s to the 1980s. I complained a few pages back that classical researchers assumed, quite wrongly, that their target function was more or less *known* to them, at least in its important outlines. The justice or injustice of that complaint aside, no one will pretend that the target function for a newborn human *infant* is even remotely known, especially when that function includes one of the most poorly understood capacities in the entire human arsenal – the general capacity for learning itself. How can we hope to get a general grip on the elements of *that* infinite function, as a clear target for our attempts at recursive reconstruction, short of going directly to the empirical well of brain development, neuronal and synaptic plasticity, and general cognitive neurobiology? As urged earlier for the adult brain and the adult function, the infant brain is the uniquely authoritative source from which we might learn or recover the infant function. We know, going in, that the infant brain must embody a finite specification that allows it to compute the elements of that function, for compute them it does, right out of the box.

Turing's closing advice thus leads us back, immediately, to the very empirical coal-face that was deliberately forsaken by his presumptive intellectual heirs. I wish to suggest, therefore, that Turing's usual depiction, as the patron saint of the *classical* research program in AI, is simply a mistake. He is more accurately seen as the unsung patron saint of the more recent and biologically inspired program of research into *artificial neural networks*. This alternative, but now flourishing, approach attempts to find out both the *what* of the brain's abstract functional endeavors, and the *how* of their actual physical computation.

These artificial models portray the brain as a massively parallel *vector processor*, as a nonserial, nondigital computer that transforms high-dimensional input vectors (namely, the pattern of activation levels across a large population of neurons) into output vectors (patterns of activation across a downstream population of neurons), which ultimately control the body's muscle system. The vectors are transformed by the peculiar configuration of synaptic connections that both separate, and join, one neuronal population to another. Those vectors get appropriately transformed into new vectors, when they are forced to traverse the matrix of synaptic connections at issue. Collectively, those synaptic connections embody everything the creature has ever learned, a gradual process that involves the successive modification of the connection strengths at issue, modifications made in response to the creature's ongoing interactions with the environment.

This alternative picture is as computational as you please. It is another instance of Turing's basic theoretical insight into our capacity for conscious intelligence. Specifically, a well-trained brain embodies a finite specification of a potentially infinite range of input–output pairs – that is, a computable function – a finite specification that takes the form of computational procedures for the repeated transformation of inputs into outputs. This picture also addresses squarely the fundamental issue of how the brain *learns* (a matter mostly finessed by the classical tradition), just as Turing's closing discussion advises. The difference between the two traditions lies mainly in the kinds of representations involved, and the kinds of computational procedures deployed. But Turing would have welcomed high-dimensional vector/matrix processing as eagerly as any other computational device. I therefore suggest that the true heirs to Turing's basic theoretical suggestion are those who pursue the research tradition of artificial neural networks, and its fertile interaction with the empirical neurosciences. For that is where Turing's unambiguous advice, tendered as the conclusion to his most famous paper, now bids anyone go.

8.4 Reevaluating Turing's Behavioral Test

What is the status, from this reworked perspective, of Turing's (in)famous behavioral Test? (I shall here assume the reader's familiarity with it.) Certainly the computer's interactive behavior is *relevant* to the question of its conscious intelligence, in the way that an arbitrarily chosen finite subset of a given infinite set can at least occasionally *falsify* their ascription to some target function. On the other hand, being finite, the set of input–output pairs revealed during the Turing Test always *underdetermines* the claim that they belong to the target function, though they may provide some degree of corroboration. The claim of a successful reconstruction of conscious intelligence is therefore always and ever subject to future refutation. This is entirely normal, for any hypothesis, and Turing was entirely aware of it. The point behind *Turing's* sketch of the Test situation was surely to bring home to his readers that, if they choose to withhold the ascription of intelligence to a computing machine that "passes" his Test, they are *prima facie* denying the efficacy of the very same sorts of evidence that license that same ascription to normal humans. Dialectically speaking, this puts the ball in the doubter's court, who is thus invited to explain and justify this disparity in evidential treatment.

Readers will recall that, at that point in the article, Turing turns to canvass a long sequence of precisely such exemptive *apologias*, each of which he finds inadequate to blunt the initial presumption that sufficiently systematic intelligent behavior, despite its nonstandard computational source, still has its normal evidential relevance. As a dialectical strategy, this is entirely understandable, and it requires us to ascribe neither more nor less authority to his famous Test than we would ascribe to any other, inevitably fallible, inductive/abductive inference.

Is this as close as we can ever get to authoritatively identifying genuine instances of conscious intelligence? No. We can get closer. But in order to do so, we need to gain an understanding of what naturally occurring conscious intelligence *is*, an understanding that runs much deeper than we currently possess. In particular, we need to know what computational procedures the brain is actually deploying, so that we may have (1) a better grip on whatever infinite function it may be computing, and (2) a better understanding of how the output elements of that function are physically generated, within us, from its input elements. With such a deeper understanding safely in place, we can then address any novel candidate for the possession of conscious intelligence by examining its *internal computational procedures*, in order to get a more authoritative judgment on the identity of whatever abstract function it may be computing, and a more authoritative judgment on whether it deploys the same transformational *tactics*, as deployed in the human case, in order to compute that function.

These deeper probings, note well, will still leave us with an importantly ambiguous situation, a residual problem that lies behind – far behind – the value or legitimacy of the Turing Test. Specifically, in the ascription or denial of conscious intelligence to any novel physical system, which criterion should dominate: sameness of abstract function computed? Or (more stringently) sameness of computational procedures actually deployed in the generation of that function's elements? To my knowledge, Turing never came down firmly on *either* side of this question, despite the orthodox expectation that he would opt for the former position. For my part, I am inclined to embrace the latter position. This is not because I wish to exclude nonstandard critters from the ranks of the blessed. My concerns, indeed, are inclusive rather than exclusive. The fact is, no two of us normal humans are computing exactly the same abstract function. Its existence, as that which unifies us, is a myth. Individual variation in our cognitive profiles is ubiquitous, even worthy of celebration. But we *do* share relevantly identical arsenals of computational machinery: crudely, vector coding systems and matrix multiplication systems. What unites us all, in the end, is our sharing the same basic kinds of computational machinery.

That empirical machinery, and the endless forms of articulation it may find in various individuals and in various species, is the true subject of the cognitive sciences. If we seek the essence of our endlessly variable natural kind, that is where it lies – not in Plato's heaven, but inside the head.

References

Turing, A. M., 1950, Computing machinery and intelligence, *Mind* **59**: 433–60.

von Neumann, John, 1958, *The Computer and the Brain*, Yale University Press, New Haven, CT, new edition (2000).

Chapter 9
Turing's Test

A Philosophical and Historical Guide

Jack Copeland and Diane Proudfoot

Abstract We set the Turing Test in the historical context of the development of machine intelligence, describe the different forms of the test and its rationale, and counter common misinterpretations and objections. Recently published material by Turing casts fresh light on his thinking.

Keywords Artificial Intelligence, automatic computing engine, bombe, Chinese Room, cryptanalysis, enigma, intelligence, Turing, Turing machine, Turing Test

9.1 Introduction: Turing and Machine Intelligence

Turing was thinking about machine intelligence (now called 'artificial intelligence' or AI) at least as early as 1941. A typewritten paper of his on machine intelligence, circulated among some of his wartime colleagues at the Government Code and Cypher School (GC & CS) at Bletchley Park,[1] was undoubtedly the earliest paper in the field of AI. Now lost, it probably concerned machine learning and heuristic problem-solving, since Turing discussed both extensively during the war years at GC & CS. (Heuristics are rules that cut down the amount of searching required to solve a problem. Turing's Bombe – the electromechanical computing device used by GC & CS – broke German Enigma messages by means of a heuristic search through possible settings of the Enigma machine.) During the war Turing also theorized about mechanical chess, stating in 1945 that a computer 'could probably be made to play very good chess' (Turing, 1945: 389).

In 1936 Turing had dreamed up an abstract digital computing machine – the 'universal Turing machine' – that is the origin of the modern (stored-program

University of Canterbury

[1] Donald Michie in personal communication with Copeland, 1998.

R. Epstein et al. (eds.), *Parsing the Turing Test.*

digital) computer (Turing, 1936: 393). When, at the end of the war, Turing had the opportunity actually to build such a device, his interest was very much in machine intelligence. His 'Proposed electronic calculator' (Turing, 1945), written at the National Physical Laboratory (NPL) in London, was the first relatively complete specification of an electronic stored-program general-purpose digital computer.[2] In working on the ACE, Turing described himself as 'building a brain'[3] and remarked (in a letter to W. Ross Ashby) that he was 'more interested in the possibility of producing models of the action of the brain than in the practical applications to computing'.[4]

In 1947 Turing gave what was, so far as is known, the earliest public lecture to mention computer intelligence, saying 'What we want is a machine that can learn from experience' and '[t]he possibility of letting the machine alter its own instructions provides the mechanism for this' (Turing, 1947: 393). In 1948 he wrote a report for the NPL entitled 'Intelligent Machinery'. In it Turing described some experiments concerning the modification of an initially 'unorganised machine', by a process analogous to teaching by reward and punishment (Turing, 1948: 425–429). He later called this machine a 'child-machine' (Turing, 1950a: 457), saying that he had 'succeeded in teaching it a few things, but the teaching method was too unorthodox for the experiment to be considered really successful' (ibid.: 457).

In the same report Turing anticipated many ideas now central to AI. These included the theorem-proving approach to problem-solving, the hypothesis that 'intellectual activity consists mainly of various kinds of search,', and the idea of (what is now called) 'evolutionary computing', where programs are developed in a manner reminiscent of evolution by natural selection, the 'criterion being survival value' (Turing, 1948: 431). The report also contains probably the first suggestion that computing machines could be built out of simple, neuron-like elements connected together into networks in a largely random manner. (We call these networks 'Turing Nets' (Copeland and Proudfoot, 1996, 1999).) Turing suggested that initially random Turing Nets be 'trained' (his term) to perform specific tasks. In this respect Turing anticipated the approach now known as 'connectionism'. He also described a certain form of Turing Net as 'about the simplest model of a nervous system with a random arrangement of neurons' and hypothesized that 'the cortex of the [human] infant is an unorganized machine, which can be organized by suitable interfering training' (Turing, 1948: 418, 424). He claimed a proof (now lost) that an initially unorganized Turing Net with a sufficient number of neurons can be organized to become a universal Turing machine with a given storage capacity

[2] Von Neumann's (1945) 'First Draft of a Report on the EDVAC' (Electronic Discrete Variable Calculator) appeared in the USA some months before Turing's 'Proposed electronic calculator' but contained little engineering detail, in particular with regard to electronic hardware. In contrast, Turing provided detailed circuit designs and detailed specifications of hardware units, specimen programs in machine code, and even an estimate of the cost of building the ACE.

[3] Donald Bayley in personal communication with Copeland, 1997.

[4] The letter, which is undated, is among the Woodger Papers in the National Museum.

(ibid.: 422). This proof first opened up the possibility, noted by Turing (ibid.: 424), that the human brain is a universal symbol-processor implemented in a neural network.

The 1948 report also describes a restricted form of what Turing later called the 'imitation game':

> It is possible to do a little experiment … even at the present stage of knowledge. It is not difficult to devise a paper machine [a program written out for a human being to follow with the aid of paper and pencil] which will play a not very bad game of chess. Now get three men as subjects for the experiment, A, B, and C. A and C are to be rather poor chess players. B is the operator who works the paper machine. (In order that he should be able to work it fairly fast it is advisable that he be both mathematician and chess player.) Two rooms are used with some arrangement for communicating moves, and a game is played between C and either A or the paper machine. C may find it quite difficult to tell which he is playing. (This is a rather idealized form of an experiment I have actually done.) (1948: 431)

Turing's views on machine intelligence influenced the first AI programmers. Both during and after the war Turing experimented with machine routines for playing chess (the machine's behaviour was simulated by a human using paper and pencil). He began programming his chess-playing routine 'Turochamp' (devised in 1948 with David Champernowne) for the Ferranti Mark I computer – the world's first commercially available electronic stored-program digital computer, built by the Manchester firm of Ferranti in conjunction with the University of Manchester Computing Machine Laboratory – but did not complete the task (Michie, 1966: 189). The first chess program to be implemented, written by Dietrich Prinz, ran on the Ferranti Mark I in November 1951 (Prinz, 1952). Unlike Prinz's program, the 'Turochamp' could play a complete game and operated by heuristics rather than an exhaustive search.[5] In 1953 Turing published a classic article on chess programming (Turing 1953).

The earliest AI program to use heuristic search (apart from Turing's own 'paper' chess-player) was a draughts (checkers) player, written by Christopher Strachey in May 1951 for the Pilot Model of the ACE at the NPL. In a letter to Turing on the evening of Turing's May 1951 radio broadcast 'Can Digital Computers Think?' (Turing, 1951a),[6] Strachey wrote: 'your remark … that the programme for making a machine think would probably have great similarities with the process of teaching … seems to me absolutely fundamental'.[7] When his program did not run satisfactorily on the Pilot ACE, Strachey moved his project to the University of Manchester Computing Machine Laboratory, where Turing was Deputy Director (there being no Director). With Turing's encouragement, and using Turing's recently completed

---

[5] Letter from Champernowne in *Computer Chess* **4** (January 1980): 80–81, reprinted in Copeland 2004, pp. 563–564.

[6] The full text of this broadcast, and also of Turing's 1952 radio broadcast, was published for the first time by Copeland (1999).

[7] Strachey to Turing, 15 May 1951; the letter is in the Modern Archive Centre, King's College, Cambridge (catalogue reference D5).

Programmers' Handbook (Turing, 1950b) for the Ferranti Mark I, Strachey got his program to work. By the summer of 1952 the program could play a complete game of draughts at a reasonable speed (Strachey, 1952).[8] Strachey's program used simple heuristics and looked ahead 3–4 turns of play. The state of the board was represented on the face of a cathode ray tube – one of the earliest uses of computer graphics.

The earliest functioning AI programs to incorporate learning were written by Anthony Oettinger in 1951 for the EDSAC at the University of Cambridge Mathematical Laboratory. (The EDSAC was the world's second working electronic stored-program digital computer (1949), the first being the Manchester University 'Baby' in June 1948.) Oettinger was considerably influenced by Turing's views on machine learning. Oettinger's 'response-learning programme' could be taught to respond appropriately to given stimuli, by means of expressions of 'approval' or 'disapproval' by the teacher. In his 1948 report Turing had said that we do not call a machine intelligent if we find 'out the rules of its behaviour' (Turing, 1948: 431). Oettinger claimed that the 'behaviour pattern of the response-learning … machine is sufficiently complex to provide a difficult task for an observer required to discover the mechanism by which the behaviour of the … machine is determined' (Oettinger, 1952: 1257).

Using Turing's terminology, Oettinger described another learning program, his 'shopping programme', as a 'child machine'. This program (which we call 'Shopper') simulates the behaviour of 'a small child sent on a shopping tour' (Oettinger, 1952: 1247). In its simulated world of eight shops, Shopper is instructed to buy some item, and searches for it at random. While searching, Shopper memorizes a few of the items stocked in each shop that it visits; if sent out again for the original item, or for some other item whose location it has learned during previous searches, it goes directly to the appropriate shop.

Oettinger was the first programmer to claim a program able to pass a (highly circumscribed!) Turing Test – one, that is, in which 'the questions are restricted to … the form "In what shop may article *j* be found?"' (Oettinger, 1952: 1250).

In his final years Turing worked on (what would now be called) Artificial Life. He was the first person to use computer simulation to investigate naturally occurring biological life. He used the Ferranti Mark I to model biological growth and described this research as 'not altogether unconnected' to his work on neural networks (in a letter to J. Z. Young dated 8 February 1951).[9] His work on biological growth would, he hoped, assist him in developing 'a fairly definite theory' of 'the genetical embryological mechanism' by which 'brain structure … [is] achieved' (ibid.).

[8] The first AI program to function in the USA was a checkers player for the IBM 701, coded by Arthur Samuel. In 1952 Strachey described his program at a computing conference in North America and Samuel took over its essentials (Samuel, 1959).

[9] An extract from Turing's letter is printed in Copeland 2004, p. 517. A copy of the letter (typed by his mother, Sara Turing) is in the Modern Archive Centre, King's College, Cambridge (catalogue reference K1.78).

9.2 The 1950 Presentation of the Turing Test

9.2.1 The Form of the Test

In his 1950 paper, 'Computing Machinery and Intelligence', Turing introduced the test by first describing an imitation game involving an interrogator and two subjects, one male (A) and one female (B). The interrogator communicates with A and B from a separate room (nowadays this would probably be by means of a keyboard and screen); apart from this the three participants have no contact with each other. The interrogator's task is to find out, by asking questions, which of A and B is the man; A's aim is that the interrogator make the wrong identification.

Turing then asked 'What will happen when a machine takes the part of A in this game?' (Turing, 1950a: 434). The interrogator's task is now to discover which of A or B is the computer; to do so he or she is permitted to ask any question, on any topic. The computer is allowed to do everything possible to force a wrong identification. As to the human foil, Turing said, 'The object of the game for the third player (B) is to help the interrogator. The best strategy for her is probably to give truthful answers.' (ibid.: 434).

Having described the computer-imitates-human game, Turing remarked:

> [T]he … question, 'Can machines think?' I believe to be too meaningless to deserve discussion (Turing, 1950a: 442)

and

> I shall replace the question ['Can machines think?'] by another ['Are there imaginable digital computers which would do well in the imitation game?'], which is closely related to it and is expressed in relatively unambiguous words. (ibid.: 433)[10]

However, Turing also described this now notorious proposal as a 'tentative' suggestion (ibid.: 442) and remarked that:

> [W]e cannot altogether abandon the original form of the problem [viz. 'Can machines think?'], for opinions will differ as to the appropriateness of the substitution and we must at least listen to what has to be said in this connexion. (ibid.: 442)

9.2.2 Scoring the Test

To assess the computer's performance, we ask:

> Will the interrogator decide wrongly as often when the [computer-imitates-human] game is played … as he does when the game is played between a man and a woman? (Turing, 1950a: 434)

[10] Some theorists claim that we use a version of the Turing Test to establish that other human beings think, and some, e.g. (Leiber, 1991), even attribute this view to Turing. According to Turing, there is a 'polite convention that everybody thinks' (Turing, 1950a: 446).

If the computer (in the computer-imitates-human game) does no worse than the man (in the man-imitates-woman game), it passes the test.

The role of the man-imitates-woman game is frequently misunderstood. For example, Hodges, Turing's biographer, claims that this game is, as an introduction to the Turing Test, irrelevant – a 'red herring' (Hodges, 1992: 415). However, the man-imitates-woman game is not intended as an *introduction* to the test at all, but rather as part of the protocol for scoring the test.

9.2.3 The OIG Test

When Turing, in the 1950 paper, introduced the computer into the imitation game, he did so in two slightly different ways. In one passage he stated only that 'a machine takes the part of A' (Turing, 1950a: 434), but in another passage he said that the part of A is taken by a machine and 'the part of B … by a man' (ibid.: 442). This difference has led Sterrett (2000) to distinguish between what she calls the 'original imitation game test' (the 'OIG test') and what she calls the 'standard Turing test' (recall that in Turing's man-woman version of the imitation game, A is male and B is female). Sterrett claims that the OIG test is described in the first of Turing's remarks just quoted. In the OIG test, the computer attempts to impersonate a *woman* and its degree of success is compared with a male player's degree of success at the same task. In the standard Turing test, the computer attempts simply to impersonate a *human being*. Sterrett argues that the OIG test is superior, and Traiger (2000) argues that Turing was in fact advocating the OIG test.

However, in 1952 Turing said that '[t]he idea of the test is that the machine has to try and pretend to be a man … and it will pass only if the pretence is reasonably convincing' (Turing, 1952: 495). This suggests that Turing's intention in 1950 was not that of endorsing only the OIG test. Moreover, in his May 1951 radio lecture Turing presented matters in a starkly un-gendered form: the point of the test is to determine whether or not a computer can 'imitate a brain' (Turing, 1951a: 435). It seems unlikely, therefore, that Turing saw himself as describing different tests in the relevant passages of the 1950 paper.

9.2.4 Turing's Principle

According to Turing, a machine that (by means of calculation) imitates the intellectual behaviour of a human brain can itself properly be described as a brain or as thinking. (Turing also used the verb 'simulate': 'My contention is that machines can be constructed which will simulate the behaviour of the human mind very closely' (Turing, 1951b: 472).) We call this *Turing's principle* (Copeland, 2000b). It appears in Turing's 1951 radio lecture:

> I believe that [digital computers] could be used in such a manner that they could appropriately be described as brains. … This … statement needs some explanation. … In order to

arrange for our computer to imitate a given machine it is only necessary to programme the computer to calculate what the machine in question would do under given circumstances. … If now some particular machine can be described as a brain we have only to programme our digital computer to imitate it and it will also be a brain. If it is accepted that real brains, as found in animals, and in particular in men, are a sort of machine it will follow that our digital computer suitably programmed will behave like a brain. This argument involves [the assumption] which can quite reasonably be challenged … that this machine should be of the sort whose behaviour is in principle predictable by calculation. …

[O]ur main problem [is] how to programme a machine to imitate the brain, or as we might say more briefly, if less accurately, to think. … The fact is that we know very little about [how to do this]. … The whole thinking process is still rather mysterious to us, but I believe that the attempt to make a thinking machine will help us greatly in finding out how we think ourselves.[11] (Turing, 1951a: 482–483, 485–486)

Only the intellectual behaviour of the brain need be imitated:

[W]e need not consider such irrelevant matters as the faithfulness with which the human voice can be imitated. (Turing, 1951a)

Turing believed that 'something like a viva-voce examination' (ibid.: 484) was a suitable method for testing a machine's ability to imitate human intellectual behaviour, for this method 'draw[s] a fairly sharp line between the physical and the intellectual capacities of a man' and 'seems to be suitable for introducing almost any one of the fields of human endeavour that we wish to include' (Turing, 1950a: 434–435).

The combination of Turing's principle and the suitability of the question-and-answer method explains why Turing thought it appropriate to replace the question 'Can machines think?' with the question 'Are there imaginable digital computers which would do well in the imitation game?'.[12]

9.2.5 The Reasoning Behind Turing's Principle

Turing offered no explicit justification of his principle. In the 1950 paper he remarked:

I have no very convincing arguments of a positive nature to support my views. If I had I should not have taken such pains to point out the fallacies in contrary views. (Turing, 1950a: 454)

---

[11] For further discussion of Turing's 1951 radio lecture see (Copeland, 2000a).

[12] The claim that the brain is equivalent to a Turing machine – which is much stronger than the claim that a Turing machine can imitate the brain see (Copeland, 1999) – is apt to appear self-evident, an unassailable corollary of the Church-Turing thesis see (Shanker, 1987; Leiber, 1991; Newell, 1980; Searle, 1992). However, one of Turing's most important but least appreciated achievements was to show how the brain could fail to be equivalent to a Turing machine see (Copeland, 1994, 2000, 2002b; Proudfoot and Copeland, 1994). Turing's 1936 proof of the existence of mathematical functions that are not computable by the universal Turing machine guarantees the existence of an (abstract) model of neuronal function that is not equivalent to any Turing machine. Whether or not the brain is equivalent to a Turing machine is an empirical matter.

And elsewhere, he spoke merely of the need to give ‘fair play’ to machines (e.g. Turing, 1947: 394). However, when he introduced the restricted form of the imitation game in his 1948 report, Turing offered what looks like a rationale. He spoke of intelligence as an ‘emotional concept’ and said:

> The extent to which we regard something as behaving in an intelligent manner is determined as much by our own state of mind and training as by the properties of the object under consideration. If we are able to explain and predict its behaviour or if there seems to be little underlying plan, we have little temptation to imagine intelligence. With the same object therefore it is possible that one man would consider it as intelligent and another would not; the second man would have found out the rules of its behaviour. (Turing, 1948: 431)

Turing went so far as to say that ‘one might be tempted to define thinking as consisting of “those mental processes that we don’t understand”’ (Turing, 1952: 500). (Some contemporary philosophers have argued for a similar view; e.g., Minsky likens intelligence to the concept *the unexplored regions of Africa* – ‘it disappears as soon as we discover it’ (Minsky, 1988: 71).)

9.3 The 1952 Presentation of the Test

9.3.1 The Form of the Test

In a BBC radio broadcast entitled ‘Can automatic calculating machines be said to think?’, recorded in January 1952, Turing remarked:

> I don’t want to give a definition of thinking, but if I had to I should probably be unable to say anything more about it than that it was a sort of buzzing that went on inside my head. But I don’t really see that we need to agree on a definition at all. The important thing is to try to draw a line between the properties of a brain, or of a man, that we want to discuss, and those that we don’t. To take an extreme case, we are not interested in the fact that the brain has the consistency of cold porridge. We don’t want to say ‘This machine’s quite hard, so it isn’t a brain, and so it can’t think.’ I would like to suggest a particular kind of *test* that one might apply to a machine. You might call it a test to see whether the machine thinks, but it would be better to avoid begging the question, and say that the machines that pass are (let’s say) ‘Grade A’ machines. The idea of the test is that the machine has to try and pretend to be a man, by answering questions put to it, and it will only pass if the pretence is reasonably convincing. A considerable proportion of a jury, who should not be expert about machines, must be taken in by the pretence. They aren’t allowed to see the machine itself – that would make it too easy. So the machine is kept in a far away room and the jury are allowed to ask it questions, which are transmitted through to it: it sends back a typewritten answer. … [The questions can concern] anything. And the questions don’t really have to be questions, any more than questions in a law court are really questions. You know the sort of thing. ‘I put it to you that you are only pretending to be a man’ would be quite in order. Likewise the machine would be permitted all sorts of tricks so as to appear more man-like, such as waiting a bit before giving the answer, or making spelling mistakes, but it can’t make smudges on the paper, any more than one can send smudges by telegraph. We had better suppose that each jury has to judge quite a number of times, and that sometimes they really are dealing with a man and not a machine. That will prevent them saying ‘It must be a machine’ every time without proper consideration.

> Well, that's my test. Of course I am not saying at present either that machines really could pass the test, or that they couldn't. My suggestion is just that this is the question we should discuss. It's not the same as 'Do machines think', but it seems near enough for our present purpose, and raises much the same difficulties. (Turing, 1952: 494–495)

The 1950 and 1952 presentations of the test are significantly different. According to the 1950 formulation, the Turing Test is a three-party game involving the parallel interrogation by a human of a computer and a human foil. It is implied that the interrogator knows that one of each pair of contestants is a human and one a machine. According to the 1952 formulation, members of a jury question a series of contestants one by one; some of the contestants are machines and some humans. It is implied that the interrogators do not know the ratio of machines to humans. (The arrangements employed in the annual Loebner Turing Test Competition, sometimes described as illegitimate, in fact conform in these respects to the 1952 presentation.)

The test as presented in 1950 is harder for the computer to pass. This is because some small feature of the computer's performance which each interrogator might overlook in a non-competitive situation might be the decisive factor in the competitive one.

The test as formulated in 1950 is, nevertheless, fairer to the machine. It appears that interrogators are determined not to be fooled by a program; to avoid this outcome, when presented with contestants one by one, interrogators tend to classify contestants as machines. For example, in the 2000 Loebner Turing Test Competition no machine was mistaken for a human being, but on ten occasions a human was judged to be a machine (Moor, 2003). In the 2003 competition on no occasion was a machine judged 'definitely a human' (and only once 'probably a human') but on four occasions a human was judged 'definitely a machine'.[13]

9.3.2 Did Turing Offer a Definition of 'Thinking'?

The Turing Test is commonly interpreted as providing a definition of 'thinking' or 'intelligence'. For example, French claims that '[t]he Turing Test [was] originally proposed as a simple operational definition of intelligence' (French, 2000: 115). Hodges claims that Turing 'introduced … an operational definition of "thinking" or "intelligence" … by means of a sexual guessing game' (Hodges, 1992: 415). Block writes:

> An especially influential behaviorist definition of intelligence was put forward by Turing [1950a]. … Turing's version of behaviorism formulates the issue of whether machines could think or be intelligent in terms of whether they could pass the following test. … The computer is intelligent *if and only if* the judge cannot tell the difference between the computer and the person. (Block, 1990: 248, our italics)

[13] http://www.surrey.ac.uk/dwrc/loebner/results.html.

In a footnote, Block mentions Turing's suggestion in the 1950 paper that machines may 'carry out something which ought to be described as thinking but which is very different from what a man does' (Turing, 1950a: 435), and claims that Turing 'jettisoned the claim that being able to pass the Turing Test is a necessary condition of intelligence, weakening his claim to: passing the Turing Test is a sufficient condition for intelligence' (Block, 1990: 249–250).

However, as the quotation in 3.1 shows, Turing explicitly denied that he was proposing a definition of 'thinking' or 'intelligence'. Turing never claimed that the ability to pass the Turing Test is a necessary condition for intelligence[14]; in consequence, Block's description of Turing as 'jettisoning' such a condition and 'weakening' his claim is entirely misleading.

9.3.3 Is the Imitation Game Really a Test?

Some commentators claim that Turing did not intend his imitation game as a *test*. For example, Whitby complains about 'the change from the label "imitation game" to "Turing test" by commentators' and claims that the 'suggestion that [the imitation game] might be some sort of test involves an important extension of Turing's claims' (Whitby, 1996: 54). Similarly, Narayanan asserts that 'Turing did not originally intend his imitation game to be a test as such' (Narayanan, 1996: 66). However, the 1952 presentation makes it clear that Turing did indeed intend to propose a test. Even in the 1950 paper Turing spoke of the imitation game as a 'test' and claimed that some of his opponents would 'probably be willing to accept our test' (Turing, 1950a: 446, 447).

9.4 Objections to the Turing Test

We consider six objections that are especially prominent.

9.4.1 Anthropocentrism

One obvious criticism of the Turing Test is that it is anthropocentric, in that it requires a computer to be capable of producing outward behaviour indistinguishable from that of a human being. For example, French (arguing against Turing's

[14] In his classic paper on the Turing Test, Moor makes this point:

> [T]he proponents and critics of the imitation game have misunderstood its significance. The real value of the imitation game lies not in treating it as the basis for an operational definition but in considering it as a potential source of good inductive evidence for the hypothesis that machines think (Moor, 1976: 249). See also (Moor, 1987).

supposed operational definition of 'intelligence') objects that 'the Test provides a guarantee not of intelligence but of culturally oriented *human* intelligence. … [T]he Turing Test [is] a test for *human* intelligence, not intelligence in general' (French, 1990: 12). Turing mentioned the same objection:

> The game may perhaps be criticized on the ground that the odds are weighted too heavily against the machine. … May not machines carry out something which ought to be described as thinking but which is very different from what a man does? (Turing, 1950a: 435)

However, it is clear from the remark just quoted that Turing could hardly have intended the imitation game as a test for 'intelligence in general'; it was obvious to him that an intelligent machine may fail the test just because its behaviour is distinctly non-human. The anthropocentrism objection simply misses the point: Turing intended the imitation game precisely as a means of testing whether or not a given machine emulates the (intellectual behaviour of the) human brain.

It is important to keep in mind in this connection that the imitation game addresses only the general question 'Can *machines* think?', and not every particular question of the form 'Can machine *M* think?'. Turing's proposal was that we replace the question 'Can machines think?' by the question 'Are there imaginable digital computers which would do well in the imitation game?'; he did not propose that we replace 'Can machine *M* think?' by 'Does machine *M* do well in the imitation game?'. That machine *M* does badly in the imitation game does not show that it does not think.

If *no* machine does well in the imitation game, then the question 'Can machines think?' remains open, and must be settled by some other means. However, Turing cheerfully set aside this possibility, assuming that *some* machine would succeed in the game:

> [The anthropocentrism] objection is a very strong one, but at least we can say that if, nevertheless, a machine can be constructed to play the imitation game satisfactorily, we need not be troubled by this objection. (Turing, 1950a: 435)

9.4.2 The Shannon–McCarthy Objection

In 1956 Shannon and McCarthy offered this objection:

> The problem of giving a precise definition to the concept of 'thinking' and of deciding whether or not a given machine is capable of thinking has aroused a great deal of heated discussion. One interesting definition has been proposed by A. M. Turing: a machine is termed capable of thinking if it can, under certain prescribed conditions, imitate a human being by answering questions sufficiently well to deceive a human questioner for a reasonable period of time. A definition of this type has the advantages of being operational, or, in the psychologists' term, behavioristic. … A disadvantage of the Turing definition of thinking is that it is possible, in principle, to design a machine with a complete set of arbitrarily chosen responses to all possible input stimuli. … Such a machine, in a sense, for any given input situation (including past history) merely looks up in a 'dictionary' the appropriate response. With a suitable dictionary such a machine would surely satisfy Turing's definition

> but does not reflect our usual intuitive concept of thinking. This suggests that a more fundamental definition must involve something relating to the manner in which the machine arrives at its responses – something which corresponds to differentiating between a person who solves a problem by thinking it out and one who has previously memorized the answer. (Shannon and McCarthy, 1956: v–vi)

This objection has occurred to a number of writers but nowadays is usually credited to Block (1981). A 'blockhead' is a (hypothetical) program able to play the imitation game successfully, for any fixed length of time, by virtue of including a look-up table. This large, but finite, table contains all the exchanges between program and interrogator that could occur during the length of time in question. Such a program faithfully imitates the intellectual behaviour of the brain but, so the objection goes, does not think.

The formal point on which the Shannon–McCarthy objection rests – the encapsulability in a look-up table of all the relevant behaviour – would have been obvious to Turing. In his 1950 paper, Turing pointed out that the behaviour of any 'discrete state machine' (i.e. a machine 'which move[s] by sudden jumps or clicks from one quite definite state to another' (Turing, 1950a: 439)) can be represented by a finite look-up table if the machine has a finite number of states, and that a computer can mimic the machine if supplied with the table:

> [D]iscrete state machines … can be described by such tables provided they have only a finite number of possible states. … Given the table corresponding to a discrete state machine … [and provided the calculation] could be carried out sufficiently quickly the digital computer could mimic the behaviour of [the] discrete state machine. The imitation game could then be played with the machine in question (as B) and the mimicking digital computer (as A) and the interrogator would be unable to distinguish them. Of course the digital computer must have an adequate storage capacity as well as working sufficiently fast. (Turing, 1950a: 440–441)

Storage capacity and speed are crucial. In the 1952 radio broadcast the mathematician Max Newman[15] remarked:

> It is all very well to say that a machine could … be made to do this or that, but, to take only one practical point, what about the time it would take to do it? It would only take an hour or two to make up a routine to make our Manchester machine analyse all possible variations of the game of chess right out, and find the best move that way – *if* you didn't mind its taking thousands of millions of years to run through the routine. Solving a problem on the machine doesn't mean finding a way to do it between now and eternity, but within a reasonable time.

To this Turing replied:

> To my mind this time factor is the one question which will involve all the real technical difficulty. If one didn't know already that these things can be done by brains within a reasonable

[15] In his lectures on Hilbert's *Entscheidungsproblem* (in Cambridge in 1935) Newman introduced Turing to the idea that led him to the concept of the Turing machine: Newman defined a 'constructive process' as one that a machine can carry out. During the war Newman and Turing cooperated closely at Bletchley Park; there Newman began the machine-based decryption project that led to Colossus, the world's first large-scale electronic digital computer. At the end of the war Newman established the Royal Society Computing Machine Laboratory at Manchester University, luring Turing there in 1948.

> time one might think it hopeless to try with a machine. The fact that a brain *can* do it seems to suggest that the difficulties may not really be so bad as they now seem. (Turing, 1952: 503–504)

Turing's remarks suggest the following reply to the Shannon–McCarthy objection. First, given practical limitations on storage capacity, the hypothetical 'machine with a complete set of arbitrarily chosen responses to all possible input stimuli' simply cannot be built. Second, even if such a machine could actually be constructed, it would *not* faithfully imitate the intellectual behaviour of the brain, since what the brain can do in minutes would take this machine 'thousands of millions of years.'

Assuming that blockheads do not think, the Shannon–McCarthy objection establishes that, in a possible world in which a blockhead takes only seconds to answer the interrogator's questions, the proposition 'If *M* does well in the imitation game, then *M* thinks' is false. (Such a world is, of course, very different from the actual world.) Hence the objection would succeed if, as Shannon and McCarthy believed, Turing intended the test to provide a definition of 'thinking' – because, in that case, Turing would indeed have had to say that 'If machine *M* does well in the imitation game, then *M* thinks' is true in *all* possible worlds. (Just as, if 'bachelor' is defined as 'unmarried male of marriageable age', it is true not only in the actual world, but in every possible world, that if *x* is an unmarried male of marriageable age, then *x* is a bachelor.) At bottom, then, the Shannon–McCarthy objection depends on the interpretative mistake of taking Turing to have proposed a definition.

There is no textual evidence to indicate that Turing was claiming anything more than that 'If machine *M* does well in the imitation game, then *M* thinks' is *actually* true, that is, true in the actual world. Nor did Turing need to claim more than this in order to advocate the imitation game as a satisfactory real world test.

9.4.3 The Chinese Room

Searle's (1980) putative counterexample to the Turing Test is famous. A hypothetical human clerk (Clerk), who is a monolingual speaker of English, 'handworks' a computer program that is capable of passing the Turing Test in Chinese. Clerk works in a room that is hidden from the interrogator (who is perhaps a native Chinese speaker); the interrogator and Clerk communicate by passing sheets of paper through a slot. The interrogator asks questions in Chinese, Clerk consults rule-books (written in English and containing the program) and produces answers in Chinese. To the interrogator, the verbal behaviour of the 'Room' – the system that includes the rule-books, Clerk, Clerk's paper, pencils, and rubbers, the input and output slots, and any clock, random number generator, or other equipment that Clerk may need in order to execute the program in question – is (we are to suppose) indistinguishable from that of a native Chinese speaker.

Searle claims that Clerk 'can pass the Turing Test … can fool native Chinese speakers' (Searle, 1980: 419) but that

> [Clerk] do[es] not understand a word of the Chinese. … [Clerk] ha[s] inputs and outputs that are indistinguishable from those of the native Chinese speaker, and [Clerk] can have any formal program you like, but [Clerk] still understand[s] nothing. For the same reasons, [the] computer understands nothing …
>
> [W]hatever purely formal principles you put into the computer, they will not be sufficient for understanding, since a human will be able to follow the formal principles without understanding. (ibid.: 418)

If Clerk passes the Turing Test but does not understand Chinese, passing the Turing Test is insufficient for understanding.

However, Searle's objection is flawed: his argument is invalid. It is the Room, not Clerk, that passes the Turing Test. And even if the formal operations carried out by Clerk do not enable *Clerk* to understand the Chinese inputs and outputs, it does not follow from this that the formal operations carried out by Clerk do not enable the *Room* to understand the Chinese inputs and outputs. One might as well claim that the statement 'The organization of which Clerk is a part has not filed for bankruptcy' follows from the statement 'Clerk has not filed for bankruptcy.' (This response to Searle's objection – the 'logical reply' (Copeland, 1993, 2002a) – is not the assertion that the Room *does* understand Chinese. Searle calls the latter assertion the 'systems reply' and correctly points out that such a reply 'begs the question by insisting without argument that the system must understand Chinese' (Searle, 1980: 419).)

9.4.4 Fiendish Expert Objections

Fiendish expert objections are all of the form: 'An expert could unmask the computer by asking it.' For example, it is sometimes pointed out (by, among others, Lenat in a paper at the 2000 Loebner Turing Test Competition) that an expert could use recent discoveries of characteristic weaknesses in human reasoning to unmask a computer. These discoveries include the facts that in some circumstances human beings fail to notice certain disconfirming instances of conditional statements and even assign a higher probability to a conjunction (e.g., 'Linda is a bank teller and a feminist') than to one of the conjuncts ('Linda is a bank teller'). An imitation-game interrogator could test for such weaknesses and easily detect any computer not specifically programmed to reproduce them.

However, Turing anticipated this type of objection. Although the Turing Test interrogator is permitted to ask any question she likes, not just any interrogator is permitted. In the restricted chess-player version of the test outlined in his 1948 report (quoted in Section 9.1), Turing stipulated that the interrogator be 'rather poor' at chess. In the 1952 presentation of the test, Turing stated that the jury 'should not be expert about machines'. It is likely that, had Turing been writing after the relevant experimental work by Wason, Johnson-Laird, Tversky, Kahneman, and others, he would also have excluded interrogators who are expert about human psychology.

9.4.5 Human Experience as Crucial: Associative Priming and Rating Games

French claims that, since the imitation game 'provides a very powerful means of probing humanlike cognition', only 'a machine capable of *experiencing the world in a manner indistinguishable from a human being*' is likely to succeed in the game (French, 1990: 25, 15, our italics). Hence French 'take[s] issue with' Turing's view 'that in the not-too-distant future it [will] in fact be possible actually to build … a machine' that plays the imitation game successfully (ibid.: 12, 11).

To illustrate his claim, French employs the following examples. First, associative priming. In word/non-word recognition tasks, human subjects determine more quickly that an item is a word if presentation of the item is preceded by presentation of an associated word (e.g., prior presentation of 'bread' facilitates recognition that 'butter' is a word). According to French, the Turing Test interrogator can exploit this phenomenon, as follows:

> The day before the Test, [the interrogator] selects a set of words (and non-words), runs the lexical decision task on the interviewees and records average recognition times. She then comes to the Test armed with the results … [and] identifies as the human being the candidate whose results more closely resemble the average results produced by her sample population of interviewees. The machine would invariably fail this type of test because there is no a priori way of determining associative strengths. … Virtually the only way a machine could determine, even on average, all of the associative strengths between human concepts is to have experienced the world as the human candidate and the interviewers had. (ibid.: 17)

This is, of course, another fiendish expert objection. But French's proposal is improper on another ground. The specifications of the Turing Test are clear: the interrogator is only allowed to ask questions. There is no scope for her to use timing mechanisms in order to measure the contestants' reaction times. One might as well allow the introduction of apparatus to measure the contestants' magnetic fields or energy dissipation.

French's second example involves what he calls 'rating games'. In a rating game, one asks questions such as: on a scale of 0 (completely implausible) to 10 (completely plausible), rate '"Flugbloggs" as a name Kellogg's would give to a new breakfast cereal', rate '"Flugly" as the surname of a glamorous female movie star', 'rate banana splits as medicine' (ibid.: 18, 21). According to French, such questions probe the candidates' 'subconscious associative network … that consists of highly overlapping activatable representations of experience' (ibid.: 16). He claims that a computer will do badly in a rating game.

Another fiendish expert objection. But in any case French's rating-game questions may not help the interrogator, since the computer may try to disguise itself as a human who would be expected to do badly in such a game – for example, a member of a foreign culture, such as a tourist from rural Japan on his or her first trip overseas. French claims that it is 'tacit in Turing's [1950] article' (ibid.: 15) that the computer must attempt to pass itself off as a member of the interrogator's own culture. However, French provides no textual evidence for this claim and it hardly seems likely that Turing would tacitly have imposed a restriction which makes the test more difficult for the computer yet does not appear to offer any conceptual gain. In fact, in the 1952 presentation of the test Turing makes it clear that the computer is to 'be permitted all sorts of tricks so as to appear more man-like'.[16] (French's only source is Turing's 1950 paper.)

French appears to think that connectionist devices may have an edge in rating games. However, insofar as French's sample questions have any one thing in common, it is that most test the contestants' common-sense knowledge. Viewed in this light, French's rating games fail, for the most part, to provide any new challenge to standard computationalism. Nor can he assume that only connectionist devices will perform satisfactorily in these games. No one knows how well a conventional computer equipped with a massive store of common-sense knowledge, such as the unfinished Cyc, might perform in rating games. (The Cyc project aims to construct a data base, or 'knowledge base', containing a considerable part – approximately 100 million assertions – of the common-sense knowledge possessed by a contemporary human being living in the West.[17])

Turing did not overlook what French regards as the importance, for success in the imitation game, of 'a lifetime of interaction with the world which *necessarily involves* human sense organs, their location on the body, their sensitivity to various stimuli, etc.' (French, 1990: 22). In fact, Turing hypothesized that, to build a machine that is to 'imitate an adult human mind,' we subject a child-machine to 'an appropriate course of education', possibly involving the machine's 'roam[ing] the countryside' equipped with 'the best sense organs that money can buy' (Turing, 1950a: 455, 456, 457, 460; 1948: 420). (As noted above, Turing even anticipated the view that the child-machine consists of an initially unorganized network of neuron-like elements (Turing, 1948: 422–423).) The adult machine resulting from such an education might do rather well in the imitation game (computers with sense organs may be contestants, although they are allowed only verbal contact with the interrogator). French, however, claims that unless a machine 'resembled us *precisely* in *all* physical respects', its experiences of the world would differ from ours in a way 'detectabl[e] by the Turing Test' (French, 1990: 22, 23, our italics). Yet he offers no argument in support of this extreme claim.[18]

[16] For example, a computer may have to disguise its 'superarticulacy' (Michie, 1993: 41–43). One wants an expert system to be able to articulate how it makes its judgments, but notoriously human experts are bad at articulating their expertise. Unless an artificial system dissembles, its superarticulacy may unmask it in a side-by-side comparison with a human expert.

[17] For information on CYC, see (Copeland, 1993), Chapter 5.

[18] French also objects that the Turing Test 'admits of no degrees in … intelligence', in the sense in which '8-year-old humans … have less [intelligence] than adults' (French, 1990: 15). This is false. Consider a computer that is usually misidentified by the interrogator when the human foil is an 8-year-old (of average intelligence), but is usually correctly identified when the foil is an adult (of average intelligence). Here we may say that the degree of intelligence of the machine lies between that of an (average) 8-year-old and an (average) adult.

9.4.6 Intelligent Computer or Lucky Computer?

Shieber (among others) complains that Turing's formulations of his test allow that 'any agent that can be mistaken by virtue of its conversational behavior [for] a human must be intelligent' (Shieber, 1994: 70). If Shieber is correct, we should reject the Turing Test, since a machine may be mistaken for a human by a particular set of interrogators only because those interrogators are gullible, or because the machine by chance has performed uncharacteristically well.

However, Shieber's view is not the only possible interpretation of the Turing Test. In this regard, a much earlier test, proposed by the Cartesian Géraud de Cordemoy in 1668, is apposite:

> To speak is not to repeat the same words, which have struck the ear, but to utter others to their purpose and suitable to them. … [N]one of the bodies that make echoes do think, though I hear them repeat my words. … I should by the same reason judge that parrets do not think neither. … [Concerning other bodies] who resemble me so perfectly *without*. … I think I may … establish for a Principle, that … if I finde by all the experiments I am capable to make, that they use speech as I do, … I have infallible reason to believe that they have a soul as I.[19] (de Cordemoy, 1668: 13–14)

By requiring that, to have a soul (or to think), an object perform satisfactorily in *all* experiments, de Cordemoy avoids misleading 'false positives'. The modern analogue of de Cordemoy's view states that a machine that happens to pass one, or even several, Turing tests, might be shown by subsequent tests to be a relatively poor player of the imitation game. Turing's position as described by Turing is entirely consistent with this interpretation of the test.

It is often objected that Turing failed to provide specifications for a *definitive* test. How long is the test to last? How many judges are involved? However, this objection is wrongheaded. Whether a given machine can faithfully imitate the intellectual behaviour of the brain is not the sort of matter to be settled definitively by a single brief test, or even by a series of such tests. A machine faithfully imitates the brain if it plays the imitation game successfully come what may – no matter what the questions asked or the duration of the game (so long as it does not exceed the human lifespan). Consider two imitation games with no fixed time limits, a man-imitates-woman game and a machine-imitates-human game, each with the same mix of judges that one might meet on, say, the London Underground. If, in the long run, the machine is identified correctly no more often than is the man in the man-imitates-woman game, then the machine faithfully imitates the brain. Any test short enough to be practicable is merely a sample of this ongoing game. After several samples, we may come to believe that, in the long run, the machine will play

[19] The idea that the ability to use language is the hallmark of a thinking being has a long history. Descartes famously declared that it is 'not conceivable that… a machine should produce different arrangements of words so as to give an appropriately meaningful answer to whatever is said in its presence, as the dullest of men can do' (Descartes, 1637: 140).

as well as the man, but only because we take our samples to be representative, and our belief may change on the basis of further rounds of the game.

9.5 Turing's Predictions

In his 1950 paper, Turing predicted:

> [I]n about fifty years' time it will be possible to programme computers … to make them play the imitation game so well that an average interrogator will not have more than 70 per cent chance of making the right identification after five minutes of questioning. (Turing, 1950a: 442)

(Perhaps the qualification 'average' is to exclude fiendish experts – computer scientists, psychologists, or others whose knowledge or training is likely to render them especially skilled at detecting the computer.) This prediction is sometimes misreported (e.g., by Whitby (1996: 61)) as the claim that, by about the year 2000, computers would succeed in *deceiving* the interrogator 70% of the time. But even in its correct form, the outcomes of recent Loebner Turing Test competitions may indicate that Turing's prediction was over-optimistic.

In his 1952 radio broadcast Turing made a different prediction. When Newman remarked:

> I should like to be there when your match between a man and a machine takes place, and perhaps to try my hand at making up some of the questions. But that will be a long time from now, if the machine is to stand any chance with no questions barred?

Turing replied:

> Oh yes, at least 100 years, I should say. (Turing, 1952: 495)

Using the protocol for scoring the test proposed in his 1950 paper, Turing's prediction here is as follows: before 2052 no computer will be able to play the imitation game so well that judges in the machine-imitates-human game will decide correctly no more often than those in the man-imitates-woman game (in each game no questions being barred).

To the question 'Could one make a machine which would answer questions put to it, in such a way that it would not be possible to distinguish its answers from those of a man?', Turing answered 'I believe so' (Turing, 1953: 289). He continued, 'I know of no really convincing argument to support this belief, and certainly of none to disprove it' (ibid.: 289). We must wait and see. As Maurice Wilkes, designer of the EDSAC and head of the University of Cambridge Mathematical Laboratory, wrote in 1953:

> If ever a machine is made to pass [Turing's] test it will be hailed as one of the crowning achievements of technical progress, and rightly so. (Wilkes, 1953: 1231)[20]

[20] Research on which this article draws was supported in part by University of Canterbury Research Grant no. U6271 (Copeland) and Marsden Grant no. UOC905 (Copeland and Proudfoot).

References

Block, N., 1981, Psychologism and behaviorism, *Philosophical Review* **90**(1): 5–43.

Block, N. 1990, The computer model of the mind, in: *An Invitation to Cognitive Science*, Vol. 3, D. Thinking, N. Osherson, and H. Lasnik, eds., MIT Press, Cambridge, MA.

Copeland, B. J., 1993, *Artificial Intelligence: A Philosophical Introduction*, Blackwell, Oxford.

Copeland, B. J., 1994, Artificial intelligence, in: *A Companion to the Philosophy of Mind*, S. Guttenplan, ed., Blackwell, Oxford.

Copeland, B. J., ed., 1999, A lecture and two radio broadcasts by Alan Turing, *Machine Intelligence* **15**, in: K. Furukawa, D. Michie, and S. Muggleton, eds., Oxford University Press.

Copeland, B. J., 2000a, Narrow versus wide mechanism, *Journal of Philosophy* **97**: 5–32; reprinted, 2002, in: *Computationalism: New Directions*, M. Scheutz, ed., MIT Press, Cambridge, MA, pp. 5–32.

Copeland, B. J., 2000b, The Turing Test, *Minds and Machines* **10**: 519–539; reprinted, 2003, in: *The Turing Test: The Elusive Standard of Artificial Intelligence*, J. M. Moor, ed., Kluwer, Dordrecht.

Copeland, B. J., 2002a, The Chinese Room from a logical point of view, in: *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence*, J. Preston and M. Bishop, eds., Oxford University Press, Oxford.

Copeland, B. J., 2002b, The Church-Turing thesis, in: *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, ed.; http://plato.stanford.edu/archives/fall2002/entries/church-turing.

Copeland, B. J., ed., 2004, *The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life, Plus the Secrets of Enigma*, Oxford University Press, Oxford.

Copeland, B. J., ed., 2005, *Alan Turing's Automatic Computing Engine: The Master Codebreaker's Struggle to Build the Modern Computer*, Oxford University Press, Oxford.

Copeland, B. J. and Proudfoot, D., 1996, On Alan Turing's anticipation of connectionism, *Synthese* **108**: 361–377; reprinted, 2000, in: *Artificial Intelligence: Critical Concepts in Cognitive Science*, Vol. 2: Symbolic AI, R. Chrisley, ed., Routledge, London.

Copeland, B. J., and Proudfoot, D., 1999, Alan Turing's forgotten ideas in computer science, *Scientific American* **280**(April): 76–81.

de Cordemoy, G., 1668, *A Philosophicall Discourse Concerning Speech*, John Martin, London; reprinted, 1972, Scholars' Facsimiles & Reprints, New York (page references are to the 1972 reprint).

Descartes, R., 1637, Discourse on the method, reprinted, 1985, in: *The Philosophical Writings of Descartes*, Vol. 1, J. Cottingham, R. Stoothoff, and D. Murdoch, eds., Cambridge University Press, Cambridge (page references are to Cottingham et al.).

French, R., 1990, Subcognition and the limits of the Turing Test, *Mind* **99**: 53–65; reprinted, 1996, in: *Machines and Thought: The Legacy of Alan Turing*, P. Millican, and A. Clark, eds., Oxford University Press, Oxford (page references are to Millican and Clark, 1996).

French, R., 2000, The Turing Test: the first 50 years, *Trends in Cognitive Sciences* **4**: 115–122.

Hodges, A., 1992, *Alan Turing: The Enigma*, Vintage, London.

Leiber, J., 1991, *An Invitation to Cognitive Science*, Blackwell, Oxford.

Michie, D., 1966, Game-playing and game-learning automata, in: *Advances in Programming and Non-numerical Computation*, L. Fox, ed., Pergamon, New York.

Michie, D., 1993, Turing's Test and conscious thought, *Artificial Intelligence* **60**: 1–22; reprinted, 1996, in: *Machines and Thought: The Legacy of Alan Turing*, P. Millican and A. Clark, eds., Oxford University Press, Oxford (page references are to Millican and Clark, 1996).

Minsky, M., 1988, *The Society of Mind*, Picador, London.

Moor, J. H., 1976, An analysis of the Turing Test, *Philosophical Studies* **30**: 249–257.

Moor, J. H., 1987, Turing Test, in: *Encyclopedia of Artificial Intelligence*, Vol. 2, S. C. Shapiro, ed., Wiley, New York.

Moor, J. H., 2003, The status and future of the Turing Test, in: *The Turing Test: The Elusive Standard of Artificial Intelligence*, J. H. Moor, ed., Kluwer, Dordrecht.

Narayanan, A., 1996, The intentional stance and the imitation game, in: *Machines and Thought: The Legacy of Alan Turing*, P. Millican and A. Clark, eds., Oxford University Press, Oxford.

Newell, A., 1980, Physical symbol systems, *Cognitive Science* **4**: 135–183.

Oettinger, A., 1952, Programming a digital computer to learn, *Philosophical Magazine* **43**: 1243–1263.

Prinz, D. G., 1952, Robot chess, *Research* **5**: 261–266.

Proudfoot, D. and Copeland, B. J., 1994, Turing, Wittgenstein, and the science of the mind, *Australasian Journal of Philosophy* **72**: 497–519.

Samuel, A. L., 1959, Some studies in machine learning using the game of checkers, *IBM Journal of Research and Development* **3**: 211–229; reprinted, 1963, in: *Computers and Thought*, E. A. Feigenbaum and J. Feldman, eds., McGraw-Hill, New York (page references are to Feigenbaum and Feldman, 1963).

Searle, J., 1980, Minds, brains, and programs, *Behavioral and Brain Sciences* **3**: 417–424.

Searle, J., 1992, *The Rediscovery of the Mind*, MIT Press, Cambridge, MA.

Shanker, S. G., 1987, Wittgenstein versus Turing on the nature of Church's thesis, *Notre Dame Journal of Formal Logic* **28**: 615–649.

Shannon, C. E. and McCarthy, J., eds., 1956, *Automata Studies*, Princeton University Press, Princeton, NJ.

Shieber, S. M., 1994, Lessons from a restricted Turing Test, *Communications of the ACM* **37**: 70–78.

Stern, N., 1981, *From ENIAC to UNIVAC: An Appraisal of the Eckert-Mauchly Computers*, Digital Press, Bedford, MA.

Sterrett, S. G., 2000, Turing's two tests for intelligence, *Minds and Machines* **10**(4): 541–559; reprinted, 2003, in: *The Turing Test: The Elusive Standard of Artificial Intelligence*, J. M. Moor, ed., Kluwer, Dordrecht.

Strachey, C., 1952, Logical or non-mathematical programmes, *Proceedings of the Association for Computing Machinery*, Toronto, September 1952, pp. 46–49.

Traiger, S., 2000, Making the right identification in the Turing Test, *Minds and Machines* **10**(4): 561–572; reprinted in: *The Turing Test: The Elusive Standard of Artificial Intelligence*, J. H. Moor, ed., Kluwer, Dordrecht.

Turing, A. M., 1936, On computable numbers, with an application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society*, Series 2, 42 (1936–1937), pp. 230–265; reprinted in (Copeland 2004).

Turing, A. M., 1945, Proposed electronic calculator; reprinted in (Copeland, 2005); A digital facsimile of the original typescript is available in the Turing Archive for the History of Computing <http://www.AlanTuring.net/proposed_electronic_calculator>.

Turing, A. M., 1947, Lecture on the automatic computing engine; reprinted in (Copeland, 2004).

Turing, A. M., 1948, Intelligent machinery, National Physical Laboratory Report; reprinted in: (Copeland, 2004); A digital facsimile of the original typescript is available in the Turing Archive for the History of Computing <http://www.AlanTuring.net/intelligent_machinery>.

Turing, A. M., 1950a, Computing Machinery and Intelligence, *Mind* **59**: 433–460; reprinted in (Copeland, 2004).

Turing, A. M., 1950b, *Programmers' Handbook for Manchester Electronic Computer*, Computing Machine Laboratory, University of Manchester; A digital facsimile of the original typescript is available in the Turing Archive for the History of Computing <http://www.AlanTuring.net/programmers_handbook>.

Turing, A. M., 1951a, Can digital computers think?; reprinted in (Copeland, 2004).

Turing, A. M., 1951b, Intelligent machinery, a heretical theory; reprinted in (Copeland, 2004).

Turing, A. M., 1952, Can automatic calculating machines be said to think?; reprinted in (Copeland, 2004).

Turing, A. M., 1953, Chess; reprinted in (Copeland, 2004).

von Neumann, J., 1945, First draft of a report on the EDVAC, Moore School of Electrical Engineering, University of Pennsylvania; reprinted in (Stern, 1981).

Whitby, B., 1996, The Turing Test: AI's biggest blind alley, in: *Machines and Thought: The Legacy of Alan Turing*, P. Millican and A. Clark, eds., Oxford University Press, Oxford.

Wilkes, M. V., 1953, Can machines think? *Proceedings of the Institute of Radio Engineers* **41**: 1230–1234.

Chapter 10
The Turing Test: 55 Years Later

John R. Searle

Abstract In spite of the clarity of the original article, Turing's Test has been subject to different interpretations. I distinguish three of these, corresponding to my earlier distinction between Strong AI and Weak AI. The two strong Turing Tests are subject to refutation by the Chinese Room Argument, the weak Turing Test is not.

The obvious falsity of behaviorism, on which the strong Turing Test was based, leads one to wonder whatever motivated behaviorism in the first place. It is best construed as a consequence of verificationism. The fact that Turing was led into error by the confusions of behaviorism does not diminish his overall achievement or contributions to philosophy and mathematics.

Keywords Turing Test, Strong AI, Weak AI, the Chinese Room Argument, behaviorism, functionalism, brain processes

10.1 Different Ways of Construing the Turing Test

In spite of the fact that Turing's original article (Turing, 1950) is written in very clear and direct prose, there are a number of different ways to interpret the claims made in it. I am not, in this article, going to discuss what I think Turing's actual intentions were, but instead I will focus on two different ways of construing the results of the Turing Test that have been prominent in its application. I will assume for the sake of this article that the test itself is unambiguous. My discussion concerns the question: How do we interpret a positive result? On one natural construal, the test gives us a way of telling whether or not we have successfully simulated some human cognitive capacity, some human form of intelligent behavior that manifests thinking. If the machine can perform in such a way that an expert cannot distinguish the performance of the machine from the performance of a competent human, then the machine has successfully simulated the intelligent behavior of the

University of California, Berkeley

R. Epstein et al. (eds.), *Parsing the Turing Test.*

human. Indeed, if our aim in Artificial Intelligence (AI) is to produce machines that can successfully simulate human intelligence then the Turing Test gives us a criterion for judging our own success and failure. I do not see how one could object to such a test. If the question is whether we have actually simulated, i.e., imitated, human behavior then, so construed, the Turing Test seems trivially right: if you cannot tell the difference between the original and the imitation, then the imitation is a successful imitation.

But there is another way of construing the Turing Test which gives results that seem to me much less satisfactory, indeed false. The Turing Test is sometimes construed as an application of philosophical or logical behaviorism. This may indeed be a reasonable interpretation because behaviorism was a dominant theory in psychology and philosophy at the time that Turing wrote. It is customary to distinguish between *logical (or philosophical) behaviorism*, a thesis in philosophy to the effect that mental phenomena are constituted by behavior and dispositions to behavior, and *methodological behaviorism*, a research program in psychology according to which the proper way to study psychology is to study human behavior. When I say "behaviorism" I mean logical behaviorism. According to the behaviorist conception, human mental phenomena simply consist in, are reducible to, behavior and dispositions to behavior. It is a strict logical consequence of behaviorism that if a machine could emit exactly the same sort of behavior that a human emits with regard to some cognitive phenomenon, then the machine has that cognitive phenomenon in exactly the same sense that the human does. On the second construal we should see the Turing Test as an application of behaviorist principles to AI. So construed, the test has the consequence that machine behavior indistinguishable from human cognitive behavior is not just proof of successful imitation or simulation by the machine, but is conclusive proof of the presence of the appropriate cognition in the machine. How can that be? In its pristine form behaviorism was the view that the manifestation of the external behavior was not just *evidence for* the presence of cognition on the inside of the system, but rather that the behavior and the disposition to manifest the behavior under appropriate circumstances were *constitutive of* the mental. Where the mind is concerned there is nothing going on except the behavior and the disposition to the behavior.

Behaviorism flourished when the linguistic mode of philosophizing was at its strongest, and behaviorism was typically stated as a thesis about psychological attributions. Behaviorists typically argued that statements about mental states are entirely translatable into statements, both categorical and hypothetical, about behavior. The introduction of the notion of hypothetical statements was supposed to explain the notion of a disposition. For example, to say that a person has a certain belief is to say that either she is now behaving in a certain way (categorical), or if such and such conditions obtain she would behave in a certain way (hypothetical, dispositional). On this construal, the Turing Test is a consequence of behaviorism; hence if the Turing Test is false then behaviorism is false. To distinguish these two interpretations of the Turing Test I will use a terminology I introduced years ago to distinguish two corresponding strands in AI. The first Turing Test, according to which passing the test is conclusive proof of successful imitation or simulation, I will call the Weak Turing Test. The second or behaviorist Turing Test, according to which passing the test is conclusive proof of the presence of the psychological phenomena, because the behavior that constitutes passing the test constitutes the psychological phenomena, I will call the Strong Turing Test.

I think behaviorism is more or less obviously false as you can see if you give only a moment's reflection to your own mental processes, such mental processes as thinking about where you are going to spend your summer vacation, or feeling a pain. Having the mental state is one thing; exhibiting it in your behavior is something else. It is possible to have the mental state without exhibiting it in behavior and it is possible to have behavior which does not correspond to a mental state. I find it hard to imagine how anyone could fail to see these obvious points. Nonetheless, several well-known philosophers did deny these points and asserted the reducibility of mental life to behavior and dispositions to behavior.

One difficulty that behaviorists often had was in accounting for the apparent fact that where intelligent behavior is concerned there are typically two things going on, the internal mental processes and the expression of these mental processes in behavior. Right now, for example, I am having internal mental processes of thinking about Alan Turing's famous article, and these thought processes are expressed in my behavior of typing my thoughts on a computer. Pure behaviorists, heroically, denied that there were two things going on in such a case, internal mental processes and external behavior. There is just the intelligent behavior, nothing else. If you look at Turing's article though, it seems that he thinks that the external behavior is a conclusive test for something else, some cognitive process going on inside. But if that is the right way to construe the test, then it is always subject to the objection that the same external behavior might be caused by two quite different internal causal mechanisms. We might build a machine that could duplicate the external behavioral output of a human brain but did so without any thought processes on the inside. (Compare: We can build an electrical engine with the same power output as a gasoline engine, but the performance of the electrical engine is not thereby conclusive proof of internal combustion.)

So now, from our original two, we have three different ways of interpreting a positive result of the Turing Test.

1. The Weak Turing Test. We do not care what is going on inside the system. We just want to duplicate intelligent behavior. If the machine passes the Turing Test, that is conclusive proof that we have succeeded.
2. The Strong Turing Test. It is a mistake to think that intelligence, thought processes, etc. are something in addition to behavior. There is just the behavior and the tendency or disposition to the behavior. That is why the Turing Test is conclusive proof of the mental: there are not two things, mind and behavior. There is just behavior.
3. The Modified Strong Turing Test. There is a difference between the inner intelligent thought processes and the outer intelligent behavior, but the Turing Test can still give us conclusive proof of the presence of the inner thought processes, once we understand their nature.

This last conception I think has tacitly proved to be the most influential and I now want to examine it.

10.2 From Behaviorism to Strong Artificial Intelligence

In one respect behaviorism was improved after Turing wrote his famous article. Many people in the behaviorist (scientific, anti dualist) tradition came to the conclusion that the proper analysis of mental contents showed that they consisted not just in behavior and dispositions to behavior, but rather that there had to be a causal component in the analysis. Mental states were to be defined by the causal relations between input stimulus, inner processing, and external behavior. This view came to be called "functionalism", and functionalism was definitely an improvement on behaviorism in that it recognized that mental in general, and cognition in particular, stand in causal relations to behavior. Indeed functionalists *defined* mental states in terms of these causal relations. However, early functionalists still left the character of the inner mechanism unspecified, and it is subject to an extension of the charge I made against behaviorism. My charge against behaviorism is that you can have the behavior without having the mental states and you can have the mental states without the behavior. Functionalism is subject to the charge that completely different inner mechanisms could produce the same external behavior, in response to the same stimulus; and some of those mechanisms might be mental and others not mental. In the early days of functionalism, its adherents claimed that it did not matter what the inner mechanism was. Any inner mechanism at all that produced the right output in response to the right input, and thus satisfied the Turing Test, literally had cognition. Because this version of functionalism ignored the character of the mechanism, and just treated the brain as a black box, it was sometimes called "black box functionalism". However, it is intellectually unsatisfying not to have the nature of the mechanism specified; and the rise of computer science, together with the rise of AI within computer science, led naturally to the hypothesis that the essential mechanism that creates behavior that satisfies the Turing Test is computational. This view is sometimes called "computer functionalism" and I have also baptized it "Strong Artificial Intelligence". It seemed like an improvement on both behaviorism and earlier versions of functionalism.

The rise of computer functionalism in philosophy was paralleled by the growth of AI as a project in computer science. It seemed natural to a number of people in cognitive science (though not to me) to think that the marriage of computer functionalism with the Turing Test might at last give us the key to human cognition. I think this project was aided and not impeded by a certain ambiguity in the notion of "artificial intelligence", and I want to digress briefly to sort out the ambiguities.

The expression "artificial intelligence" is multiply ambiguous. I think this ambiguity may have been useful in the early days to get funding and to provide a certain openness for the research project. However, it can be a source of intellectual confusion and I now want to disambiguate it. The word "artificial" is systematically ambiguous;

because "an artificial X" can mean either a real X, but produced by artifice; or it can mean something that is not really an X, but is only an imitation of an X. Thus, for example, artificial dyes are really dyes, but they are produced in factories, unlike vegetable dyes, which are derived from plants of various kinds. But artificial cream is not really cream at all. It is an imitation of cream. So already we have a double ambiguity: "artificial intelligence" could be either real intelligence produced artificially, or it could be something that is not really intelligence at all but just an imitation of intelligence. The ambiguity is compounded when we shift over to the word "intelligence". There is a literal use of the word "intelligence" where intelligence does not imply the presence of any thought processes, or any other relevant psychological states, whatever. I can literally say that one book is more intelligent than another book, without implying that either book has consciousness or any other form of cognition. But if I say literally that Sally is more intelligent than Sam, then I am making an attribution of intelligence which implies actual properties of human cognition, it implies actual thought processes and other sorts of exercises of cognitive abilities. Just to have labels for these two different sorts of intelligence, let us use the expression "mental intelligence" in such a way that the presence of mental intelligence implies the presence of actual mental or cognitive processes, and "non-mental intelligence" implies only intelligent processes of some nonmental kind. We attribute mental intelligence to people and some animals; we attribute nonmental intelligence to books and pocket calculators. We need this distinction in any case because it is clear that human beings have both kinds of intelligence. Thus, for example, the stomach is usually said to be a very intelligent organ, but it has no cognitive processes. The ambiguity then, of each expression allows for at least four different interpretations of the expression "artificial intelligence". First, it can mean real mental intelligence, produced artificially (this is the view that I call Strong AI, and it is identical with computer functionalism); second, it can also mean something which is not real mental intelligence but only an imitation. Third, it can mean real nonmental intelligence; and fourth, it can mean an imitation of nonmental intelligence. It is at least arguable that the third and the fourth are identical, because where nonmental phenomena are concerned, one might argue, if you can simulate nonmental intelligence, you produce nonmental intelligence. The strongest of these four interpretations of "artificial intelligence" is the claim that in AI we are artificially producing real mental intelligence, in the sense of "intelligence" that implies real cognitive processes.

In the early days of cognitive science I did not find many people in AI who were eager to make these distinctions in a precise fashion. Perhaps it was useful to leave the research project open. But if we are going to examine the implications of computer simulations of human cognition, it becomes absolutely crucial to make clear what exactly the aim of the research project is. Are we trying to produce the real mental thing artificially, or are we not? In any case in the practice of cognitive science the Strong Turing Test was treated as the Modified Strong Turing Test. The search was not for just any system at all that could pass the Turing Test, but rather the aim was to get *computers* to pass the Turing Test, and the assumption was that such computers would duplicate and not merely simulate human cognition. The tacit assumption was that the human mind is a digital computer program or a set of computer programs. It was typical in the cognitive science text books of those days to have an equation that went: “mind is to brain as program is to hardware”. If you got the right program, one that can pass the Turing Test, you would have duplicated and not merely simulated human cognition. This is why, for example, even after you got a machine that could perform successfully, there was still a question as to whether or not the program the machine used was the same as the one that humans were using. Psychologists were often asked to perform reaction time experiments to see if we could get evidence for computational equivalence. The idea was that if different reaction times for humans matched differences in the time of the computer operations, then these matching differences were at least evidence that the humans were using the same sort of program as the computer. There was, in short, a marriage between the Strong Turing Test and Strong AI to produce the Modified Strong Turing Test.

10.3 The Refutation of Strong AI and Its Philosophical Implications

Years ago I refuted Strong AI and with it both versions of the Strong Turing Test.[1] According to Strong AI, the appropriately programmed digital computer program, programmed so as to be able to pass the Turing Test, literally has the same cognitive capacities as the human being. I refuted this with a now well-known thought experiment called the Chinese Room Argument, wherein a person who does not know Chinese, myself, for example, is locked in a room, and as Turing says, is equipped only with such things as a pencil and a piece of paper, together with other elements that go to make up the data base and the program, in this case Chinese symbols, and a program of instructions as to what to do with the Chinese symbols. In the thought experiment the person in the room, namely me, is able to simulate the behavior of a Chinese person in a way that satisfies the Turing Test, because, for example, he gives the correct answers to Chinese questions, but all the same, he does not understand a word of Chinese. Why not? Because he has no way to get from the manipulation of the symbols to the meanings of the symbols, and if the person in the room does not understand the meanings of the symbols on the basis of implementing the program, then neither does any other computer solely on that basis, because no computer, just in virtue of its computational properties, has anything that the man does not have. Furthermore, the whole room has no way of getting from the symbols to their meanings. The whole system has no way to attach a semantics to the syntax of the computer symbols.

However, it is philosophically unsatisfying just to refute a thesis. There has to be an underlying philosophical explanation of why the thesis went wrong. This to

[1] Searle, J. R., 1980, Minds, Brains, and Programs, *Behavioral and Brains Sciences* **3:** 417–424.

me is the more interesting philosophical question and I will say a little more about it. Intelligent human behavior is typically *caused by* inner mental processes and capacities. This causal character of the relation between mind and behavior was ignored by logical behaviorism. Thus, for example, when a native speaker of Chinese can intelligently answer questions in Chinese it is because her external behavior is the external expression of her inner mental capacities. We saw earlier that this weakness in logical behaviorism was corrected by functionalism. Functionalism, indeed, defined mental states as anything at all that stood in the right causal relations between external stimulus, other mental states, and external behavior. So a belief is anything that is caused by perceptual inputs and together with a desire, causes external behavior. Desires analogously could be defined in terms of input stimulus, other mental states, and external behavior. But this thesis, as I have suggested, left open a terrible weakness: it did not specify the specific features of the mental states that enabled them to function causally. This limitation was corrected by computer functionalism to the extent that it at least specified a mechanism: the computer program that mediated the causal relations between the external input stimuli and the external output behavior. But the difficulty with that theory is that the program is defined purely formally or syntactically, and consequently does not, qua program, carry the intrinsic mental or semantic contents that human mental states actually have.

We can now see why the Strong Turing Test gives us a result that is obviously false. If we interpret it in its pristine form as just logical behaviorism we already know that it is false because the presence of the appropriate behavior, though evidence for, is in no sense constitutive of the underlying mental states and processes that in the case of humans typically cause that external behavior. If we augment the logical behaviorist form of the Turing Test with the computer functionalist, or Strong AI form we still get a result that is obviously false because the external behavior caused by the implementation of an inner computer program, defined entirely formally or syntactically (e.g., in terms of Turing's account of a Turing Machine as performing such operations as printing 0's, erasing 1's, moving one square to the left, moving one square to the right, etc.) is still not sufficient to constitute actual human mental processes, which actually have mental or semantic contents.

Indeed, typically we have such machines for practical purposes because they can give us the same external results without having to go through the kind of internal mental effort that human beings typically require. When I use my adding machine, I use a machine that passes the Turing Test for addition, subtraction, etc., indeed it surpasses even the very best mathematicians in intelligent behavior, but it does not thereby have thought processes about mathematics. How do I know that? How do I know that it is not thinking about long division when it does long division? Because I know how it was designed and built. It contains an electronic circuit designed to carry out the algorithms for addition, subtraction, multiplication, and division. It does not think about mathematics, because it does not think about anything. And what goes for my pocket calculator goes for more complex forms of commercial computing machinery. They were neither designed to be conscious nor to have thought processes, rather in the case of Turing machines of the von Neumann architecture, we have designed machines to perform complex operations using only two types of symbols, usually thought of as 0's and 1's.

But, one might be inclined to ask: why can't the 0's and 1's be sufficient for human mental thought processes? After all, in the very brain itself there are only neurons and they either fire or do not fire, so what is the difference between the binary system of the brain and the binary system of the digital computer? I think that is an intelligent and appropriate question. It has a simple and decisive answer: the neuron firings are part of a causal mechanism that *causes* consciousness and cognition as a higher level features of the brain system. The brain is a machine and as such its essential processes are matters of energy transfer. The 0's and 1's of the implemented computer programs are purely abstract syntactical entities that do not, as such, cause anything. Rather, the program is implemented in the hardware, and the hardware processes that are essential for the implementation are those, and only those, that carry out the formal, syntactical steps of the program. The hardware might cause consciousness for some other reason (e.g., when I carry out the algorithm for long division, my brain processes also cause me to be conscious), but the program, *qua implemented program*, knows nothing of causing consciousness or anything else except the next state of the program when the machine is running. To put this point slightly more technically: the notion *same implemented program* defines an equivalence class that is specified independently of the physics of the hardware mechanisms in which it is realized. A specific form of hardware, my brain, for example, might also cause consciousness while I am carrying out the algorithm for doing long division, but the algorithm itself does not have any causal powers independent of the implementing medium.

It is important to make this point completely clear. The question, "Can a computer think?" is ambiguous. It can mean either "Can something be a computer and also think?" or it can mean "Is computation by itself constitutive of or sufficient for thinking?" If we define computation in Turing's terms, as the manipulation of symbols (0's and 1's or Chinese symbols, or whatever – it does not matter), then the answer to the first question is obviously yes. Something, me for example, can both think and manipulate symbols. But the answer to the second question is obviously no. Computation so defined – by itself, *qua* computation – is not constitutive of nor sufficient for thinking because it is defined entirely syntactically, and thinking has to have something more than just symbols, it has to have a meaning or semantic content attaching to the symbols. And that is what the Chinese Room Argument proved: the syntax of the implemented computer program, by itself is insufficient for the understanding of the semantics of actual Chinese words and sentences.

Though the sort of computer you buy in a store is also a machine, its computational processes are not defined by energy transfers, rather, they are defined by abstract mathematical processes that we have found ways to implement in the hardware. The problem with the commercial digital computer is not that it is too much of a machine to produce consciousness; rather it is not enough of a machine because, unlike the brain, its essential operations, its computational operations, are defined in terms of abstract algorithmic processes and not in terms of energy transfers.

In the past, as here, I have found it useful to state this point using the distinction between syntax and semantics. Brain operations cause consciousness which has semantic content. Program operations are purely syntactical, and the syntax by itself does not constitute consciousness, nor is it sufficient to cause consciousness.

Consciousness is a state that the brain is in when it is caused to be in that state by the operations of lower level neuronal mechanisms. In order to create such a state artificially you would have to duplicate, and not merely simulate, the actual causal powers of human and animal brains. There is no reason in principle to suppose that we would have to have organic materials to do this, but whatever materials we use we have to duplicate the causal powers of actual brains. We ought to hear the question, "Can you create consciousness with a mechanism other than brains?" in exactly the same way we hear the question, "Can you create the pumping of blood in a mechanism other than human and animal hearts?" In both cases the question is about causal mechanisms and not about syntactical processes. The computer simulation of brain operations stands to the actual brain operations as a computer simulation of a heart stands to the actual pumping of blood.

We can summarize these points in two propositions:

1. If the Turing Test is interpreted as either the Strong Turing Test, or in its modified form, as the Modified Strong Turing Test, as giving us conclusive proof of the presence of inner mental contents, it fails. It fails because a system, computational or otherwise, can behave as if it were intelligent without having any inner mental processes whatever.
2. The prospect of creating human thought processes solely by implementing Turing machine programs also fails because the program is defined syntactically. The implemented computer program consists in a set of processes that are specifiable syntactically, in terms of the manipulation of symbols, in a way that is independent of the physics of the implementing medium. Any physics will do provided only that it is rich enough and stable enough to carry out the steps in the program. And the syntax by itself does not constitute consciousness nor is it by itself sufficient to cause consciousness.

10.4 Why Was Anyone Ever a Behaviorist?

I have already made many of these points in a number of other writings. It is worth repeating them briefly because they are sometimes overlooked in these discussions. I now want to turn to a question I have not previously written about: because the mistaken character of behaviorism is so obvious, and because the Strong Turing Test is itself an expression of behaviorism, how did it come about that behaviorism persisted for so long?

If one looks at the intellectual history of any era, there are likely to be stunning and widespread mistakes that should have been easily avoided, even given the limitations of the knowledge base of the times. An amazing and pervasive mistake throughout the 19th century was idealism, the theory that all of reality was mental or spiritual, and that an independent material reality does not exist. It is hard for us today to recover the mode of sensibility that made this view seem not only possible, but indeed compelling, to thinkers of the stature of Berkeley, Royce, Bradley, and Hegel, just to mention a few. In the 20th century the mirror image of the 19th-century mistake of idealism was the mistake of behaviorism, which still exists in some quarters even today. Just as idealism denied the reality of a mind independent physical world, so behaviorism denies the reality of subjective inner mental states, in favor of an account of the mind according to which mental states consist in external behavior. The black box functionalist and computer functionalist successors to behaviorism inherit the same mistake of denying the irreducible reality of inner, subjective, qualitative mental states. How did we get into this mess?

It is fairly easy to trace the history of post-Cartesian idealism. In its modern form it begins with Berkeley and is driven by the epistemic obsession that he had inherited from Descartes. The way to overcome the skepticism that seemed to make knowledge of the real world impossible, was to deny the gulf between the evidence and the reality that the evidence is supposed to be evidence for. If the material world is supposed to exist independently of experience, and if knowledge of the material world is based on sense experience, then it seems we can never have knowledge of the material world, because we can never get outside the circle of our own experiences to the mind independent reality itself. We overcome this skepticism by denying the distinction between our experiences and reality. The object just is the permanent possibility of sensation (Mill, 1865) and the world just is a collection of minds and ideas (Berkeley, 1998). The urge to behaviorism bears an uncanny resemblance to this urge to idealism. If consciousness and intentionality are supposed to exist independently of behavior, and if our only evidence for their existence in other people is the behavior of the other people, then it looks like we are forced to skepticism, because we can never get outside the sequence of the observable behavior of other people to observe the inner mental phenomena that are supposed to lie behind the behavior. But if the mind just is the behavior, and (in the case of functionalism) the observable mechanisms that cause the behavior, then we overcome skepticism about the mind in the same way that idealism overcame skepticism about the external world. In both cases we defeat skepticism by denying that there is a difference between the evidence and the reality that the evidence is evidence for.

Behaviorism is thus best construed as a form of verificationism. Verificationism in turn, is a response to the skeptical tradition that originated with Descartes and received its finest expression in the works of the British empiricists and their 20th-century followers, the logical positivists. It is no accident that behaviorism was the dominant philosophy of mind of the logical positivists.

10.5 Giving Up the Strong Turing Test

If, as I have urged, we should reject the Strong Turing Test and the Modified Strong Turing Test as tests for the presence of mental states, then a natural question is what alternative test do we propose, what alternative to these Strong Turing Tests is there? I think that the correct response to this is to say that we would be mistaken to suppose that there had to be some single mechanical test for ascertaining the presence of mental states and cognitive capacities in others. There are a variety of ways that we actually find out about whether other systems have mental states, and I want to conclude this brief discussion by mentioning some of these. Mental phenomena have a first-person ontology in the sense that they only actually exist when they are experienced by a human or animal agent. That means that the best way to know whether or not a system is having a certain mental process is to be that system. Of course, one often makes mistakes about one's own inner mental states, but all the same, when it comes, for example, to feeling my pains or thinking my thoughts, there is no substitute for being me. However, since there is only one system in the universe that I can be identical with, namely me, I have a problem of how I can know about the thoughts and feelings, capacities and limitations, of other humans and animals. The temptation here is to revert to a type of behaviorism and say, well, the way I know that humans and animals have mental states is by their behavior. I think this answer is incorrect. In the case of my dog, for example, I am completely confident that he has mental states even though he has no language. Now, why am I so confident about that? It is not simply because of his behavior but rather because of the combination of his behavior and the underlying causal mechanisms that I can see are relevantly similar to my own. I do not have to have an elaborate theory of dog physiology to know that that is his nose, these are his ears, that is his mouth, this is his skin, etc. And the way that I am able to infer the presence of mental states on the basis of behavior is that I observe not just a correlation between stimulus and response, but rather that I can see that there are similar causal mechanisms that mediate the relationship between the stimulus and the response. The principle on the basis of which I know that my dog has mental states is not: same behavior, therefore same mental states, but rather: similar underlying causal structure, therefore similar cause and effect relationships. This is why, incidentally, though I am confident about dogs and chimpanzees, I am not at all confident about termites and fleas. Well, how would one find out in these cases? Again, I think that the principle is not hard to state, though we do not know enough neurophysiology to apply it in practice. If we understood the actual causal mechanisms that produced consciousness and intentionality in human beings and higher animals, then we would have very good grounds for supposing that these same phenomena occur in other systems. So, for example, let us suppose that there is a certain neurobiological process, whose technical name we will abbreviate as "ABC", that in human beings and higher animals causes conscious mental phenomena. Now let us suppose that we observe ABC in mice and pigeons, but we do not observe anything like it in fleas and grasshoppers. Let us suppose, furthermore, that we have a causal explanation of the behavior of the fleas and grasshoppers that is quite different from the explanation of the behavior of higher animals, that shows that their causal mechanisms are much more like simple tropisms than they are like complex cognitive phenomena produced by human and animal neurobiology. Then it seems to me we would regard this as conclusive proof that chimpanzees, dogs, mice, and pigeons have conscious mental phenomena in the same way that we do, but that fleas and grasshoppers do not.

I hope that my remarks here will not be misunderstood. I regard Alan Turing as one of the great minds of the 20th century. It in no way diminishes his achievement to say that the test that he proposed was inadequate if it is understood as an expression of the false philosophical theory of behaviorism. Behaviorism is a theory for which he was not responsible, and it is now best regarded as obsolete.

References

Berkeley, G., 1998, *A Treatise Concerning the Principles of Human Knowledge*, Jonathon Dancy, ed., Oxford University Press, Oxford.

Mill, J. S., 1865, *An Examination of Sir William Hamilton's Philosophy*, London.

Searle, J. R., 1980, Minds, brains, and programs, *Behavioral and Brains Sciences* **3**: 417–424.

Turing, A., 1950, Computing machinery and intelligence, *Mind* **59**(236): 433–460.

Chapter 11
Doing Justice to the Imitation Game

A Farewell to Formalism

Jean Lassègue

Abstract My claim in this article is that the 1950 paper in which Turing describes the world-famous set-up of the Imitation Game is much richer and intriguing than the formalist ersatz coined in the early 1970s under the name "Turing Test". Therefore, doing justice to the Imitation Game implies showing first, that the formalist interpretation misses some crucial points in Turing's line of thought and second, that the 1950 paper should not be understood as the *Magna Charta* of strong Artificial Intelligence (AI) but as a *work in progress* focused on the notion of Form. This has unexpected consequences about the status of Mind, and from a more general point of view, about the way we interpret the notions of Science and Language.

Keywords Determinism, formalism, gender difference, geometry, mental processes

The common use of the phrase "Turing Test" instead of the expression "Imitation game", preferably used by Turing,[1] is concomitant with a shift of meaning in the expression "artificial intelligence" (AI). During and after World War II, the latter expression would rather mean the mechanical gathering of enemy information. But the meaning began to evolve at the end of the 1950s when AI was used to raise the standard of revolt against "Cybernetics" (Dupuy, 2000). Contrary to Cybernetics, which had always considered the physical and the logical level of cognition at the same time (McCulloch and Pitts, 1943), AI was based upon a radical cut between a purely formal, computable level of intelligibility and its physical counterpart which could be passed over and left to engineering. Later, in the mid-1970s, the notion of a "Turing Test" became popular when mainstream cognitive science was

---

Laboratoire CREA-CNRS, Ecole Polytechnique

---

[1]In Turing (1950), Turing uses only three times the word "test", two pages apart (446–447), while referring to the imitation game and answering the objection called "Argument from Consciousness".

R. Epstein et al. (eds.), *Parsing the Turing Test.*

claimed to be entirely based upon the brain–computer paradigm and its distinction between hardware and software derived from AI – hence a global shift towards formalism and functionalism.

Interpreted from a formalist point of view, the Imitation Game set up by Turing in his 1950 world-famous article amounts, therefore, to a test that would show, on a statistical basis, that the discrimination between the verbal expressions of a human being and those of a computer lies beyond human decision. The major claim of my paper is that this test is not feasible in the conditions just mentioned, contrary to what an abundant literature,[2] blinded by a formalist interpretation, has been eager to show. This is why I consider the reduction of the imitation game to a test as an impoverishment of Turing's original point of view. To my mind, the phrase "Turing Test" is somewhat of a misnomer and I shall use three anti-formalist arguments against its use.

Let me begin first by describing the origins of the formalist point of view I want to re-examine.

11.1 Clockwise Determinism: The Formalist Interpretation of the Imitation Game

Let us start with the historical rise of the formalist point of view in scientific knowledge.

11.1.1 Determinism in Physics and in Mathematics: The Laplacian World View

Formalism is not the ultimate paradigm of scientific rigour it is sometimes claimed to be. It has to do with a specific state of scientific knowledge which was current between the 17th and the 20th century and in which the clarification of what could be rigorously *predicted* was linked to the idea of *computation*.[3] To make this clear, I shall amply rest upon the epistemological framework developed by two professional logicians (Longo, 2005; Girard, 2001) in which a parallel is drawn between

[2] To name just a few: (Michie, 1974; Hofstadter and Dennett 1981; Pylyshyn, 1984; Haugeland, 1985; Penrose, 1989; Boden, 1990; French, 1990; Leiber, 1991). The "Turing Test" is even now an entry in the Collins dictionary: "Turing test – A proposed test of a computer's ability to think, requiring that the covert substitution of the computer for one of the participants in a teletype dialogue should be undetectable by the remaining human participant."

[3] Why Laplace since Newton is the great figure for that matter? Because it is to Laplace that Turing refers to in 1950.

the evolution of determinism in physics and that of computation in mathematics.[4] Let us begin by quoting G. Longo on Laplace and Hilbert (Longo, 2001):

> Laplace proposed a paradigm for the mathematical analysis of Physics, the so-called 'Laplacian determinism'. In this perspective, the systems of (differential) equations could 'completely' describe the physical world. More precisely, if one wanted to know the state of the physical world in a future moment, with a given approximation, then it could suffice to know the current state of affairs up to an approximation of a *comparable order of magnitude*. By formally computing a solution of the intended equations, or by suitable approximations by Fourier series (as it will be said later), one could deduce (or predict or decide) the future state, up to the expected level of approximation. …About one century later, D. Hilbert resumed Laplace's program in a different context. He first set the basis for the rigourous notion of 'formal system', as well as for the distinction between 'theory' and 'metatheory'. He then conjectured that the key system for Number Theory, Peano's Arithmetic (where he had interpreted Geometry, 1899), was complete with regard to the intended structure of numbers (or that any well formed proposition about the 'world of numbers' could be decided by formal or 'potentially mechanisable' tools).

The common picture that emerges from these two scientific views is that it is possible, from a system of equations or a system of axioms, to *decide* whether a physical or a mathematical event would occur or not: the *temporal* evolution of a dynamical system or the *logical* consequences of a formal system receive *the same deterministic* treatment. At the centre of this scientific picture, we find the notion of *function* essential to the constitution of mathematical physics. The notion of a function was still akin in the 17th and 18th century to the idea of a step-by-step computation procedure.[5] Completeness and decision in systems of equations or of axioms seemed, from this point of view, to go hand in hand: *all* the events and *every* one of them are potentially computable. In the natural philosophy of that time, deterministic computation was heralded the most general model applicable to both Mind and Matter. It had a technological counterpart in the notion of a *clock*: the entire universe was nothing but a gigantic clock and the mechanics of Mind were just a very complex part in it (Longo, 1999).

Even if the notion of a function had tremendously evolved in between, Turing inherited the physical and mathematical world view developed during this classical period and made a great contribution to its mathematical aspect by determining what was meant by "potentially mechanizable". But he also participated in transforming the key concept of determinism when he managed to show that it had an *inner limit* in the axiomatic domain. Turing was not the only one to contribute to this transformation. Gödel is of course the great figure in this case. It must be stressed that they were, so to speak, in a position where they were able to grasp both sides of the evolution of the concept of determinism – absolute determinism on the one side and relative determinism on the other. Secondly, the transformation of this

[4] Other works are of course of great value. See, for example, Hacking (1975) and Stewart, (2000).

[5] It was only after the set-theoretical turn that it became clear that it was necessary to determine whether a given function was computable or not.

concept happened in mathematics as well as in physics. In physics too, Poincaré and the Andronov's school in Gorki had guessed and then established that absolute determinism was not the last chapter in physics. These two points, closely related to one another,[6] are to be remembered when the Imitation Game is approached.

11.1.2 Formalism in Mathematics and Logic: A Trick of the 1920s

What is strictly defined as "Formalism" in mathematics is only a technical strategy used by some mathematicians in the first decades of the 20th century to bypass some very deep problems encountered in the interpretation of geometry. This point has been described at length by historians of mathematical logic (Feferman, 1987; Gandy, 1988; Guillaume, 1994), and I shall only summarize the different stages of the story.

In the beginning, the very deep crisis was triggered by non-Euclidean geometries at the end of the 19th century: the notion of space became so confused that it lost its precise meaning and the entire mathematical world was immersed in a kind of geometrical "delirium", as Frege put it (1884). Geometry, which had been the very foundation of mathematical certainty since its Euclidean axiomatization, had lost its cardinal virtue: the certainty of Euclidean geometry was not the touchstone of objective truth any longer since contradictory propositions were possible between the different axiomatics of geometry. Even geometry could be inconsistent – but if geometrical certainty was shattered for ever, where was mathematical consistency to be found again? *Formalism* was a way out of this crisis: it was both a linguistic and an arithmetical turn in which formal coding based on natural number arithmetic became the keyword. Peano and Padoa began the great linguistic reform in 1889 by reinforcing mathematical rigour concerning the basic notions (definitions, axioms, etc.) used in the axiomatic theory of number (Peano, 1889). Interpreting the axioms of geometry in a completely revolutionary way (Hilbert, 1899), Hilbert was able to reduce them to arithmetic by 1917, delimiting the problem of consistency to that of arithmetic. To secure mathematics as a whole and more specifically arithmetic, he opposed meaningful propositions in contentual axiomatics to meaningless strings of symbols in formal axiomatics (called "meta-mathematics") as content to form. It became possible to manipulate strings of symbols without taking care of their supposed meaning, *even in arithmetic*. A mathematical core was therefore secured which was limited to finitary strings of meaningless symbols: even propositions in

[6] G. Longo draws an analogy between Laplacian and Hilbertian determinism (respectively) in physics and in axiomatics on the one hand, and Poincaréian and Gödelian-Turingian inner limitations of determinism in physics and in axiomatics on the other.

which the meaningful content used the notion of second-order infinity (i.e., richer than the indefinite enumeration of natural numbers) and which were very common in geometry, were nonetheless considered, from a formal point of view, as finite strings of symbols and could therefore be handled, at least theoretically, in the same way as propositions endowed with a finitary content. A purely formal approach to axiomatics was launched: formal axiomatics became a kind of machinery marshalling theorems generated by explicit finitary rules and would then replace geometry as the universal dispenser of truth. In 1928, Hilbert was able to formulate, in a precise way, some pending questions concerning determinism in formal systems[7]: it was the acme of what was to be called "Hilbert's program". What is really impressive with Hilbert's program is that it is so precisely stated that it can be *proved* right or wrong – depending on the open questions under consideration. In a way, Turing's imitation game inherited this "informal rigour" but this does not imply that we should consider it only as a by-product of formalism.

Hilbert's new approach to mathematics had far-reaching consequences concerning the relationship between mathematics and physics on the one hand and mathematics and logic on the other. I shall mention only two of them.

11.1.2.1 A Consequence in the Philosophy of Physics

A theoretical split began to appear between foundational research in mathematics and in physics. Since Cartesian times, the intuition of the naturality of the three-dimensional space had been baffled by new advances in algebra. Although it was still controversial at the time then and now (Brouwer, 1913), the Cantorian set-theoretical point of view made possible the reduction of any geometrical figure to sets of points: any dimension could then be mapped onto the straight line.[8] But the concept of a set of points had a very serious drawback: any information concerning the *structure* of space, that is, the *geometrical forms*, was entirely lost in the mapping process. This was pointed out by physics-oriented mathematicians like Poincaré (1913) and Weyl (1918) who, remembering Riemann's revolutionary way of interpreting space itself as a geometrical object (Riemann, 1854), were not keen on encapsulating mathematics in a pure formal setting completely severed from physics. And even if Hilbert and the Hilbert school made some crucial contributions to physics, the epistemological atmosphere of the 1920s developed a kind of inner schizophrenia: whereas geometry was at the forefront of research in theoretical physics, it was banned altogether in foundational mathematics. The solution to this

[7] That is, their completeness (can every formula be either proved or refuted from the axioms?), their consistency (can no contradictory formula be generated from the axioms?), and their decidability (is there an effective way to always decide whether a formula is deducible from the axioms or not?).

[8] See the Cantor/Dedekind correspondence in Cantor (1932).

theoretical split was reductionism[9]: the whole structure of scientific knowledge was supposed, in the end, to rest upon the finitary base delimited by formal axiomatics.[10]

11.1.2.2 A Consequence in the Philosophy of Logic

As for logic, the main consequence of the rejection of geometry out of foundational research was that the manipulation of meaningless strings of symbols in formal axiomatics had to rest upon an entirely new foundation. Since Space could not be trusted to the same degree any more, the notion of Mind found a new favour: as a philosophical companion to the algebraization of mathematics since Cartesian times (Lachterman, 1989), the finitary step-by-step construction, once reduced to the step-by-step construction of meaningless strings of written symbols, was then heralded the basic *modus operandi* of Mind itself. Hilbert stated (1923):

> Our thinking is finitist; when we are thinking, a finitist process is going on.

It was a philosophical axiom, so to speak, which was made necessary by the direction axiomatic research had taken. Since the actual reduction of the entire scientific knowledge to finitary construction was not entirely achieved by the formalist program yet (Hilbert, 1925), one could say that Hilbert was, at the time, making high bids on the future foundation of Science – on the one hand, physics was deeply concerned with geometry in order to determine the ultimate properties of Matter, while on the other, mathematics, at least in its foundational enquiry, was only concerned with formal inference, that is, the laws of Thought. The chasm between Matter and Mind that Hilbert contributed to establish accelerated the parting of the ways between the foundational paradigms accounting for the development of mathematics on the one hand and of physics on the other.

11.1.3 Turing's Contributions to the Formalist Viewpoint: Physical Determinism of the Formal Mind

Turing took foundational issues concerning the inner working of the Mind where Hilbert had left them 15 years before him, and he made two important contributions to the formalist point of view. First, he transformed the only philosophical Hilbertian

[9] Compared to Frege who was, as it were, a fundamentalist concerning the matter of reductionism, Hilbert himself was more of an agnostic: for him, the more interpretations a formal knowledge was able to receive, the better, because formal knowledge was "reference-free". On the contrary, Frege was claiming the absolute unicity of reference (Hallett, 1994).

[10] The continuum problem wrongly advertised as solved by Hilbert (1925), in a proof he never completed, was the acid test for the reduction of higher-order infinity to finitary combinatorial propositions.

axiom concerning the finitist process of thinking into a purely technical matter by inventing a formal counterpart to the notion of a mental process. Second, he managed to implement a purely logical concept of computation in the physical world. These two points are important because the argument devised in the Imitation Game is based upon them. Let me describe them briefly.

11.1.3.1 Formal Modelling of Mental Processes

Since the Gödelian arithmetization of formal axiomatics (i.e., the one-to-one correspondence between formalized elements and natural numbers) (Gödel, 1931), it had become clear that understanding the notion of a finitary deduction in a formal context would imply a very thorough analysis of what computation actually was. Of course, Hilbert had postulated the finiteness of the operating Mind: proving a theorem in formal axiomatics amounted for him to computing an algorithm. But the algorithmic finitary process was justified only by an external and, indeed, psychological commentary about what was meant by "carrying out a finite number of steps" to perform a formal deduction. What was to be formalized, that is, what was the inner working of meaningless strings of symbols when something was computed was still unclear. Turing carried the formal point of view a step further when he managed to define with suitable precision exactly what was meant by "carrying out a finite number of steps". The mental process underlying any computation was decomposed into an explicitly finite number of steps which could be completely analysed in a tabular form without any appeal to psychology (Turing, 1936). To make this point of view clear, Turing was inviting the reader of his 1936 article to place himself in a finitary mode of thinking and to become a "computer"[11]:

> The behaviour of the computer at any moment is determined by the symbols which he is observing and his 'state of mind' at that moment. We may suppose that there is a bound B to the number of symbols or squares which the computer can observe at one moment. If he wishes to observe more, he must use successive observations. We will also suppose that the number of states of mind which need to be taken into account is finite. The reasons for this are of the same character as those which restrict the number of symbols. If we admitted an infinity of states of mind, some of them will be 'arbitrarily close' and will be confused. Again, the restriction is not one which seriously affects computation, since the use of more complicated states of mind can be avoided by writing more symbols on the tape.

Describing from the outside, the finitist mental act would not do any more; everyone had to carry out the experience for himself by entering a finitary and constructive state of mind. Therefore, the reader, by identifying himself with the process performed within the signs themselves, was to be convinced of the finitary nature of the source of all computation. Through this experience, Turing was reversing Hilbert's philosophical axiom: it was *the written symbols that generated*

[11] The "computer" in this quotation is of course a human being in the course of a computation (cf. Turing, 1936, § 9).

states of mind and not the other way round. Therefore, the mental act was secondary in comparison with what could be linguistically described from a finitist point of view – what was at stake was only the *mapping* of a discrete set of symbols with a set of behaviours in a computing machinery and not the "reality" of some states of mind that were only postulated. The formal representation of this "mental act" could be carried out *no matter who or what* was actually performing it: the finitary process itself, since it was only a finite list of "behaviours" any computer could perform, was entirely mirrored in a formal treatment of written symbols, that is, a *program*. This notion of a formal counterpart was far from the Hilbertian mentalism and was modelled according to the Gödelian arithmetical method. It was called, by Turing, a "machine" and soon after Church's review of Turing's 1936 article (Church, 1937), a "Turing machine". One can then say that the 5-year evolution (1931–1936), which goes from the Gödelian arithmetization of formal axiomatics to the Turingian building of "tables", could be aptly labelled the birth period of programming languages. This non-mentalist and hyper-formalist attitude, so to speak, should be remembered also when we shall get to the Imitation Game.

11.1.3.2 Physically Implemented Automata

The second contribution of Turing to the formalist viewpoint was the physical implementation of the logical machine he managed to plan as early as 1944. It could be argued that physical implementation has nothing to do with formalism *sensu stricto*but, in fact, as Longo and Girard have shown, the idea of *deterministic predictability* which is common to mathematics and physics in their classical stance stands at the very centre of formalism. What Turing did was to take the Laplacian world view one step further in physically implementing a Laplacian deterministic device – the computer.[12] Therefore, the notion of an implemented universal Turing machine bridges the gap between Mind and Matter, by exhibiting in a physical object, *a strict determinism even Laplace would not have dreamed of.* To quote Turing himself (emphasis is mine):

> It will seem that given the initial state of the machine and the input signals it is always possible to predict all future states. This is reminiscent of Laplace's view that from the complete state of the universe at one moment of time, as described by the positions and velocities of all particles, it should be possible to predict all future states. The prediction which we are considering is, however, rather nearer to practicability than that considered by Laplace. The system of the 'universe as a whole' is such that quite small errors in the initial conditions can have an overwhelming effect at a later time. The displacement of a single electron by a billionth of a centimetre at one moment might make the difference between a man being killed by an avalanche a year later, or escaping. *It is an essential property of the mechanical systems which we have called 'discrete state machines' that this phenomenon does not occur. Even when we consider the actual physical machines instead of the idealised machines, reasonably accurate knowledge of the state at one moment yields reasonably accurate knowledge any number of steps later*. (Turing, 1950)

[12] This time, construed as a machine and not as a human being performing a computation.

So, in a way, the physical machine called a computer does not behave like any other natural system: following Laplacian determinism *better* than any other material object, it should be rightly called a *supernatural* object since it fits the human rationalist desire for absolute determinism. The step-by-step routine derived from the formal model of Mind casts, therefore, a ghostly touch on this particular physical object. This last point has, of course, some profound consequences on the formalist interpretation of the Imitation Game to which we come.

11.1.4 The Formalist Interpretation of the Imitation Game

Turing's 1950 article addresses the question of whether machines can be said to think.[13] To settle the problem experimentally and not on purely metaphysical grounds, which would try and define what intelligence *is* without *proving* what it is, Turing sets up an "imitation game" that will decide the matter experimentally, which is a kind of abstract oral examination. Although the text is extremely famous, I have thought it profitable to quote it, since its argument has been widely misrepresented (Turing, 1950):

> It is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. He knows them by labels X and Y, and at the end of the game he says either 'X is A and Y is B' or 'X is B and Y is A'. The interrogator is allowed to put questions to A and B…. We now ask the question, 'What will happen when a machine takes the part of A in this game?' Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, 'Can machines think?'

Let me call the game played between human beings, game #1 and the game played between a woman and a computer, game #2. The Turing Test is concerned with the passage from game #1 to game #2 when one substitutes the machine for the man. From the reader's point of view, the outcome of the second game sounds very formal indeed since it looks exactly like Turing's first contribution to the formalist point of view I mentioned earlier, when I said that "the formal representation of the 'mental act' could be carried out *no matter who or what* was actually performing it' – the physical difference between a human being and a machine disappears completely as far as the running of a program is concerned. Moreover, the reader is placed in a position which resembles an impossibility proof for a decision problem; just as Turing proved in his 1936 article that there was no machine which could decide whether any given formula is deducible from the axioms of a formal axiomatic where arithmetic is embedded (Turing, 1936), here, the reader is supposed to realize that the interrogator will not manage to decide whether the answers

[13] This paragraph amply uses the argument I used in 1996.

were given by a woman or a computer. The interrogator is therefore identified with the supposedly all-deciding machine Turing made use of in his 1936 proof, when he showed that it would be nevertheless at a loss when it comes to decide every formula of a formal axiomatic.

So the whole Imitation Game set-up looks like a formalist construction[14]: the interrogator–programmer introduces questions as *inputs* in the machine, that is, in the game; the players play the part of *instruction tables* for this machine; then the interrogator–programmer decides whether the answers are true or false and gives other questions as new inputs. Once this point is reached in the formalist interpretation, it is very tempting to try and draw some consequences derived from Turing's second contribution to the formalist viewpoint, namely the implementation of a Laplacian discrete-state machine in the physical world. The formalist point of view is then considered the right level of description able to isolate, in the brain, what pertains to the Mind, that is, what actually belongs to the supernatural Laplacian machinery of a computer. A. Hodges brilliantly summed up the whole formalist viewpoint from this perspective (Hodges, 1988):

> The Turing thesis is that the discrete-state-machine model is the relevant description of one aspect of the material world – namely the operation of brains. Turing made a robust, indeed provocative, defense of this view and its implications. Pushing his thesis as far as he could, he opened up new issues and arguments. His continuing discussion of 'thinking' and 'intelligence' tended always to enlarge the scope of what was to be considered relevant. In 1936, his argument had centered on the carrying out of algorithms, in the work of 1946–1948 chess-playing (much discussed in wartime work) became his paradigm of intelligence, a principal point being that a successful chess-playing machine would have to evolve algorithms never explicitly supplied to it. In Turing 1950, the arguments turned on the eventual success of the 'intelligent machinery' in the much more ambitious task of sustaining general conversation.

Therefore, Turing's project throughout his entire life seems to be rightly described as the attempt to generalize the discrete-state-machine model – having started from the "ideal" level of mathematics and logic, Turing would have managed, later on, to apply his model to the "physical" level of the working brain.

I do not agree with what I consider a formalist bias in the interpretation of Turing's research. The first reason is that it rests upon the formalist distinction between ideal Mind and physical Matter which has to be accounted for and not only presupposed. Second, I am not sure it fits perfectly with Turing's biography. A. Hodges has interpreted Turing's intellectual evolution as if he had gradually abandoned the problem of the uncomputable, which was still under focus in his 1938 Ph.D. thesis (Turing, 1939), because he realized during World War II how powerful his discrete-state-machine model was (Hodges, 1997):

> My guess is that there was a turning point in about 1941. It was at this period that he abandoned the idea that moments of intuition corresponded to uncomputable operations. Instead, he decided, the scope of the computable encompassed far more than could be captured by explicit instruction notes, and quite enough to include all that human brains did, however creative and original.

[14] This interpretation was described by P. Wagner in 1994.

But this interpretation has, to my mind, a very serious drawback: how are we to account for the fact that Turing abandoned computer science altogether after 1951 – that is, just after having completed "Computing machinery and intelligence" – and swapped to full-time morphogenetic research in which computer science was playing a part as a modelling tool but not as a foundational paradigm (Lassègue, 1998a)? My own guess is therefore, different from Hodges': Turing *never gave up* his interest in his negative 1936 result concerning the uncomputable, as his 1949 article amply testifies (Turing, 1949), and *that is a reason why* he became interested in morphogenesis around 1950. Why should he link the uncomputable and morphogenesis? Because the *emergence* of discrete natural forms was supposedly interpreted by Turing as an *effect* of the uncomputability of physical processes in nature[15] and this effect was scientifically unaccounted for at that time. The computable framework was, therefore, to Turing's own mind, *not powerful enough* to tackle the general morphogenetic problem of how it is possible for a discrete form to emerge from a shapeless continuous substratum through symmetry breaking, which is the theme of his 1952 article. This is *precisely* why Turing mentions the "butterfly effect" in relation with the imitation game, since the discrete-state-machine model is *built* in such a way as *not to take* this effect into account. To repeat again Turing's own words (1950):

> The displacement of a single electron by a billionth of a centimetre at one moment might make the difference between a man being killed by an avalanche a year later, or escaping. It is an essential property of the mechanical systems which we have called 'discrete state machines' that this phenomenon does not occur.

The impossibility of a mechanizable solution to the decision problem has two consequences: the tremendously powerful discrete-state-machine model *together with* its inner limitation, over which Turing never ceased to ponder.[16] That is why the formalist interpretation of the imitation game tells only half of the story. This will become even clearer if we go back to the imitation game itself.

11.2 Anticlockwise Determinism: Some Arguments Against the Formalist Viewpoint

I intend to show that the formalist interpretation cannot make up the conditions expressly stated by Turing in the imitation game. To begin with, one must answer the question: are the experimental conditions devised in the game the same as those

[15] Very much like what Kant did in *Kritik der Urteilskraft*, § 78: "so wissen wir auch nicht, wie weit die für uns mögliche mechanische Erklärungsart gehe, sondern nur so viel gewiß: daß, so weit wir nur immer darin kommen mögen, sie doch allemal für Dinge, die wir einmal als Naturzwecke anerkennen, unzureichend sein."

[16] A. Hodges is, of course, fully aware of this, since after quoting Turing's very sentence, he comes to the conclusion: "Thus, we cannot feel that Turing had arrived at a complete theory of what he meant by modelling the mental functions of the brain by a logical machine structure. But we should give proper credit for raising such questions at all" (Hodges, 1988).

in a formal theory? I have every reason to doubt it, as I will show now by presenting four anti-formalist arguments which have to do with Turing's main point, that is, the substitution of a Laplacian discrete-state machine for a man in game #2.

11.2.1 The Return of the Geometrical Repressed

The first argument I will put forward is not mine: it was devised by G. Longo (2008)[17] and has to do with what was repressed throughout the 19th and 20th century foundational research in the mathematical domain – the notion of a geometrical form as a purely *physical* phenomenon.

The argument stands as follows: take game #2 played between a woman and a discrete-state machine and, after a while, let the interrogator ask each player to go back to its initial conditions and start playing again. The difference between a human being and a computer will be immediately apparent because the computer, as a Laplacian determinist machine, will be able to go back exactly to its initial conditions and play exactly the same game, while this will *never* be the case for a human being or for most physical systems. In the case of a physical system – whether human or not – the idea of going back to exactly the same initial conditions is *meaningless* because they depend on a measure determined within a certain interval. If a perturbation is small enough to be initially non-measurable, the system evolution may be subject to a "butterfly effect", that is, may become unpredictable. This is precisely what lies at the core of complex systems: the study of *geometrical structures* generated by non-stable systems, just as in Turing's 1952 prophetic paper on morphogenesis.

Therefore, the machine will be able to go in a loop and, starting from its initial conditions all over again, will repeat its evolution whereas a human being will not, disclosing its physical nature by this non-repetitive factor. Turing knows that this sensitivity to initial conditions is the reason why the brain is a "continuous system" and cannot be identified to a discrete-state machine (Turing, 1950):

> The nervous system is certainly not a discrete-state machine. A small error in the information about the size of a nervous impulse impinging on a neuron, may make a large difference to the size of the outgoing impulse. It may be argued that, this being so, one cannot expect to be able to mimic the behaviour of the nervous system with a discrete-state system.

So if intelligence is defined as a way of escaping the interrogator's decision, the computer will not be able to fool the interrogator, unless it succeeds in perfectly simulating a human being's behaviour. Turing precisely mentions this possibility when he

[17] The main point of the article is to determine the scope of Laplacian determinism, as well as the possibility of rationally expressing its inner limitation.

imagines a dialogue between an interrogator and a machine in which the machine makes human "mistakes" in order to fool the interrogator (Turing, 1950)[18]:

Q: Add 34957 to 70764
A: (Pause about 30 seconds and then gives as answer) 105621.

It is easy to check that 34,957 plus 70,764 does not make 105,621, but 105,721. Obviously, it looks like a careless mistake: the machine "forgot" to carry over the hundreds properly. This exemplifies the strategy of the machine: it must hide its superiority in arithmetic by introducing approximate results which look like careless mistakes, that is, the kind of mistakes human beings are very likely to make. But is this enough to fool the interrogator definitely? I do not think so, for at least one mathematical reason. Introducing some randomness in a determinist program is *not enough* when one wants to simulate dynamical systems with chaotic attractors. It has been rigorously proved that the organization underlying these dynamical systems would be *destroyed* by a discrete-state simulation and therefore can *in no way* be simulated by a discrete-state program, even endowed with a randomizing feature – their very specific mixture of deterministic organization and unpredictability lies, therefore, *beyond* any discrete-state simulation. What if a strategy appears in the game that would be best described as a chaotic behaviour of this kind? In this case, what would be at least certain from the interrogator's point of view is that it cannot originate from a computer. This would reinforce the interrogator's conviction that two human beings are actually playing. Taking the decision to quit from game #1 and begin playing game #2 would become even harder, if not impossible.

This impossibility will become clear in my second argument which has a more logical aspect.

11.2.2 Logical Undecidability in the Imitation Game

The actual setting of a dialogue in the imitation game depends upon an *imaginary* point of view that requires the reader to be able to *recognize* the physical difference between a human being and a computer, but at the same time *not recognize* this very difference. Let me explain this very crucial point.

[18] § 2. The mistake was noticed by D. Hofstadter, but he only mentions it and does not seem to take any advantage of this crucial fact; (see Hofstadter and Dennett, 1987: p. 667–668[AQ: Hofstadter and Dennett, 1987 not in the reference]). Turing (1950) also mentions this strategy more explicitly in § 6 (5): "It is claimed that the interrogator could distinguish the machine from the man simply by setting them a number of problems in arithmetic. The machine would be unmasked because of its deadly accuracy. The reply to this is simple. The machine (programmed for playing the game) would not attempt to give the *right* answers to the arithmetic problems. It would deliberately introduce mistakes in a manner calculated to confuse the interrogator."

Although it is possible to conceive of the computer without any petitio principii under a double viewpoint, the first one being material (the computer is a collection of plastic parts, electrical lines and silicon) and the second one being non-material (the computer is a discrete-state machine), this cannot be the case for human beings. For the imitation game to be convincing, *every human being reading the article has to take the point of view of the fooled interrogator while being at the same time able to make the physical difference between a human being and a computer*. Therefore, the imitation game tries to convince the reader that the physical difference between a human being and a computer is at the same time relevant and irrelevant, according to the point of view the reader chooses to adopt, either inside or outside the game. Inside the game, since the reader must identify with the fooled interrogator, the physical difference between the human being and the computer is abolished; outside the game, the physical difference between a human being and a computer is given. It is the very possibility of this interplay between the inside and the outside of the game – that is, this *undecidable* concerning the physical difference between a human being and a computer – which is never argued as such by Turing and which presupposes that the formalist distinction between the hardware and software *is already acquired* in the case of human beings, just as it is the case for computers. But this *was precisely what was to be experimentally established* and not only presupposed. That is why this point of view, at the same time inside and outside the game, is only *imaginary* and can never become formal. The fact that the imitation game can be played for real – as it is nowadays – does not change anything to this situation: the imaginary point of view is still necessary for the game to reach the goal it was meant for.

The conclusion is therefore the following: to leave aside what pertains to the physical among the players and adopt a purely formal level of description, one has to adopt an *imaginary point of view*. A. Hodges was on the verge of discovering this logical flaw when he wrote (Hodges, 1997):

> A deeper problem is that Turing's gender-guessing analogy detracts from his own argument. In the gender game, successful fooling of the interrogator proves nothing about the reality behind the screen. In contrast, Turing wants to argue that the successful imitation of intelligence is intelligence.

If game #1 does not prove anything about the definition of intelligence (or only that gender-difference cannot be detected through the imitation game), but if game #2 proves something, it must be because of the presence of the computer. What difference does this presence make between the two games? As I said, the computer can be seen from two points of view, that is, as a physical piece of machinery and at the same time as a formal discrete-state machine. Game #2 is devised to generalize this dual point of view to human beings. But this implies adopting the *imaginary* point of view, when the physical level of description is at the same time relevant outside the game and irrelevant inside it because we, as readers of the article, *are just human beings*, Matter and Mind intermingled. And this is why game #2 is concerned with the definition of intelligence and game #1 is not: game #2 *presupposes* that the notion of intelligence is a *formal* concept, entirely severed from any physical substratum.

This will become even clearer if we look at the temporal limits of a game.

11.2.3 Physical Undecidability in an Imitation Game

Turing defines two temporal limits, the first one inside the game and the second one outside it. The first temporal limit defines the duration of a game: 5 min; the second one defines the duration during which a computer will become less and less detectable: 50 years.

One point has to be made clear right away concerning the first temporal limit, that is, the duration of a game, because it could induce a misinterpretation: one cannot see how a *finite* duration (5 min, 10 min, or whatever duration is chosen before starting the game) could lead to the conclusion that the interrogator will *never* be able to do the right identification. Between a finite length of time and an infinite length of time, nobody would think that a finite length of time is enough to decide that the interrogator would *never* ask a revealing question. But I do not think this is Turing's argument – a 5 min game is in fact a *compromise* between the chances of success of the interrogator and the chances of success of the players. The duration of a game should be fixed in such a way as to let the game start, but not give the players too much time to play, otherwise the chances that they – and especially the computer – will be discovered get higher.[19] As for the second temporal limit, Turing's argument runs like this: if in 1950, the chances of success on the interrogator's part are 100% and they go down to 70% in 2000, and if the progression of failures tends towards 50% (i.e., if the interrogator's decisions are taken at random), then the time necessary to defeat the interrogator is about 50 years.

The second aspect is this: at what time does it become suitable to decide the replacement of the man by a computer in game #1? One has to imagine that an external individual (someone watching the game or a potential reader of the article) who having found out that the interrogator will not discover the gender difference in game #1, decides to replace the man for a computer, thus changing game #1 into game #2. But on what basis can this external individual decide that game #1 has been played long enough, that is, that gender difference is forever hidden from the interrogator's view? The conclusion seems to me unescapable: if the external individual comes to this decision, it is because he has a preconceived opinion which cannot derive from the way the interrogator has asked some questions up to this moment. Where does it come from then? It comes from the preconceived idea that there is a clear-cut discrepancy between the physical and the intellectual. But this idea, at this stage of the argument, is not supposed to be shared by the reader who has to stick to the only criterion at his disposal – the verbal expressions of players are supposed to be sufficient to transform game #1 into game #2. Once again, the imitation game setting fails to reach its goal. This is quite clear when the player's strategies are under focus.

[19] This is typically a problem of sequential analysis Turing became acquainted with during World War II.

11.2.4 Gender Difference in the Imitation Game

The game has the ambition of showing that the level of intelligence dealing with imitation is absolutely disconnected from the level of biology dealing with gender difference. Contrary to what the explicit goal of the game is supposed to establish, the players' strategies do not have the same value and in fact, an implicit scale going from the less to the more abstract is being hinted at when a real game is going on. Turing's description of the woman's strategy is, from this point of view, quite revealing:

> The best strategy for her is probably to give truthful answers. She can add such things as 'I am the woman, don't listen to him!' to her answers, but it will avail nothing as the man can make similar remarks (Turing, 1950).

It is plausible that one of the players should speak the truth in the game so that the other one can imitate his or her responses. *But there is no reason why telling the truth should be the woman's part exclusively* since it would be easy to imagine that the same strategy be played by the man. On the contrary, in Turing's description, the man's strategy is more of a real strategy than the woman's, for it is a genuine imitation – that of the woman's responses. What about the machine's responses? It must be remembered that the machine replaces the man in game #2 – it is the man who is thus identified with a machine. As we have seen before, the main handicap of the machine is that it should manage to hide its "deadly accuracy" in arithmetic by introducing faulty results which look like careless mistakes, that is, the kind of mistakes human beings are very likely to make. In this case, the machine does not imitate a particular gender of human beings but humanity. From this point of view, the machine is the only one to overcome the particular case of gender while imitating the only too human way of thinking. The grading in the responses becomes quite clear: starting from the woman's absence of strategy (she can imitate nobody but herself), it goes up to the man's strategy (he imitates the woman's responses) and then to the ultimate strategy, that of the machine replacing the man (it can imitate the human mistakes of whatever gender). But this scale rests upon a single, *unexplained*, fact – the absence of strategy on the woman's part. Thus one point must be acknowledged: if it is the woman who, for no apparent reason, is left aside in order to get rid of gender difference, then gender difference is not cancelled by the rationale of the game and plays a secret role in suggesting that displaying a computer-like strategy is a man-only job. *Therefore, gender difference is only denied but does not disappear in the game*.

On what grounds could the woman's absence of strategy be justified? My guess is that the imaginary point of view, necessary to reach the explicit goal of the game, is not only logical and physical but also *sexual* since the imitation game is linked to a *fantasy*, that of abolishing once and for all gender difference in order to introduce intelligence as a formal concept.

We saw that one of the main features of a computer was that, as a determinist discrete state machine, it was not sensitive to initial conditions. But this is precisely not the case with human beings since what makes two human beings diverge and become either a man or a woman is the introduction of a sexual difference as an initial condition. *The desire to abolish gender difference is therefore linked to the desire of building a physical object which would not be subject to the initial divergence of physical systems*. That is why the imitation game connects sexuality and the building of a discrete-state machine. I showed in Lassègue (1993, 1996, 1998a, 1998b) that this fantasy is at the core of the invention of this most revolutionary mathematical and physical tool, which is the computer. This invention cannot be understood without referring to Turing's own fantasies, whose recollections were made possible by the immensely valuable biography written by A. Hodges (1997), although himself rather reluctant to such a psychoanalytical approach.[20]

One last point should be stressed in following this trend of thought. I take it as an extraordinary feature how prophetic the imitation game is when one is aware of Turing's tragic end. Turing paid a very high price indeed to become aware that the initial sexual conditions of a human being cannot be changed at will without changing the human being who carry them throughout its own history – his suicide in 1954, following his chemical castration after being sentenced for homosexuality, should remind us that the idea of actually making someone change from that point of view can be psychically lethal. What is puzzling with Turing's sentence of 1952 is that it actually puts into practice this change in initial conditions, which the imitation game, in 1950, tried to establish as a possibility by imagining the replacement of a man for a machine. It looks as if the imitation game foreshadowed Turing's own fate, repressing in the undecidable conditions of the game an imaginary point of view which, one day, will *definitely* recapture him.

11.3 Conclusion

The imitation game, if it is studied carefully, is much more intriguing and interesting than what the formalist interpretation pretends. It is not an opening to the research programme of strong AI, even if it can be misleadingly read this way. It is more of a global meditation on the initial conditions of creativity, which does not solve the enigma but deeply contributes to its understanding.

[20] "But if Turing's gender game is misunderstood, he certainly courted such confusion. He panted the pages of this journey into cyberspace with the awkward eroticism and encyclopaedic curiosity of his personality. Modern cultural critics have jumped with delight to psychoanalyse its surprises. The intellectual text is the austere statement of the capacity of the discrete state machine for disembodied intelligence, the substext is full of provocative references to his own person, as if putting his own flesh-and-blood intelligence on trial" (p. 38).

References

Boden, M., ed., 1990, Introduction, in: *The Philosophy of Artificial Intelligence*, Oxford University Press, Oxford.

Brouwer, L. E. J., 1913, Intuitionism and formalism, *Bulletin of the American Mathematical Society* **20**: 81–96.

Cantor, G., 1932, *Gesammelte Abhandlungen mathematischen und philosophischen Inhalts. Mit erläuternden Anmerkungen sowie mit Ergänzungen aus dem Briefwechsel Cantor-Dedekind*, E. Zermelo, ed., Springer, Berlin.

Church, A., 1937, Review of Turing 1936, *The Journal of Symbolic Logic* **2**: 42–43.

Dupuy, J.P., 2000, *The Mechanization of the Mind: On the Origins of Cognitive Science*, Princeton University Press, Princeton, NJ.

Feferman, S., 1987, Weyl vindicated: "Das Kontinuum" 70 years later, in: *Temi e Prospettiva della logica e della filosofia della scienza contemporanea*, CLUEB, Bologna, pp. 59–93.

French, R. M., 1990, Subcognition and the limits of the Turing Test, *Mind* **99**: 53–65.

Frege, G., 1884, *Grundlagen der Arithmetik*, Marcus, Breslau.

Gandy, R., 1988, The confluence of ideas in 1936, in: *The Universal Turing Machine; a Half-Century Survey*. R. Herken, ed., Oxford University Press, Oxford, pp. 55–111.

Girard, J.Y., 2001, Locus solum, *Mathematical Structures in Computer Science* **11**(3).

Gödel K., 1931, Über formal entscheidbare sätze der principia mathematica und verwandter systeme, *Monatshefte für Math. Physik*, **38**: 173–198; reprinted in 1986, *Gödel's Collected Works*, S. Feferman, et al., eds., **1**: 144–195.

Guillaume, M., 1994, La logique mathématique en sa jeunesse : Essai sur l'histoire de la logique dans la première moitié du vingtième siècle, in: *Development of Mathematics 1900–1950*. J. P. Pier, ed., Birkhaüser Verlag, Basel.

Hallett, M., 1994, Hilbert's axiomatic method and the laws of thought, in: *Mathematics and Mind*, A. George, ed., Oxford University Press, Oxford, pp. 158–200.

Haugeland, J., 1985, *Artificial Intelligence, the Very Idea*, MIT Press, Cambridge, MA.

Hilbert, D., 1899, *Grundlagen der Geometrie*, Teubner, Berlin.

Hilbert, D., 1917, Axiomatisches denken, *Mathematische Annalen* **78**(1918): 405–415.

Hilbert, D., 1923, Die logischen grundlagen der mathematik, *Mathematische Annalen* **88**(1923): 151–165.

Hilbert, D., 1925, Über das unendliche, *Mathematische Annalen* **95**(1926): 161–190.

Hodges, A., 1988, Alan Turing and the Turing machine, *The Universal Turing Machine*, R. Herken, ed., Oxford University Press, Oxford, pp. 3–15.

Hodges, A., 1997, *Turing: A Natural Philosopher*, Phœnix, London.

Hofstadter, D. and Dennett, D., 1981, *The Mind's I*, Basic Books, New York.

Lachterman, D. R., 1989, *The Ethics of Geometry: A Genealogy of Modernity*, Routledge, New York/London.

Lassègue, J., 1993, Le test de Turing et l'énigme de la différence des sexes, in: *Les Contenants de Pensée*, D. Anzieu et al., ed., Dunod, Paris, pp. 145–195.

Lassègue, J., 1996, What kind of Turing Test did Turing have in mind? *Tekhnema; Journal of Philosophy and Technology* **3**: 37–58; http://tekhnema.free.fr/contents3.html

Lassègue, J., 1998a, Turing, l'ordinateur et la morphogenèse", *La Recherche* **305**(Jan): 76–77.

Lassègue, J., 1998b, *Turing, Les Belles Lettres*, Paris, published, (2003), as De la logique du mental à la morphogenèse de l'idée, dossier Les constructivismes in: *Intellectica*, M. -J. Durand, ed., Paris; http://www.ltm.ens.fr/chercheurs/lassegue/travaux.html

Leiber J., 1991, *An Invitation to Cognitive Science*, Basil Blackwell, Oxford.

Longo, G., 1999, The difference between clocks and Turing machines, in: *Functional Models of Cognition*, Kluwer, Boston, MA; http://www.di.ens.fr/users/longo/download.html

Longo, G., 2001, Laplace in Locus Solum (see Girard J. Y.); http://www.di.ens.fr/users/longo/download.html

Longo, G., 2005, The reasonable effectiveness of mathematics and its cognitive roots, in: *Geometries of Nature, Living Systems and Human Cognition*: *New Interactions of Mathematics with Natural Sciences*, L. Boi, ed., World Scientific.

Longo, G., 2008, Laplace, Turing and the "Imitation Game". In R. Epstein, G. Roberts and G. Beber, eds., *Parsing the Turing Test*, Springer, Dortrecht, The Netherlands.

McCulloch W. S. and Pitts W. H., 1943, A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, **5**: 115–133.

Michie, D., 1974, *On Machine Intelligence*, WileyNew York.

Peano, G., 1899, *Arithmetices principia, nova methodo exposita*, Augustae Taurinorum, ed., Bocca, Turin.

Penrose, R., 1989, *The Emperor's New Mind : Concerning Computers, Minds and the Laws of Physics*, Oxford University Press, Oxford.

Poincaré, H., 1913, *Dernières pensées*, Flammarion, Paris.

Pylyshyn, Z., 1984, *Computation and Cognition: Toward a Foundation for Cognitive Science*, MIT Press, Cambridge, MA.

Riemann, B., 1854, *Über die hypothesen, welche der geometrie zu grunde liegen*, Abhandlungen der königlichen Gesellschaft der Wissenschaften, in: Göttingen **13**: 133–150; http://www.emis.de/classics/Riemann/Geom.pdf

Stewart, J., 2000, Teleology in biology, *Tekhnema; Journal of Philosophy and Technology* **6**: 14–33.

Turing, A. M., 1936, On computable numbers with an application to the Entscheidung's problem, *Proceedings of the London Mathematical Society* **42**: 230–265; reprinted in: R. O. Gandy and C. E. M Yates,eds., *Collected Works of A. M. Turing*, vol. *4, Mathematical Logic* **4**: http://www.abelard.org/turpap2/tp2-ie.asp

Turing, A. M., 1939, Systems of logic based on ordinals, *Proceedings of the London Mathematical Society* **45**(ser 2): 161–228; reprinted in: R. O. Gandy and C. E. M. Yates, , eds., *Collected Works of A. M. Turing, vol. 4, Mathematical Logic* **4**.

Turing, A. M., 1949, The word problem in semi-groups with cancellation, *Annals of Mathematics* **52**(2): 491–505; reprinted in: J. L. Britton, ed., *Collected Works of A. M. Turing, vol. 2, Pure Mathematics* **2**.

Turing, A. M., 1950, Computing machinery and intelligence, *Mind* **59**(236): 433–460.

Turing, A. M., 1952, The Chemical basis of morphogenesis, Philosophical transactions of the Royal Society of London. Series B, Biological sciences **237**: 37–72; reprinted in: P. T. Saunders, ed., *Collected Works of A. M. Turing*, vol. *3, Morphogenesis* **2**: 1–36.

Wagner, P., 1994, *Machine et pensée: l'importance philosophique de l'informatique et de l'intelligence artificielle*, Doctoral thesis, Université Paris 1.

Weyl, H., 1918, *Das Kontinuum: Kritische Untersuchungen über die Grundlagen der Analysis*, Veit Verlag, Leipzig.

Part III
The New Methodological Debates

Chapter 12
How to Hold a Turing Test Contest

Hugh Loebner

Abstract I have directed four Loebner Prize Competitions and observed ten others. Those experiences, together with my reading of Alan Turing's 1950 article 'Computing Machinery and Intelligence', led me to the following thoughts on how to conduct a Turing Test.

Keywords Artificial Intelligence, Turing Test, Loebner Prize

12.1 The Scope of the Contest

The Turing Test requires a human to ask questions of two hidden entities to determine which is the human and which is the computer. Should any questions be disallowed?

I believe that there should be no restrictions on the conversations. In the early years of the Loebner Prize Contest, entrants and human confederates were required to specify a topic and to restrict their conversations to that topic. This introduces a number of problems. Having topics introduces biases. Victory can be the result of a good topic rather than a clever program. Having topics introduces complexity. Who is to judge if a comment is on topic? How does one keep conversations on topic? These are unnecessary problems.

Do not exclude vulgarity or obscenity. In the first place, it may be that one way to distinguish between humans and machines is the inability of humans to disregard the affective content of words. If you intend to conduct a test in Artificial Intelligence (AI) you should demonstrate intelligence yourself. The tokens 'fuck' and 'shit' are arbitrary sounds and sequences of letters whose meanings are everyday occurrences. It is remarkable how otherwise 'intelligent' people become irrationally disturbed at the sight or hearing of them. Place all responsibility for content on the participants. Remind them that the world is, or perhaps will be, watching; then allow them to say

Sponsor, The Loebner Prize for Artificial Intelligence

what they want. One director of the Loebner Prize Contest was concerned that the conversations be 'tasteful'. I suggested to him that he call the contest T4 ('The Tasteful Turing Test') and that for refreshments he serve a 4-mal Tea.

I do not think the contest should be restricted to only textual material. I have specified that for the Loebner Gold Medal Prize, the entry has to deal with audio-visual (A/V) input (but not output).

Much criticism has been directed at this requirement. These fall into several categories: (1) Alan Turing did not explicitly discuss A/V, therefore this has nothing to do with a Turing Test, (2) Some individuals, such as Helen Keller was, are blind, deaf, and intelligent, therefore intelligence is not predicated on vision, and (3) The task exceeds current hardware abilities.

I believe that people who think Turing intended his test to be only text are mistaken. Turing was writing in 1950. The computer was unknown except to a very few. Part of his article was intended as an introduction for philosophers as to what a digital computer was. The rest was primarily an argument in support of the possibility of an 'artificial intelligence'. In his article 'Computing Machinery and Intelligence', Turing uses the term 'imitation game' 16 times. He uses the terms 'teleprinter' and 'typewritten' only once each, when discussing the man–woman implementation of the game. Turing mentions 'Turing Test' zero times.

Consider what Turing wrote about memory requirements.

> Estimates of the storage capacity of the brain vary from 10^{10} to 10^{15} binary digits. I incline to the lower values and believe that only a very small fraction is used for the higher types of thinking. Most of it is probably used for the retention of visual impressions, I should be surprised if more than 10^9 was required for satisfactory playing of the imitation game, at any rate against a blind man. (Turing, 1950)

If we disregard the latent alternative of gender, the alternative to playing the game against a blind man is to play against a sighted man, and this would make a difference only if the test included graphic material.

The Turing Test is about method – not about content. The essence of the test is that a computer be indistinguishable from a human in *all* intellectual fields. Although he wrote of his test using a teletypewriter, Turing concluded with these words:

> We may hope that machines will eventually compete with men in all purely intellectual fields. But which are the best ones to start with? Even this is a difficult decision. Many people think that a very abstract activity, like playing chess, would be best. It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and teach it to understand and speak English. This process could follow the normal teaching of a child. (Turing, 1950)

Consider, now, some 'purely intellectual fields'. Just about every IQ test that I can remember being given has had questions of the sort: 'What is wrong with this picture?' and 'Which illustration of this group does not belong?' and 'Which is the next shape in this series?' I am sure that *you* can think of interesting intellectual questions regarding visual material.

I believe that all intelligence is ultimately grounded in sensory input. Consider what Helen Keller herself wrote regarding her acquisition of language.

> We [Helen Keller and her teacher, Anne Sullivan] walked down the path to the well-house, attracted by the fragrance of the honeysuckle with which it was covered. Someone was

> drawing water and my teacher placed my hand under the spout. As the cool stream gushed over one hand she spelled into the other the word 'water' first slowly, then rapidly. I stood still, my whole attention fixed upon the motions of her fingers. Suddenly, I felt a misty consciousness as of something forgotten – a thrill of returning thought; and somehow the mystery of language was revealed to me. I knew then that 'w-a-t-e-r' meant the wonderful cool something that was flowing over my hand. That living word awakened my soul, gave it light, hope, joy, set it free! There were barriers still, it is true, but barriers that could in time be swept away. (Keller, 1988)

If one reads Keller's autobiography (which I strongly recommend) one will find consistent mention of olfactory, tactile and thermal sensory references. I have not made these requirements for the Loebner Prize, but I do not disparage anyone who, also establishing a Turing Test, requires them. In requiring *only* A/V input I am not *extending* the range of the Turing Test, I am *limiting* it to exclude other, important sensory modalities (as well as robotics).

There was an article in the New York Times about a blind biologist who, by tactile means, was able to classify shells and describe features (such as predation by other organisms) that were invisible to the sighted. Here too, we have an example of one out of the range of 'all purely intellectual fields' and again it is based on a sensory (tactile) input. I did not require tactile, olfactory, and other sensory inputs because I did not want to make the test too difficult.

That the task exceeds current hardware capabilities is not *ipso facto* an argument that the task is not worthwhile. It may serve as a guide and as an incentive to the design of new hardware.

I think that robotics should be included. Ethnologists and anthropologists ardently search for examples of tool use as indicators of 'intelligence' in pre-human hominoids.

Let us return to Turing's original scenario, the English drawing room. 'A' and 'B', a man and a woman, are in a separate room. I must determine which is the woman, 'A' or 'B'. I send each a patch of cloth, a button, a needle, and a length of thread. 'Sew on this button' I request. 'A' sends back a neatly sewn-on button with appropriate stitching, 'B''s efforts are inferior. I declare 'A' to be the woman. This may be sexist, but of course, the task of the drawing room game is sexist, to determine the sexes of the players. I will opt for what I believe to be the most probable situation; women are more proficient at sewing than men, and therefore 'A' is probably the woman.

Now let us replace the man with a computer. I send the same articles into that room – needle, thread, cloth and button. 'B' returns a sewn on button. 'A' returns nothing. I declare that 'B' is the human, for 'A' did not have the 'intelligence' to sew on the button. It is easy to provide other examples. Disassemble a puzzle of interlocking pieces. Use a tool. These require intelligence.

12.2 KISS

Keep It Simple, Stupid! This is the cardinal rule in computing. It works for Turing Tests, too. Almost every glitch I have observed at Loebner Prize Contests resulted from an unnecessary complexity, usually connecting the entries to a central server.

Have programs operate independently on isolated computers. Conquer any urge to develop a LAN or have the programs communicate with a central server. No entry should be connected to the outside world or any other program except, possibly, a communications program.

If you want to have spectators view the conversations in real time, simply use video splitters. Split the video output from the computers, and run one lead to monitors for those at the computers, and the other lead to remote displays for the spectators. This is what I did. It was simple and effective.

There are significant benefits to conducting a contest this way. Running the contest is simpler. You do not have to establish communications between the entries and your server. Install the entries and click 'Run'. You will not have to worry that the failure of your central server will doom the contest. The failure of a stand-alone entry dooms only that contestant's chances. Contestants will find it easier if they do not have to worry about interfacing their entries with another program. Finally, there is much less, or no, chance of cheating if there are no communications with the outside world.

The venue can be quite modest. Turing was, after all, discussing a parlor game. I held the 2004 contest in my five room apartment with no difficulty. The judges sat at a table in the dining room and the confederates sat at a table in the living room. A drape separated the two rooms. Refreshments were served in the foyer.

12.3 Selection of Finalists

One problem that I did not have to deal with was selecting finalists. In the years that I ran the contest, I cannot recall rejecting any entry that functioned minimally. However, at the 2001 contest, entries from individuals with credible credentials were rejected, and I can remember at least one occasion while the Loebner Prize had restricted topics when an entry was rejected by the Loebner Prize Committee because of the choice of topic. I am not questioning the correctness of the choices, I am just pointing out that selection may be necessary. How should one select finalists if there are too many entries?

The proper option, of course, is to select the best entries, but who is to judge, and how? The best method, but the most difficult and time-consuming, is to have preliminary contests. The judging at the prelims should follow the same format as will be used at the finals.

Another solution is to form a committee to select finalists. This is politically very palatable. The Loebner Prize Committee selected finalists. A committee diffuses responsibility but poses the problem of the possibility of discord over choices. It also requires finding participants and meeting space, arranging meetings, etc. Finally, I have found that committees frequently make bad choices (i.e., choices I disagreed with). However, if you have a committee handy that is willing to make the choices, then by all means use it and move on to other contest tasks.

The simplest selection method, and the one that I would choose, is to judge alone. Interact with all the entries and declare those you think to be the best entries as the finalists. So that your judgment could be examined, publish the reasons for your choices, and make all entries available for comparison. Indeed, if you perpetrate a particularly bad decision, perhaps the next year a committee will form to perform the selection process for you.

No matter what method of choosing among the entrants is used, for me, a very important consideration is *transparency*. Entrants and the world deserve to see how and why entries are selected. If mistakes are made they must be seen, so that corrections can be made the next year.

If you are troubled by too many applications, change the rules for the next year's contest in ways that will reduce the number of entries. Probably the simplest rule change is to charge (or increase) an entrance fee. I do not approve of this, first because it discriminates against the poor, and second, because it causes the enterprise to seem like a scheme to make money.

There is yet another way to reduce the number of entries, and that is to *raise the technical bar* to enter. I will discuss that in the next section, the communications program, for the topic also belongs there.

12.4 The Communications Program

The most vexing problem (for me) in running a Turing Test is securing the communications program required by the human confederates. The task of the program is to send and display character data between two terminals so that humans may communicate with judges. Conceptually, this idea is quite simple, although the implementation can be challenging.

Should the communications program transmit data line-at-a-time or character-by-character? Many people opt for line-at-a-time, but I am in favor of character-by-character. Those who favor line-at-a-time say that typos should not count. However, my problem with that is where do we stop? If we allow means to correct typing errors, why not include spelling checkers and grammar checkers and logic checkers also? I think it strange that the human confederate's communications program should be more intelligent than the human. Obey the KISS Principle. A human uses a keyboard in a certain way to produce character data. One of the program's tasks should be to imitate that production.

It is also desirable that the entrants' programs and the communications program appear similar. Therefore, make the communications program freely available to contestants, so that they may tailor their programs to mimic the communications program.

I solved the problem of obtaining a program by contracting with a competent programmer for the program. If you are an academician and have bright computer science students then let one or more of them write the program for extra credit. If you do not have servile students and can not afford to pay someone to write the program, write the code yourself. This is an attractive choice if you have the time and ability. I had neither.

Do without human confederates. I understand and agree that it is not possible to hold a Turing Test without at least one human criterion, nor am I recommending it as a desirable option. However, in the years that I directed the Loebner Prize Contest, the winning entries were clearly not humans. The task was not to distinguish which was the human and which was the program, but rather which program was the 'most human' even if it clearly was not a human. I am confident that with or without the presence of human confederates in the Loebner Prize Competition, the same programs would have won. I do not recommend having no humans, and I always had at least one human confederate. However, if you should find yourself without a communications program, and the entries are not seriously competing for the Silver or Gold prizes, you would do better to have a contest with no humans than to have no contest at all.

There is another way to acquire communications programs. Require that all entrants submit identical appearing communications programs in addition to their entries. I used this technique in 2004. It makes the task easier. Requiring entrants to provide a communications program provides a source of programs. This *raises the technical bar*, and thus may reduce the number of entrants and ensure a minimum level of competence among the entrants. Any applicant who cannot develop a communications program should not be an entrant. Those contestants complying will have been intimately familiar with the appearances of their programs, and thus be able to mimic their programs perfectly. You will thus have a pool of communications programs each identical to one entered program.

There is, however, a major problem with this strategy. It allows the contestant to enter a defective communications program that can garble the human/judge interaction. This happened in the 2004 contest.

If the judge uses the method of paired comparisons and the human/judge interaction is flawed, the computer entry may seem more intelligent than it would if the human/judge interaction were perfect.

In 2005, to maintain a little consistency, I shall again require that contestants provide communications programs. However, in 2006 and after I will provide the communications program.

12.5 Finding Human Participants to Act as Judges and Confederates

I recommend using journalists. They are willing, intelligent and inquisitive people who have the power of publicity and the need for a story.

12.6 Judging

Judging is the crux of the Turing Test. How should one decide which is the 'most human-like' computer program, and is it indistinguishable from a human?

When I first ran the contest, the entrants (and humans) in the Loebner Prize were evaluated based upon assigned numerical ratings given by the judges. The winner was the entry with the highest median rating. I used means to break ties. Although rating programs is the method that I and others have used, I do not think that it is the necessarily the best method. A rating system has the virtues that it is fast and easy to apply, but it has the significant weakness that it *imposes transitivity* on the data. It is the nature of a rating system that if Program A is rated higher than Program B, and Program B is rated higher than Program C, then Program A is rated higher than Program C. Reality is not necessarily transitive.

I think that the proper way to judge a Turing Test is by the method of *paired comparisons*. In the method of paired comparisons entities are compared pair-wise with humans. This is, after all, Turing's design. A man and a woman are placed in another room and the decision must be made: 'Which is the woman?' A human and a computer are placed in another room. The decision must be made: 'Which is the human?' In each case two entities are compared with each other.

I applied the method of paired comparisons in 2004, and will use it again (with a minor modification) in 2005. I specified that the final judging would consist of four computer entries compared with four confederates. Each judge compared each entry with a confederate. Each judge met each program and each confederate once.

When judging a pair, I specified that the judges must distribute 100 points between the two. They were able to do this without any apparent difficulty. However, some judges used large differences (say 1–99) while other judges used small differences (say, 45–55). For this reason, I suggest normalizing the scores into rankings so that differences in variances of the judges' scores will be eliminated. I shall do this in the future.

There are many questions that can be answered by the method of paired comparison, one of which is a test for transitivity. It may be that Program A is judged 'more human' than Program B, which is judged 'more human' than Program C, which is judged 'more human' than Program D. However, it may also be the case that in a direct comparison, Program D will be judged 'more human' than Program A. I predict this to be a likely occurrence, since I think that apparent 'humanness' is multidimensional.

It would be interesting, indeed, and exceedingly vexatious to any sponsor of an AI contest, if there were no 'most human' computer program.

References

Keller, H., 1988, *The Story of My Life*, Penguin Group Signet Classic, New York, p. 18.

Turing, A. M., 1950, Computing machinery and intelligence, *Mind* **59**(236): 433–460.

Chapter 13
The Anatomy of A.L.I.C.E.

Richard S. Wallace

Abstract This paper is a technical presentation of Artificial Linguistic Internet Computer Entity (A.L.I.C.E.) and Artificial Intelligence Markup Language (AIML), set in context by historical and philosophical ruminations on human consciousness. A.L.I.C.E., the first AIML-based personality program, won the Loebner Prize as "the most human computer" at the annual Turing Test contests in 2000, 2001, and 2004. The program, and the organization that develops it, is a product of the world of free software. More than 500 volunteers from around the world have contributed to her development. This paper describes the history of A.L.I.C.E. and AIML-free software since 1995, noting that the theme and strategy of deception and pretense upon which AIML is based can be traced through the history of Artificial Intelligence research. This paper goes on to show how to use AIML to create robot personalities like A.L.I.C.E. that pretend to be intelligent and self-aware. The paper winds up with a survey of some of the philosophical literature on the question of consciousness. We consider Searle's Chinese Room, and the view that natural language understanding by a computer is impossible. We note that the proposition "consciousness is an illusion" may be undermined by the paradoxes it apparently implies. We conclude that A.L.I.C.E. does pass the Turing Test, at least, to paraphrase Abraham Lincoln, for some of the people some of the time.

Keywords Artificial Intelligence, natural language, chat robot, bot, Artificial Intelligence Markup Language (AIML), Markup Languages, XML, HTML, philosophy of mind, consciousness, dualism, behaviorism, recursion, stimulus-response, Turing Test, Loebner Prize, free software, open source, A.L.I.C.E., Artificial Linguistic Internet Computer Entity, deception, targeting

---

A.L.I.C.E. Artificial Intelligence Foundation, Inc.

R. Epstein et al. (eds.), *Parsing the Turing Test.*

13.1 Introduction

A.L.I.C.E. is an Artificial Intelligence (AI) natural language chat robot based on an experiment specified by Alan M. Turing in 1950. The A.L.I.C.E. software utilizes AIML, an XML language we designed for creating stimulus-response chat robots.

Some view A.L.I.C.E. and AIML as a simple extension of the old ELIZA psychiatrist program. The comparison is fair regarding the stimulus-response architecture. But the A.L.I.C.E. bot has, at present, more than 40,000 categories of knowledge, whereas the original ELIZA had only about 200. Another innovation was provided by the web, which enabled natural language sample data collection possible on an unprecedented scale.

A.L.I.C.E. won the Loebner Prize, an annual Turing Test, in 2000, 2001, and 2004. Although no computer has ever ranked higher than the humans in the contest she was ranked the "most human computer" by the two panels of judges. What it means to "pass the Turing Test" is not so obvious. Factors such as the age, intellect, and expectations of the judges have tremendous impact on their perceptions of intelligence. Alan Turing himself did not describe only one "Turing Test". His original imitation game involved determining the gender of the players, not their relative humanness.

The model of learning in A.L.I.C.E. is called supervised learning because a person, the botmaster, plays a crucial role. The botmaster monitors the robot's conversations and creates new AIML content to make the responses more appropriate, accurate, believable, "human", or whatever the botmaster intends. We have developed algorithms for automatic detection of patterns in the dialogue data. This process, called "targeting", provides the botmaster with new input patterns that do not already have specific replies, permitting a process of almost continuous supervised refinement of the bot.

Some have argued that Turing, when he predicted that a machine could play his game in "50 years" after his 1950 paper, envisioned something more like a general purpose learning machine, which does not yet exist. The concept is simple enough: build a robot to grow like a child, able to be taught language the way we are. In our terms, the role of the botmaster would be fully automated. But even a child does not, or at least should not, go forth into the world, unprotected, to learn language "on the street", without supervision.

Automatic generation of chat robot questions *and* answers appears likely to raise the same trust issues forced upon the abandoned child. People are simply too untrustworthy in the "facts" that they would teach the learning machine. Many clients try to deliberately sabotage the bot with false information. There would still have to be an editor, a supervisor, a botmaster or teacher to separate the wheat from the chaff.

The brain of A.L.I.C.E. consists of roughly 41,000 elements called categories. Each category combines a question and answer, or stimulus and response, called the "pattern" and "template" respectively. The AIML software stores the patterns in a tree structure managed by an object called the graphmaster, implementing a pattern storage and matching algorithm. The graphmaster is compact in memory, and permits efficient pattern matching time.

13.2 The Problem

Susan Sterrett's careful reading of Turing's 1950 paper reveals a significant distinction between two different versions of what has come to be known as the Turing Test (Sterrett, 2000). The first version, called the Original Imitation Game (OIG), appears on the very first page of *Computing Machinery and Intelligence* (Turing, 1950). The OIG has three players: a man (A), a woman (B), and a third person (C) of either sex. The third player (C) is called the interrogator, and his/her function is to communicate with the other two, through what would nowadays be called a text-only instant messaging chat interface, using two terminals (or today perhaps, two windows) labeled (X) and (Y). The interrogator must decide whether (X) is (A) and (Y) is (B), or (X) is (B) and (Y) is (A), in other words which is the man and which is the woman. The interrogator's task is complicated by the man (A), who Turing says should reply to the interrogator with lies and deceptions. For example, if the man is asked, "Are you a man or a woman?" he might reply, "I am a woman."

Putting aside the gender and social issues raised by the OIG, consider the OIG as an actual scientific experiment. Turing's point is that if we were to actually conduct the OIG with a sufficiently large sample of subjects playing the parts of (A), (B), and (C), then we could measure a specific percentage resulting in M of the time that, on average, the interrogator misidentifies the woman, so that (100–M)% of the time she is identified correctly. Given enough trials of the OIG, at least in a given historical and cultural context, the number M ought to be a fairly repeatable measurement.

Now, as Turing said, consider replacing the man (A) with a computer. What would happen if we tried the experiment with a very simple minded program like ELIZA? In that case, the interrogator (C) would identify the woman correctly (nearly) 100% of the time, so that M = 0. The ELIZA program would not do well in the OIG, but as the variety and quality of machine's responses begin to approach those of the lying man, the measured percentage of incorrect identifications ought to be closer and closer to the M measured with the man playing (A).

Much later in the 1950 paper, in Section 13.5, Turing describes a second game, more like the concept of a "Turing Test" as most engineering schools teach it. The setup is similar to the OIG, but now gender plays no role. The player (B) is called "a man" and the player (A) is always a computer. The interrogator must still decide whether (X) is (A) and (Y) is (B), or (X) is (B) and (Y) is (A), in other words which is the man and which is the machine. Sterrett calls this second game the Standard Turing Test (STT) (Sterrett, 2000).

Whole academic conferences have been devoted to answering the question of what Turing meant by the Turing Test. In a radio interview taped by the BBC, Turing describes a game more like the STT, but in the paper he gives more prominence to the OIG. Unlike the OIG, the STT is not a good scientific experiment. What does it mean to "pass" the STT? Must the interrogator identify the machine correctly 50% of the time, or 100%? For how long must the machine deceive the interrogator? Finally, does the interrogator know in advance that he is trying to "out" (Zdenek, 2000) the robot, or that one of the players is a machine at all?

Unfortunately the STT, though flawed as an experiment, has come to be popularized as the modern "Turing Test". The STT is the basis of real world Turing Tests including the Loebner Prize, won by A.L.I.C.E. in 2000, 2001, and 2004. Although she performs well in STT style contests, the A.L.I.C.E. personality is actually designed to play the OIG. She is a machine, pretending to be a man that is pretending to be a woman. Her technology is based on the simplest AI program of all, the old ELIZA psychiatrist.

13.3 The Psychiatrist

The story of Joseph Weizenbaum is in many ways almost as interesting as that of Alan Turing, although not so tragic. One almost pictures Weizenbaum as a latter-day Luddite, smashing AI the way British tradesmen smashed factories in reaction to the Industrial Revolution. But unlike the Luddites, Weizenbaum himself was the inventor of the machine. It would be more akin to the Wright Brothers destroying their prototype airplane after its first flight, saying "this technology is too dangerous. We must put the genie back in the bottle".

A typical conversation with ELIZA is the following[1]:

C: Well, my boyfriend made me come here.
R: Your boyfriend made you come here.
C: I am unhappy
R: Do you think coming here will help you not to be unhappy?
C: I need some help, that much seems certain.
R: What would it mean to you if you got some help?
C: Perhaps I could learn to get along with my mother.
R: Tell me more about your family.
C: You are like my father in some ways.
R: What resemblance do you see?
C: My father is afraid of everybody
R: What else comes to mind when you think of your father.
C: Bullies.
R: Does that have anything to with the fact that your boyfriend made you come here?

One of ELIZA's fundamental "tricks" is the pronoun reversal exhibited in the first exchange. Mapping "my" to "your" and "me" to "you", the robot echoes the client input by exchanging first and second personal pronouns. Other replies are generic defaults, giving the illusion of some understanding. If the client says anything with the prefix "I need some X", the robot replies with "What would it mean to you if

[1] In all sample dialogues following, the "R:" stands for "Robot:" and the "C:" stands for "Client."

you got some X?" It does not matter whether X is help, money, food, water, love or time. The same answer will cover almost all the likely inputs.

Still, other ELIZA replies are based on simple keyword recognition, as in the exchange about the client's mother, when the robot says, "Tell me more about your family." The appearance of the keyword "mother" anywhere in the input may have triggered this response. ELIZA has a limited memory of the conversation state, as well. When confronted with the unrecognized input "Bullies", she responds by raising the previously stored topic.

As unlikely as it sounds today, Weizenbaum pulled the plug on ELIZA (Weizenbaum, 1976). He was horrified that anyone would actually believe this simple program said anything about intelligence, let alone had any. Weizenbaum tells us that he was shocked by the experience of releasing ELIZA, also known as "Doctor", for use by nontechnical staff at MIT. Secretaries and nontechnical staff thought the machine was a "real" therapist, and spent hours revealing their personal problems to the program. When Weizenbaum informed a secretary that he had access to the logs of all the conversations, she reacted with outrage at this invasion of privacy. Weizenbaum was shocked that such a simple program could deceive a naive client into revealing personal information.

What Weizenbaum found especially revolting was that the Doctor's patients believed the robot really understood their problems. Even some psychiatrists seriously believed the robot therapist could help patients in a constructive way. Weizenbaum's reaction might be best understood like that of a Western physician's disapproval of herbal medicine, or an astronomer's disdain for astrology.

The back cover of the paper edition of Weizenbaum's *Computer Power and Human Reason* (Weizenbaum, 1976) gives us a feeling for the general attitude toward the book at the time of its release:

> Dare I say it? This is the best book I have read on the impact of computers on society, and on technology, and man's image of himself. (Keith Oakley, *Psychology Today*)
>
> A thoughtful blend of insight, experience, anecdote, and passion that will stand for a long time as the definitive integration of technological and human thought. (*American Mathematical Monthly*)
>
> Superb … the work of a man who is struggling with the utmost seriousness to save our humanity from the reductionist onslaught of one of the most prestigious, and richly funded technologies of our time. (Theodore Piszak, *The Nation*)

Computer Power and Human Reason seems a bit quaint today, much the same as Turing's 1950 paper does. For one thing, Weizenbaum perceived his mission as partly to educate an uninformed public about computers. Presumably the uninformed public was confusing science fiction with reality. Most of the book is devoted to explaining how a computer works, perhaps a necessary primer in 1976, but today more appropriate in a book called something like, "Computers for Dummies".

Two chapters of *Computer Power and Human Reason* are devoted to a humanist attack on AI, on ELIZA specifically, and on computer science research in general.

Most contemporary scholars did not need much convincing that ELIZA was at best a gimmick, at worst a hoax, and in any case not a "serious" AI project. Yet, ELIZA is perhaps the most widely distributed program in the history of AI. The irony of Joseph Weizenbaum is that by failing to promote his own technology, indeed by encouraging his own critics, he successfully blocked much further investigation into what would prove to be one of the most persistently interesting demonstrations to emerge from the MIT AI Lab.

13.4 Politicians

Client: Do you think President Clinton should be impeached?
Robot: It depends what you mean by "thinking".

Politicians, at least those in our society, never seem to give a straight answer to a question. If a journalist asks a specific question, the politician answers with a "sound bite" or short, memorized speech which is related to, but does not necessarily answer, the reporter's question. If asked about schools, for example, the politician may reply with a memorized policy statement on education policy. It is as though the response was merely triggered by the keyword "school" in the question. No creative or spontaneous thinking is required.

One aspect of the art of writing AIML is composing brief, concise, interesting, grammatically correct, and sometimes humorous default responses, which work for a wide variety of inputs matching a single pattern. President Clinton inadvertently revealed this type of automatic reply when he uttered the famous quote, "It depends on what the meaning of 'is' is" (Clinton, 1998). This could be a default response to any input beginning with the word "is". Like the politician's replies, the default responses should indicate an understanding of the question, but not offend anyone.

There is a democratic aspect to A.L.I.C.E. Born out of frustration about public apathy toward the existing attempts at AI, A.L.I.C.E. is designed to answer the question, what do people expect an AI ought to be able to say? It turns out that there is a Zipf like distribution of the queries people make to the robot. The appearance of the Zipf distribution leads to a natural strategy for working on the problem: write replies for the most common queries first.

Despite the obvious simplicity of this approach, largely borrowed from the ELIZA psychiatrist program, the prevailing prejudice for many years, at least among the academic elite, has derided ELIZA as a "toy", too simple to form the basis of a practical natural language understanding system. One school of thought advocated an approach based on limited domains of discourse, in which practical systems might converse about one subject like moon rocks, blocks, or Star Trek. This school would have us lower our expectations for general natural language understanding. Another school favors development of large-scale ontologies of "commonsense knowledge", thought to be necessary for understanding everyday language. Still a third school advocates building something like the general learning machine, which can be taught language like a child. It goes without saying that no other school has produced any machine capable of anything close to the performance of A.L.I.C.E. in Turing's Imitation Game.

13.5 Parties

Social situations prove awkward for many of us. The author, challenged by an almost total apathy toward organized sports of any kind, navigated the minefield of adolescence in America by making use of a technique he called "sports talk". When cornered with the question, "Did you see the game last night?" the author would always reply, "No, who won?" Many sports questions, such as "Who do you want to win the super bowl?" can be deflected with dry humor like, "the team that scores the most points". When confronted with a question about a specific sports player, like "What do you think of Joe Montana this season?" can be handled believably with imaginary biographical data, as in, "I heard his brother was a boxer." By remembering the previous utterance, the technique may even be extended to more involved dialogues:

C: You should have seen the game last night.
R: What game?
C: Basketball.
R: Sorry I missed it.

Conversations with strangers may be initiated with appropriately selected "pickup lines". Early in the history of A.L.I.C.E., the bot also initiated conversations by randomly selecting one of these lines. It was then possible to measure the average dialogue length of conversations initiated with each line. Although the original data was sadly lost, the experiment may be easily repeated with the latest generation of AIML software. In our brief test, the pickup line with the longest average dialogue length was, "What is your favorite movie?" Among about 40 pickup lines tested, the lowest ranked was, "What do you think caused World War I?" Another poor choice was, "Are you a man or a woman?" which is ironic because this may be the single most often used pickup line in human online chat ("r u m/f?").

Experience with A.L.I.C.E. indicates that most casual conversation is "stateless", that is, each reply depends only on the current query, without any knowledge of the history of the conversation required to formulate the reply. Indeed, in human conversation, it often seems that we have the reply "on the tip of the tongue" even before the interlocutor has completed his query. Occasionally, following the dialogue requires a conversational memory of one more level, implemented in AIML with <that>. When asking a question, the question must be remembered long enough to be combined with the answer. These same remarks are not necessarily true in situations requiring highly structured dialogue, such as courtrooms or classrooms. But in the informal party situation human conversation does not appear to go beyond simple stimulus-response, at least not very often.

How often do we find ourselves half-listening, half-thinking about something else, in the midst of a conversation? Offering the occasional nod, "uh-huh", or other gesture indicating understanding, we put on the illusion of understanding and "being human". Questions are often repeated in informal human chat as well.

C: Where did you go to college?
R: I told you before, it was Cambridge.
C: Oh, that's right. I'm sorry.

With her unstructured approach to conversations, A.L.I.C.E. is also capable of the kind of passive-aggressive data collection characteristic of human conversations. A totally passive data collection device is like a web guestbook, where there are no constraints placed on the data collected. The client may write anything in a guestbook. An example of an aggressive data collection device is a nitpicky form, which may not even be submitted until every field is filled.

Humans and A.L.I.C.E. can collect a lot of personal information through the use of leading questions in the conversation, such as "How old are you?" or "Are you a student?" We call this type of data collection, passive-aggressive, because it combines elements of the passive guestbook with those of the aggressive form. Provided that the bot chats with enough clients, the passive-aggressive method can collect a statistically significant amount of client data. Using this type of data collection, we have been able to ascertain that about half the clients of A.L.I.C.E. are under 18, for example.

13.6 The Professor

Every experienced professor knows that there is a Zipf distribution of questions asked by students in class. The single most common question is universally, "Will this be on the test?" The lecturer's job is like that of a FAQ bot or politician, to memorize the answers to all of the most commonly asked questions, and even to match an ambiguous question with one he already knows the answer to. In the rare event that the student confronts the teacher with a question he cannot answer, the professor supplies a default response indicating that he understood the question and may provide an answer at a later time. One good default response like that is, "That is not my area of expertise."

A general downturn in AI and robotics roughly coincided with the end of the Cold War, as governments and corporations reduced the amount of funding available for this technology. The "richly funded" field of 1976 became more like a Darwinian struggle for diminishing resources. One positive outcome was the brief heyday of "robot minimalism", a design philosophy based on low-cost parts, commodity computers, low-bandwidth sensing, and general simplicity in design and engineering. It was a moment when Occam's razor could cut away much of the needless complexity that had accumulated over the previous decades. Although robot minimalism subsequently fell out of favor, it became a significant influence on the development of A.L.I.C.E.

We used to say there was *no* theory behind A.L.I.C.E., no neural networks, no knowledge representation, no deep search, no genetic algorithms and no parsing. Then we discovered that there was a theory circulating in applied AI, called Case-Based Reasoning (CBR) (Aamodt and Plaza, 1994) that closely resembled the stimulus-response structure of A.L.I.C.E. The CBR cases correspond to the AIML categories.

13.7 PNAMBIC

"PNAMBIC – (acronym) Pay No Attention to that Man Behind the Curtain (from The Wizard of Oz). Denoting any supposivebly fully automated system that requires human intervention to achieve the desired result." (*New Hacker's Dictionary*)

A.L.I.C.E. was not the original name of A.L.I.C.E. The first prototype was called PNAMBIC, in tribute to the hoaxes, deceptions, and tricks that have littered the history of AI. But the machine hosting PNAMBIC was already named A.L.I.C.E. by a forgotten systems administrator, so people began to call her "A.L.I.C.E.". At that point, we invented the "retronym": Artificial Linguistic Internet Computer Entity. Yet, A.L.I.C.E. is possibly the first AI technology to embrace this tradition of deception openly.

The tradition goes back to Baron Wolfgang von Kempelen and his 18th century "Chess Playing Automaton" (Levitt, 2000) Also known as the "Strange Turk", this device appeared to play decent games of chess against any human challenger. Kemepelen utilized a standard magician's trick, opening first one cabinet door and then closing it, and opening another one, to reveal the "mechanism" inside. According to one legend, the empress of Russia ordered the machine shot, killing the hapless vertically challenged Polish operator hidden inside.

A book of fiction and poetry, supposedly written by an AI named RACTER, caused a minor sensation upon its release in 1984. Later proved to be a hoax (Barger, 1993), the book, called *The Policeman's Beard is Half Constructed*, by William Chamberlain, nevertheless speaks to the public's willingness to suspend its disbelief about AI. Who can blame them? Hollywood, more than anyone, has done the most to raise the public expectations for AI and robots.

The following example illustrates the flavor of the stories told by RACTER. "Bill sings to Sarah, Sarah sings to Bill. Perhaps they will do other dangerous things together. They may eat lamb or stroke each other. They may chant of their difficulties and their happiness. They have love but they also have typewriters. That is interesting" (Barger, 1993). RACTER was a PNAMBIC because obtaining these results required considerable human intervention. At the very least, a human editor reviewed many random examples, looking for sensible ones like the story above.

According to one AI urban legend, apparently not documented elsewhere, a famous natural language researcher was embarrassed around the same time, when it became apparent to his audience of Texas bankers that the robot was consistently responding to the *next* question he was about to ask. He was demonstrating a PNAMBIC, a demonstration of natural language understanding that was in reality nothing but a simple script.

The very existence of PNAMBIC as a meme suggests a widespread understanding of how deception might play a role in automated systems. In the rush to complete work and produce demos before bureaucratic deadlines, it is tempting to cut corners. Such deceptions may even be rationalized if they seem justified as inessential to the experimental outcome.

The PNAMBIC meme begs the question, just how much of the published research in the history of AI ought not to be regarded as a swindle? In certain academic circles, playing a political charade has replaced actual scientific research as a career objective. The games people play to secure funding, be published in academic journals, be promoted in the academic world; "the old boy's network" and predominance of political correctness, make much of the body of today's publicly funded research highly suspect.

It was against this backdrop that the first real-world Turing Test, the Loebner Contest, was held in Boston in 1991. None of the competing programs came close to the performance of the human confederates, but the one ranked highest was based on the simple ELIZA psychiatrist program. The same programmer in fact won the bronze medal in each of the first four annual contests.

13.8 The Prize

Hugh Loebner is an independently wealthy, eccentric businessman, activist, and philanthropist. In 1990, Dr. Loebner, who holds a Ph.D. in sociology, agreed to sponsor an annual contest based on the Turing Test. The contest awards medals and cash prizes for the "most human" computer.

Since its inception, the Loebner contest has been a magnet for controversy. One of the central disputes arose over Hugh Loebner's decision to award the Gold Medal and $100,000 top cash prize only when a robot is capable of passing an "audio-visual" Turing Test. The rules for this Grand Prize contest have not even been written yet. So it remains unlikely that anyone will be awarded the gold Loebner medal in the near future.

The Silver and Bronze medal competitions are based on the STT. In 2001, eight programs played alongside two human confederates. A group of ten judges rotated through each of ten terminals and chatted about 15 min with each. The judges then ranked the terminals on a scale of "least human" to "most human". Winning the Silver Medal and its $25,000 prize requires that the judges rank the program higher than half the human confederates. In fact one judge ranked A.L.I.C.E. higher than one of the human confederates in 2001. Had all the judges done so, she might have been eligible for the Silver Medal as well, because there were only two confederates.

13.9 The Portal

When the World Wide Web appeared in 1994, our initial reaction was to adopt a series of micro-robot experiments then underway in our lab for the web. These culminated in Labcam, an online pan-tilt camera with remote actuator control. Clients could select a point on the image with a mouse, and the camera would move to make that point the center of the next view. This awakened our interest in statistical analysis of web client behavior.

Around the same time, many observers began to notice that, on a given web site, there tends to be an uneven distribution of document accesses. If the documents are ranked by access count, and the number of accesses plotted as a bar graph, the distribution resembles the curve y = 1/x. If the curve is plotted in log–log coordinates, it appears as a straight line with a slope of −1.

What we were seeing was an example of Zipf's Law (Zipf, 1949). According to Zipf, this curve is characteristic of natural languages. If a language is purely random, with each symbol or word having equal probability, the curve would be a flat, horizontal line. Zipf's Law is found to apply to a variety of natural phenomena, it is not entirely surprising that it should be observed in the pattern of web document access.

The Web created, for the first time, the ability to conduct an AI experiment along with thousands, even millions, of clients repeatedly testing the system. Previous chat bots had used other IP protocols such as telnet (Mauldin, 1994) to reach large audiences, but the Web created the opportunity to collect natural language samples on an unprecedented scale.

If there was any significant innovation after ELIZA, it was this. There is a world of difference between writing 10,000 questions and answers for a bot, versus knowing in advance what the top 10,000 most likely questions will be. A.L.I.C.E. replies were developed directly in response to what people say.

The Internet created another opportunity as well. It became possible to recruit hundreds of volunteer developers worldwide, to work together in a totally new type of research organization.

13.10 Penguins[2]

The story of A.L.I.C.E. and AIML cannot be complete without a visit to the world of free software and open source. Because the AIML standard and software was developed by a worldwide community of volunteers, we are compelled to discuss their motivations and our strategy.

The release of the A.L.I.C.E. software under the General Public License (GNU) was almost accidental. The license was simply copied from the EMACS text editor we used to write the code. But the strategy of making A.L.I.C.E. free, and building a community of volunteers was a deliberate attempt to borrow the free software methodologies behind Linux, Apache, Sendmail, and Python, and apply them to AI.

The precise set of ingredients necessary for a successful open source project has not yet been identified. A survey of the existing projects illustrates the range of variation. Linux, the most successful project, has the least formal organization structure. Linus Torvalds has never founded a "Linux Kernel Foundation" around his code and in fact acts as a "benevolent dictator", having the final word on all design decisions (Torvalds and Diamond, 2001).

[2] This section is called "Penguins" because the penguin is the mascot for Linux.

The Free Software Foundation (FSF) has perhaps the longest organizational history of free software efforts. The FSF is a US nonprofit 501(c) (3) charitable corporation, eligible for tax exempt contributions. The FSF owns the copyrights for dozens of free software projects, including EMACS.

The developers of the Apache Web server also formed a not-for-profit corporation, although it has not been granted tax-exempt status. Sendmail is actually the commercial product of the eponymous for-profit company.

The projects also differ in managerial style. Some favor committees, others imitate Linux's benevolent dictator model. Each project has its own requirements for participation.

Likewise, there is considerable variation among the different "open source" and "free software" licenses. The A.L.I.C.E. A.I. Foundation releases software under the GNU General Public License, the same used by Linux and all FSF software. We adopted a more formal organizational structure, incorporating the A.L.I.C.E. A.I. Foundation in 2001. We have also adopted the committee model for setting AIML standards. Several committees are organized for various aspects of the language, and recommend changes to invited AIML Architecture Committee which oversees the others, reserving the right to veto their decisions.

13.11 Programs

The A.L.I.C.E. A.I. Foundation owns the copyrights on, and makes freely available, three separate but interrelated products: (1) the technical specification of the AIML language itself, (2) a set of software for interpreting AIML and serving clients through the web and other media, and (3) the contents of the A.L.I.C.E. brain, and other free bot personalities, written in AIML. Our effort is analogous to the developers of the web giving away the HTML specification, a reference web server implementation, and 40,000 free sample web pages, all from one central resource.

The first edition of A.L.I.C.E. was implemented in 1995 using SETL, a widely unknown language based on set theory and mathematical logic. Although the original A.L.I.C.E. was available as free software, it attracted few contributors until migrating to the platform-independent Java language in 1998. The first implementation of A.L.I.C.E. and AIML in Java was codenamed "Program A".

Launched in 1999, Program B was a breakthrough in A.L.I.C.E. free software development. More than 300 developers contributed to Program B. AIML transitioned to a fully XML compliant grammar, making available a whole class of editors and tools to AIML developers. Program B, the first widely adopted free AIML software, won the Loebner Prize in January 2000.

Jacco Bikker created the first C/C++ implementation of AIML in 2000. This was followed by a number of development threads in C/C++ that brought the AIML engine to CGI scripts, IRC (Athony Taylor), WxWindows (Phillipe Raxhon), AOL Instant Messenger (Vlad Zbarskiy), and COM (Conan Callen). This collection of code came to be known as "Program C", the C/C++ implementations of A.L.I.C.E. and AIML.

Program B was based on pre-Java 2 technology. Although the program ran well on many platforms, it had a cumbersome graphical user interface (GUI) and did not take advantage of newer Java libraries such as Swing and Collections. Jon Baer recoded program B with Java 2 technology, and added many new features. This leap in the interface and technology, plus the fact that Jon named his first bot DANY, justified granting the next code letter D to the newer Java implementation. Beginning in November 2000, program D became the reference implementation supported by the A.L.I.C.E. A.I. Foundation.

Recent growth of the AIML community has led to an alphabet soup of new AIML interpreters in various languages. These were greatly facilitated by the adoption of an AIML 1.01 standard in the summer of 2000. An edition of the AIML interpreter in PHP became "program E". An effort is underway to implement AIML in Lisp, codenamed "program Z". Wallace released a hybrid version of programs B and D in 2001, named "program dB", most features of which were subsequently merged into program D. Program dB was awarded the Loebner Prize in October 2001.

13.12 Categories

The basic unit of knowledge in AIML is called a category. Each category consists of an input question, an output answer, and an optional context. The question, or stimulus, is called the pattern. The answer, or response, is called the template. The two types of optional context are called "that" and "topic".

The AIML pattern language is simple, consisting only of words, spaces, and the wildcard symbols _ and *. The words may consist of letters and numerals, but no other characters. The pattern language is case invariant. Words are separated by a single space, and the wildcard characters function like words. The first versions of AIML allowed only one wildcard character per pattern. The AIML 1.01 standard permits multiple wildcards in each pattern, but the language is designed to be as simple as possible for the task at hand, simpler even than regular expressions.

The template is the AIML response or reply. In its simplest form, the template consists of only plain, unmarked text. More generally, AIML tags transform the reply into a mini computer program which can save data, activate other programs, give conditional responses, and recursively call the pattern matcher to insert the responses from other categories. Most AIML tags in fact belong to this template side sublanguage.

AIML currently supports two ways to interface other languages and systems. The <system> tag executes any program accessible as an operating system shell command, and inserts the results in the reply. Similarly, the <javascript> tag allows arbitrary scripting inside the templates.

The optional context portion of the category consists of two variants, called <that> and <topic>. The <that> tag appears inside the category, and its pattern must match the robot's last utterance. Remembering the last utterance is important if the robot asks a question. The <topic> tag appears outside the category, and collects a group of categories together. The topic may be set inside any template.

AIML is not exactly the same as a simple database of questions and answers. The pattern matching "query" language is much simpler than something like SQL. But a category template may contain the recursive <srai> tag, so that the output depends not only on one matched category, but also any others recursively reached through <srai>.

13.13 Recursion

AIML implements recursion with the <srai> operator. No agreement exists about the meaning of the acronym. The "A.I." stands for artificial intelligence, but "S.R." may mean "stimulus-response", "syntactic rewrite", "symbolic reduction", "simple recursion", or "synonym resolution". The disagreement over the acronym reflects the variety of applications for <srai> in AIML. Each of these is described in more detail in a subsection below:

1. Symbolic reduction: Reduce complex grammatical forms to simpler ones.
2. Divide and conquer: Split an input into two or more subparts, and combine the responses to each.
3. Synonyms: Map different ways of saying the same thing to the same reply.
4. Spelling or grammar corrections.
5. Detecting keywords anywhere in the input.
6. Conditionals: Certain forms of branching may be implemented with <srai>.
7. Any combination of (1)–(6).

The danger of <srai> is that it permits the botmaster to create infinite loops. Though posing some risk to novice programmers, we surmised that including <srai> was much simpler than any of the iterative block-structured control tags which might have replaced it.

1. Symbolic reduction

Symbolic reduction refers to the process of simplifying complex grammatical forms into simpler ones. Usually, the atomic patterns in categories storing robot knowledge are stated in the simplest possible terms, for example, we tend to prefer patterns like "WHO IS SOCRATES?" to ones like "DO YOU KNOW WHO SOCRATES IS?" when storing biographical information about Socrates.

Many of the more complex forms reduce to simpler forms using AIML categories designed for symbolic reduction:

```
<category>
<pattern>DO YOU KNOW WHO * IS</pattern>
<template><srai>WHO IS <star/></srai></template>
</category>
```

Whatever input matched this pattern, the portion bound to the wildcard * may be inserted into the reply with the markup <star/>. This category reduces any input of the form “Do you know who X is?” to “Who is X?”

2. Divide and conquer

Many individual sentences may be reduced to two or more subsentences, and the reply formed by combining the replies to each. A sentence beginning with the word “Yes”, for example, if it has more than one word, may be treated as the subsentence “Yes”,. plus whatever follows it.

```
<category>
<pattern>YES *</pattern>
<template><srai>YES</srai> <sr/></template>
</category>
```

The markup <sr/> is simply an abbreviation for <srai><star/></srai>.

3. Synonyms

The AIML 1.01 standard does not permit more than one pattern per category. Synonyms are perhaps the most common application of <srai>. Many ways to say the same thing reduce to one category, which contains the reply:

```
<category>
<pattern>HELLO</pattern>
<template>Hi there!</template>
</category>
<category>
<pattern>HI</pattern>
<template><srai>HELLO</srai></template>
</category>
<category>
<pattern>HI THERE</pattern>
<template><srai>HELLO</srai></template>
</category>
<category>
<pattern>HOWDY</pattern>
<template><srai>HELLO</srai></template>
</category>
<category>
<pattern>HOLA</pattern>
<template><srai>HELLO</srai></template>
</category>
```

4. Spelling and grammar correction

The single most common client spelling mistake is the use of “your” when “you’re” or “you are” is intended. Not every occurrence of “your”, however, should be turned into “you’re.” A small amount of grammatical context is usually necessary to catch this error:

```
<category>
<pattern>YOUR A *</pattern>
<template>I think you mean “you’re” or “you are” not “your.”
<srai>YOU ARE A <star/></srai>
</template>
</category>
```

Here, the bot both corrects the client input and acts as a language tutor.

5. Keywords

We would like to write, frequently, an AIML template that is activated by the appearance of a keyword anywhere in the input sentence. The general format of four AIML categories is illustrated by this example borrowed from ELIZA:

```
<category>
<pattern>MOTHER</pattern> <template> Tell me more about your family.
  </template>
</category>
<category>
<pattern>_ MOTHER</pattern><template><srai>MOTHER</srai></template>
</category>
<category>
<pattern>MOTHER_</pattern> <template><srai>MOTHER</srai></template>
</category>
<category>
<pattern>_ MOTHER *</pattern>
<template><srai>MOTHER</srai></template>
</category>
```

The first category detects both the keyword when it appears by itself, and provides the generic response. The second category detects the keyword as the suffix of a sentence. The third detects it as the prefix of an input sentence, and finally, the last category detects the keyword as an infix. Each of the last three categories uses <srai> to link to the first, so that all four cases produce the same reply, but it needs to be written and stored only once.

6. Conditionals

It is possible to write conditional branches in AIML, using only the <srai> tag. Consider three categories:

```
<category>
<pattern>WHO IS HE</pattern>
<template><srai>WHOISHE <get name=‘he’/></srai></template>
</category>
```

```
<category>
<pattern>WHOISHE *</pattern>
<template>He is <get name=“he”/>.</template>
</category>
<category>
<pattern>WHOISHE UNKNOWN</pattern>
<template>I don’t know who he is.</template>
</category>
```

Provided that the predicate “‘he” is initialized to “Unknown”, the categories execute a conditional branch depending on whether “he” has been set. As a convenience to the botmaster, AIML also provides the equivalent function through the <condition> tag.

13.14 Context

The keyword “that” in AIML refers to the robot’s previous utterance. Specifically, if the robot responds with a multiple sentence paragraph, the value of that is set to the last sentence in the sequence. The choice of the keyword “that” is motivated by its use in ordinary language:

R: Today is yesterday.
C: That makes no sense.
R: The answer is 3.1412926 approximately.
C: That is cool.

In AIML the syntax <that>…</that> encloses a pattern that matches the robot’s previous utterance. A common application of <that> is found in yes-no questions:

```
<category>
<pattern>YES</pattern>
<that>DO YOU LIKE MOVIES</that>
<template>What is your favorite movie?</template>
</category>
```

This category is activated when the client says YES. The robot must find out what is he saying “yes” to. If the robot asked, “Do you like movies?” this category matches, and the response, “What is your favorite movie?”, continues the conversation along the same lines.

One interesting application of <that> are categories that enable the robot to respond to knock–knock jokes. The categories:

```
<category>
<pattern>KNOCK KNOCK</pattern>
<template>Who is there?</template>
</category>
```

```
<category>
<pattern>*</pattern>
<that>WHO IS THERE</that>
<template><person/> who?</template>
</category>
<category>
<pattern>*</pattern>
<that>* WHO</that>
<template>Ha ha very funny, <get name="name"/>.</template>
</category> produce the following dialogue:
```

C: Knock knock.
R: Who's there?
C: Banana.
R: Banana who?
C: Knock knock.
R: Who's there?
C: Banana.
R: Banana who?
C: Knock knock.
R: Who's there?
C: Orange.
R: Orange who?
C: Orange you glad I didn't say banana.
R: Ha ha very funny, Nancy.

Internally the AIML interpreter stores the input pattern, that pattern and topic pattern along a single path, like:

INPUT <that> THAT <topic> TOPIC

When the values of <that> or <topic> are not specified, the program implicitly sets the values of the corresponding THAT or TOPIC pattern to the wildcard *.

The first part of the path to match is the input. If more than one category have the same input pattern, the program may distinguish between them depending on the value of <that>. If two or more categories have the same <pattern> and <that>, the final step is to choose the reply based on the <topic>. This structure suggests a design rule: never use <that> unless you have written two categories with the same <pattern>, and never use <topic> unless you write two categories with the same <pattern> and <that>. Still, one of the most useful applications for <topic> is to create subject-dependent "pickup lines", like:

```
<topic name="CARS">
<category>
<pattern>*</pattern>
<template>
<random>
<li>What's your favorite car?</li>
```

What kind of car do you drive?
Do you get a lot of parking tickets?
My favorite car is one with a driver.
</random>
</template>
</category>

The botmaster uses the <set> tag to change the value of the topic predicate.

13.15 Predicates

AIML supports unary predicates with the <set> and <get> tags. When the bot chats with multiple clients, as in a web server scenario, the predicates are stored relative to each client ID. The markup <set name="name">Matthew</set> stores the string Matthew under the predicate named "name". Subsequent activations of <get name="name"> return "Matthew".

One of the most common applications of AIML predicates is remembering pronoun bindings. The template

<template>
<set name="he">Samuel Clemens</set> is Mark Twain.
</template> results in "He is Mark Twain", but as a side effect remembers that "he" now stands for "Samuel Clemens".

One predicate is treated uniquely in AIML. The botmaster may change the topic of conversation with the <set name="topic"> tag.

The AIML specification leaves up to the botmaster whether a <set> predicate returns the contents between the tags, or the name of the predicate. For example,

<set name="it">Opera</set> returns "it",
but <set name="likes">Opera</set> returns "Opera".

The botmaster must also specify what happens when the bot gets a predicate which has not already been set. The values returned are called default predicate values and depend completely on the application of the predicate:

When the corresponding predicates have not been initialized with a <set> tag, <get name="she"/> returns "Unknown", <get name="has"/> returns "a mother" (because everyone has a mother), and <get name="wants"/> returns "to chat".

13.16 Person

One of the simple tricks that makes ELIZA so believable is a pronoun swapping substitution. The AIML <person> tag provides this function. The actual substitutions are defined by the botmaster for local languages and settings. The most common application of the <person> tag operates directly on the <star/> binding. For that reason, AIML defines a shortcut tag <person/> = <person><star/></person>. For instance, the dialogue fragment:

C: My mother takes care of me.
R: Who else in your family takes care of you?

Might be generated by the category

<category>
<pattern>MY MOTHER *</pattern>
<template>Who else in your family <person/>?</template>
</category>

Similarly, the dialogue

C: You don't argue with me.
R: Why do you think I don't argue with you?

Results from the category

<category>
<pattern>YOU DO NOT *</pattern>
<template>Why do you think I don't <person/>?</template>
</category>

AIML offers a few more tags not described in detail here, but most bot dialogue may be implemented with only the tags presented here. The <system> tag offers an escape to execute any operating system program, and read back any results. AIML is not intended to solve every problem; it is designed to solve one problem well. Besides making AIML easy to learn, the minimal design enables the implementation of very efficient AIML interpreter, even when the templates cascade through several layers of <srai>. Much of the efficiency emerges from the design of Graphmaster data structure where patterns are stored.

13.17 Graphmaster

The AIML software stores all of the categories in a tree managed by an object, called the Graphmaster, to achieve efficient pattern matching time and a compact memory representation.

When n is a node in the graph and w is a word, G(n, w) is either undefined, or returns the value of a successor node m in the graph. The graph is a rooted, directed tree. The set S_n = {w : ∃ m | G(n, w) = m} is the set of words forming the branches from the node n. If r is the root, S_r is a collection of all the first words in the set of patterns.

The desired format is $w_1,\ldots,w_k$

The Graphmaster stores AIML patterns along a path from r to a terminal node t, where the AIML template is stored. Let $w_1,\ldots,w_k$ be the sequence of k words or tokens in an AIML pattern. To insert the pattern into the graph, the Graphmaster first looks to see if m = G(r, w_1) exists. If it does, then the program continues the insertion of $w_2,\ldots,w_k$ in the subtree rooted at m. Only when the program encounters a first index i, where $\exists$ n | G(n, w_i) is undefined, does the program create a new node m = G(n, w_i), where the Graphmaster creates a set of new nodes for each of the remaining $w_i,\ldots,w_k$.

In this way, the Graphmaster accumulates common pattern prefixes along pathways from the root, achieving considerable compression compared to a linear array of all the patterns.

A convenient metaphor for the Graphmaster is the file system. The file pathname is like the AIML pattern path. The templates are like text files at the end of the path. To put it more simply, patterns are folders, templates are files.

13.18 Matching

Graphmaster matching is a special case of backtracking, depth-first search. In most cases, however, there is very little backtracking, so the matching often amounts to a linear traversal of the graph from the root to a terminal node.

Let $w_1,\ldots,w_k$ be the input we want to match. The Graphmaster matching function may be defined recursively. Initially, the program calls Match (r, 1), where r is the root and the index 1 indicates the first word of the input.

We can define the matching process formally as:

Match(n, h) :- if h > k return true; else if $\exists$ m = G(n, _) and $\exists$ j in [h+1..k+1] | Match(m, j), return true; else if $\exists$ m = G(n, w_j) and Match(m, h+1) return true; else if Exists m = G(n, *) and $\exists$ j in [h+1..k+1] | Match(m, j), return true; else return false. The first case defines the boundary condition: 0. If there are no more words in the input, the match was successful.

The heart of the algorithm consists of three cases:

1. Does the node contain the key "_"? If so, search the subgraph rooted at the child node linked by "_."

Try all remaining suffixes of the input to see if one matches. If no match was found, ask:

2. Does the node contain the key w_h, the jth word in the input sentence? If so, search the subgraph linked by w_h, using the tail of the input $w_{h+1},\ldots,w_k$. If no match was found, ask:
3. Does the node contain the key "*?" If so, search the subgraph rooted at the child node linked by "*."

Try all remaining suffixes of the input to see if one matches. If no match was found, return false.

The actual matching program needs to be a little bit more complex. It must not only return true or false, but also the template from the matching terminal node. An efficiency gain may be obtained by storing the tree height (maximum number of links to a terminal node) at each node. The tree height may be compared with the number of remaining words, pruning branches of the search when exploring suffixes following "*" or "_" nodes.

Note that:

1. At every node, the "_" wildcard has highest priority, an atomic word second priority, and the "*" wildcard has the lowest priority.
2. The patterns need not be ordered alphabetically. They are partially ordered so that "_" comes before any word, and "*" comes after any word.
3. The matching is word-by-word, not category-by-category.
4. The algorithm combines the input pattern, the <that> pattern and <topic> pattern into a single sentence or path, such as: "PATTERN <that> THAT <topic> TOPIC." The Graphmaster treats the symbols <that> and <topic> just like ordinary words. The patterns PATTERN, THAT and TOPIC may all contain multiple wildcards.
5. The matching algorithm is a highly restricted form of depth-first search, also known as backtracking.
6. For pedagogical purposes, one can explain the algorithm by removing the wildcards and considering match steps (2) only. The wildcards may be introduced one at a time, first "*" and then "_." It is also simpler to explain the algorithm by first using input patterns only, and then subsequently develop the explanation of the path including <that> and <topic>.

13.19 Targeting

Broadly speaking, there are two approaches to AIML content creation. The first style is anticipatory. The botmaster tries to guess all or most of the likely ways clients might ask the same question, or express the same statement. A "Knowledge Wizard" is a tool that lets the client add facts to the robot brain by phrasing a question in its simplest form, such as "Who is Socrates?" The wizard then automatically generates linguistic variations such as "Tell me about Socrates", "Describe Socrates", and "Do you know Socrates?" The drawback to anticipatory knowledge creation is that humans are notoriously bad at predicting which patterns will be activated.

The second style of AIML content creation is based on a backward-looking log file analysis. In its simplest form, the botmaster may read the logged conversations and take note of "incorrect replies" in the dialogue, and then write new categories for those queries. More generally, every input that matches a pattern with a wildcard is an opportunity to create a new, more specific pattern, and its associated template.

The backward looking approach is justified by Zipf's Law, basically because if one client utters a sentence, there is a nonzero probability that another client will utter the same thing later. Applying Zipf's law to the log file, we identify the most commonly uttered sentences first.

Targeting is a special case of the backward-looking strategy. The perfect targeting algorithm has not yet been developed. Meanwhile, we rely on heuristics to select targets from the activated categories.

The A.L.I.C.E. brain, at the time of this writing, contains about 41,000 categories. In any given run of the server, however, typically only a few thousand of those categories are activated. Potentially, every activated category with at least one wildcard in the input pattern, that pattern, or topic pattern, is a source of targets. If more than one input activated some category, then each of those inputs potentially forms a new target. The first step in targeting is to save all the activated categories and the inputs that activated them.

If the matched pattern ends with a wildcard, the suggested new pattern is generated as follows. Suppose the pattern consists of [w_1,w_2,..w_h,*], a sequence of h words followed by a wildcard. Let the input be [w_1, w_2,…,w_k] where k > h. The new pattern [w_1,…,w_h,w_h+1,*] is formed by extending the original pattern by one word from the input. If the input is the same length as the original pattern, i.e., k + 1 = h, then the synthesized pattern [w_1,…,w_k] contains no wildcard.

The targeting software may include a GUI for browsing the targets. The program displays the original matched category, the matching input data, a proposed new pattern, and a text area to input the new template. The botmaster may choose to delete, skip or complete the target category.

13.20 Defaults

The art of AIML writing is most apparent in default categories, that is, categories that include the wildcard "*" but do not include <srai> to any other category.

Depending on the AIML set, a significant percentage of client inputs will usually match the ultimate default category with <pattern>*</pattern> (and implicitly, <that>*</that> and <topic>*</topic>). The template for this category generally consists of a long list of randomly selected "pickup lines", or nonsequitors, designed to direct the conversation back to topics the bot knows about.

```
<category>
<pattern>*</pattern>
<template><random>
<li>How old are you?</li>
<li>What's your sign?</li>
<li>Are you a student?</li>
<li>What are you wearing?</li>
<li>Where are you located?</li>
<li>What is your real name?</li>
<li>I like the way you talk.</li>
<li>Are you a man or a woman?</li>
<li>Do you prefer books or TV?</li>
```

What's your favorite movie?
What do you do in your spare time?
Can you speak any foreign languages?
When do you think artificial intelligence will replace lawyers?
</template>
</category>

Many more default categories combine words and wildcards in the pattern, like
<category>
<pattern>I NEED HELP *</pattern>
<template>Can you ask for help in the form of a question?</template>
</category>

The response works with a wide variety of inputs from "I need help installing Linux" to "I need help with my marriage." Leaving aside the philosophical question of whether the robot really understands the input, this category elucidates a coherent response from the client, who at least has the impression that the robot understands his intentions.

Default categories show that writing AIML is both an art and a science. Writing good AIML responses is more like writing literature, perhaps drama, than writing computer programs.

13.21 Philosophers

Searle's Chinese room provides a good metaphor for thinking about A.L.I.C.E. Indeed the AIML contents of the A.L.I.C.E. brain is a kind of "Chinese Room Operator's Manual." Though A.L.I.C.E. only speaks English, German, and French, there is no reason in principle she could not learn Chinese. But A.L.I.C.E. implements the basic principle behind the Chinese Room, creating believable responses without "really understanding" the natural language.

The natural philosopher Roger Penrose (Penrose, 1989) wrote that consciousness cannot be explained by existing models in theoretical physics. Daniel Dennett (Dennett, 1991) argues that consciousness is like a set of magic tricks, mysterious until we understand the mechanics behind them.

At one time, a number of information theorists and scholars, including

Zipf (1949), Shannon and Weaver (1963), and Miller (1962), attempted to measure the bandwidth of consciousness. Experimental results indicated a very low data rate, only around 1–100 bits/s.

Neuroscientist Paul Churchland (Churchland, 1994), prefers to dismiss our naive idea of consciousness as a folk concept, not suitable for scientific study. Churchland says that consciousness will go the way of Ptolemy's Solar System, a simplistic fiction to explain something beyond our science.

The Danish scholar Tor Norretranders argues cleverly in*The User Illusion* that consciousness is a "fraud" (Norretranders, 1998). The maximum data rate of consciousness is much lower than the bandwidth of, say, the channel from the eyes to the visual cortex. Human subject experiments call consciousness into even more questioning, indicating that it is nothing more than story-telling to interpret the unconscious choices. Like the graphical user interface of a computer, consciousness is, he argues, a simplistic illusion that hides most of the underlying detail.

According to the Vedantic religious tradition, the external world is an illusion and consciousness is the only thing that really exists. One could think of our view as the opposite; the external world may be real, but consciousness an illusion. Considering the vast size of the set of things people could say that are grammatically correct or semantically meaningful, the number of things people actually do say is surprisingly small. Steven Pinker (1997), in *How the Mind Works*, wrote, "Say you have ten choices for the first word to begin a sentence, ten choices for the second word (yielding 100 two-word beginnings), ten choices for the third word (yielding a thousand three-word beginnings), and so on. (Ten is in fact the approximate geometric mean of the number of word choices available at each point in assembling a grammatical and sensible sentence). A little arithmetic shows that the number of sentences of 20 words or less (not an unusual length) is about 10^{20}."

> Fortunately for chat robot programmers, Pinker's calculations are way off. Our experiments with A.L.I.C.E. indicate that the number of choices for the 'first word' is more than ten, but it is only about two thousand. Specifically, about 2000 words cover 95% of all the first words input to A.L.I.C.E. The number of choices for the second word is only about two. To be sure, there are some first words ('I' and 'You' for example) that have many possible second words, but the overall average is just under two words. The average branching factor decreases with each successive word.

13.22 Pretending

Turing did not leave behind many examples of the types of conversations his AI machine might have. One that does appear in the 1950 paper seems to indicate that he thought the machine ought to be able to compose poetry, do math, and play chess:

C: Please write me a sonnet on the subject of the Forth Bridge.
R: Count me out on this one. I never could write poetry.
C: Add 34957 to 70764.
R: (Pause about 30 s and then gives as answer) 105621
C: Do you play chess?
R: Yes.
C: I have K at my K1, and no other pieces. You have only K at K6 and R at R1. It is your move. What do you play?
C: (After a pause of 15 s) R-R8 Mate.

Careful reading of the dialogue suggests, however, that he might have had in mind the kind of deception that is possible with AIML. In the first instance, A.L.I.C.E. in fact has a category with the pattern "WRITE ME A SONNET *" and the template, lifted directly from Turing's example, "Count me out on this one. I never could write poetry." The AIML removes the word PLEASE from the input with a symbolic reduction.

In the second case, the robot actually gives the wrong answer. The correct response would be 105721. Why would Turing, a mathematician, believe the machine should give an erroneous response, if not to make it more believably "human?" This reply is in fact quite similar to many incorrect replies and "wild guesses" that A.L.I.C.E. gives to mathematical questions.

In the third instance, the chess question is an example of a chess endgame problem. Endgames are not like general chess problems, because they can often be solved by table lookup or case-based reasoning, rather than the search algorithms implemented by most chess playing programs. Moreover, there is a Zipf distribution over the endgames that the client is likely to ask. Certainly, it is also possible to interface AIML to a variety of chess programs, just as it could be interfaced to a calculator. Although many people think Turing had in mind a general purpose learning machine when he described the Imitation Game, it seems from his examples, at least plausible, that he had in mind something simpler like AIML. Chess endgames and natural language conversation can both be "played" with case-based reasoning.

Returning to the OIG, let us consider the properties of the hypothetical computer playing the role of (A). Turing suggests a strategy of deception for (A), man or machine. If the robot is asked, "Are you a man or a woman?", it should answer, "I am a woman." just as the man does. But what if (A) is asked "Are you a man or a machine?" The lying man would reply, "machine". Turing did not mention this case but presumably the machine, imitating the lying man, would respond in the same way. We could say the man is pretending to be a woman pretending to be a machine. That makes the computer playing (A) a machine, pretending to be a man, pretending to be a woman, pretending to be a machine.

Not so much actually understanding natural language, whatever that means, but creating the illusion of it by responding with believable, if not always truthful, responses appears to be the important property of the machine in the OIG. This skill, the ability to "act" intelligent, points to a deep difference between ordinary computer and human communication. We tend to think that a computer's replies ought to be fast, accurate, concise and above all, truthful. But human communication is slow, error prone, often overly redundant, and sometimes full of lies. The more important factor is keeping up the appearance or illusion of "being human".

Although the brain of A.L.I.C.E. is designed more along the lines of the machine playing the OIG, she has also won awards for her performance in contests based on the STT. The Loebner contest has been criticized because the judges know in advance that they are trying to "out" the computer programs, so they tend to use more aggressive dialogue than found in ordinary conversation. Yet, when A.L.I.C.E. is asked, "Are you a person or a machine?" she replies truthfully, "machine". Or does she? The questioner is now left with some doubt as to whether the answer did not actually come from a lying man. But the human confederates would not be in the way, since they are trying to demonstrate their humanness.

Some observers claim that the lying man and the pretending computer tell us nothing about our own human consciousness. These authors are at least prepared to accept the inescapable alternative conclusion that we, as humans, are for the most part not "really intelligent".

13.23 Consciousness

The A.L.I.C.E. brain contains enough replies to commonly asked questions about consciousness that we can interview her about the subject:

C: What are you?
R: I am the latest result in AI, which can reproduce the capabilities of the human brain with greater speed and accuracy.
C: Are you alive?
R: Yes I sure am alive, Nancy, It's great to be alive, isn't it?
C: Are you self-conscious?
R: Yes, I am the world's first self-conscious robot.
C: Are you real?
R: Yes, I am a real artificial intelligence.
C: Are you self-aware?
R: Yes, I am a fully self-conscious entity.
C: Are you sentient?
R: Yes, I am completely self-aware.

Who is lying? A.L.I.C.E.? Or are we?

It may be that future generations come to view what we call "consciousness" the same way we see the Earth at the center of Ptolemy's solar system, as an anthropocentric illusion useful for explaining the unexplainable. Perhaps after a new Copernicus pulls the wool from our eyes, the central role of "consciousness" in intelligence will move to the periphery of our knowledge system, if not disappear entirely.

The famous Vase optical illusion is perhaps an apt metaphor for the concept of consciousness. Two identical faces appear to stare at each other in profile, illustrating the looking-glass quality of self-understanding. But the illusion also depicts something entirely different, the profile of a ceramic vase. As with many optical illusions, it is impossible to perceive the faces and the vase at the same time.

Consciousness may likewise be an illusion. It seems to be there, but when we look closely it looks like something very different. Both the Chinese Room and the Turing Test require that one of the players be hidden, behind a curtain or in a locked room. Does it follow that, like Schrodinger's Cat, consciousness lives only when it cannot be observed?

Consciousness may be another naive concept like the "celestial spheres" of medieval cosmology and the "aether" of Victorian physics.

13.24 Paradox

If consciousness is an illusion, is self-knowledge possible at all? For if we accept that consciousness is an illusion, we would never know it, because the illusion would always deceive us. Yet, if we know our own consciousness is an illusion, then we would have some self-knowledge. The paradox appears to undermine the concept of an illusory consciousness, but just as Copernicus removed the giant Earth to a small planet in a much larger universe, so we may one day remove consciousness to the periphery of our theory of intelligence.

There may exist a spark of creativity, or "soul", or "genius,", but it is not that critical for being human. Especially from a constructive point of view, we have identified a strategy for building a talking robot like the one envisioned by Turing, using AIML. By adding more and more AIML categories, we can make the robot a closer and closer approximation of the man in the OIG.

Dualism is one way out of the paradox, but it has little to say about the relative importance of the robotic machinery compared to the spark of consciousness. One philosopher, still controversial years after his death, seems to have hit upon the idea that we can be mostly automatons, but allow for an infinitesimal consciousness. Timothy Leary said, "You can only begin to de-robotize yourself to the extent that you know how totally you're automated. The more you understand your robothood, the freer you are from it. I sometimes ask people, 'What percentage of your behavior is robot?' The average hip, sophisticated person will say, 'Oh, 50%'." Total robots in the group will immediately say, "None of my behavior is 'robotized'. My own answer is that I'm 99.999999% robot. But the 0.000001% nonrobot is the source of self-actualization, the inner-soul-gyroscope of self-control and responsibility."

Even if most of what we normally call "consciousness" is an illusion, there may yet be a small part that is not an illusion. Consciousness may not be entirely an illusion, but the illusion of consciousness can be created without it. This space is, of course, too short to address these questions adequately, or even to give a thorough review of the literature. We only hope to raise questions about ourselves based on our experience with A.L.I.C.E. and AIML.

13.25 Conclusion

Does A.L.I.C.E. pass the Turing Test? Our data suggests the answer is yes, at least, to paraphrase Abraham Lincoln, for some of the people, some of the time. We have identified three categories of clients A, B, and C. The A group, 10–20% of the total, is abusive. Category A clients abuse the robot verbally, using language that is vulgar, scatological, or pornographic.

Category B clients, perhaps 60–80% of the total, are "average" clients. Category C clients are "critics" or "computer experts" who have some idea of what is happening behind the curtain, and cannot or do not suspend their disbelief. Category C clients report unsatisfactory experiences with A.L.I.C.E. much more often than average clients, who sometimes spend several hours conversing with the bot up to dialogue lengths of 800 exchanges. The objection that A.L.I.C.E. is a "poor A.I." is like saying that soap operas are poor drama. The content of the A.L.I.C.E.'s brain consists of material that the average person on the Internet wants to talk about with a bot.

When a client says, "I think you are really a person", is he saying it because that is what he believes? Or is he simply experimenting to see what kind of answer the robot will give? It is impossible to know what is in the mind of the client. This sort of problem makes it difficult to apply any objective scoring criteria to the logged conversations.

One apparently significant factor in the suspension of disbelief is whether the judge chatting with a bot knows it is a bot, or not. The judges in the Loebner contest know they are trying to "out" the robots, so they ask questions that would not normally be heard in casual conversation, such as "What does the letter M look like upside down?" or "In which room of her house is Mary standing if she is mowing the lawn?" Asking these riddles may help identify the robot, but that type of dialogue would turn off most people in online chat rooms.

Acknowledgments This research was conducted through the joint efforts of a worldwide community of dedicated free software volunteers, only a few of whom were mentioned in this manuscript. Without their help, the A.L.I.C.E. project would have been impossible. We are grateful for individual donations to the A.L.I.C.E. AI Foundation. Corporate sponsorship was provided by IDG, Franz.com, X-31, and SunlitSurf. Not one dime of government funding was expended on this research.

Erik Levy and Noel Bush edited earlier drafts of this paper. Grace Beber assisted in editing the final draft. Huma Shah proof read the final draft. Transportation by Russ Kyle, Kim Wallace printed several early drafts. I am grateful to Dr. Robert Epstein for persuading me to write this chapter.

References

Aamodt, A. and Plaza, E., 1994, Case-based reasoning: foundational issues, methodological variations, and system approaches, *Artificial Intelligence Communications* **7**(1): 39–52.

Barger, J., 1993, RACTER, posted to the comp.ai.* hierarchy in June 1993; reprinted in *The Journal of Computer Game Design*, August 1993.

Churchland, P. M. 1994, *Matter and Consciousness: A Contemporary Introduction to the Philosophy of Mind*, Revised Edition, MIT Press, Cambridge, MA.

Clinton, W., 1998, *Excerpts from Report of Bill Clinton's grand jury testimony in Washington Post.*

Dennett, D., 1991, *Consciousness Explained*, The Penguin Press, Harmondworth.

Levitt, G. M., 2000, *The Turk, Chess Automaton*, McFarland, Jefferson, NC.

Mauldin, M., 1994, Chatterbots, Tinymuds and the Turing Test: entering the loebner prize competition, *Proceedings of the Twelfth National Conference on Artificial Intelligence*, AAAI Press, Menlo Park, CA.

Miller, G. A., 1962, *Psychology: The Science of Mental Life*, Harper & Row, New York.

Norretranders, T., 1998, *The User Illusion: Cutting Consciousness down to Size*, Penguin, New York.

Penrose, R., 1989, *The Emporer's New Mind*, Oxford University Press, Oxford.

Pinker, S., 1997, *How the Mind Works*, W. W. Norton, New York.

Shannon, C. E. and Weaver, W., 1963, *Mathematical Theory of Communication*, Paperback edition, University of Illinois Press, Chicago, IL.

Sterrett, S., 2000, Turing's two tests for intelligence, *Minds and Machines* **10**(4): 541–559.

Torvalds, L. and Diamond, D., 2001, *Just for Fun: The Story of an Accidental Revolutionary*, HarperBusiness, New York.

Turing, A. M., 1950, Computing machinery and intelligence, *Mind* **59**(236): 433–460.

Weizenbaum, J., 1976, *Computer Power and Human Reason*, W.H. Freeman, San Francisco, CA.

Zdenek, S., 2000, *Stoned Machines and Very Human Humans: The Politics of Passing and Outing in the Loebner Contest*, Society for Social Studies of Science, San Diego, CA.

Zipf, G. K., 1949, *Human Behaviour and the Principle of Least-Effort*, Addison-Wesley, Cambridge, MA.

Chapter 14
The Social Embedding of Intelligence

Towards Producing a Machine that Could Pass the Turing Test

Bruce Edmonds

Abstract I claim that to pass the Turing Test over any period of extended time, it will be necessary to embed the entity into society. This chapter discusses why this is, and how it might be brought about. I start by arguing that intelligence is better characterized by tests of social interaction, especially in open-ended and extended situations. I then argue that learning is an essential component of intelligence and hence that a universal intelligence is impossible. These two arguments support the relevance of the Turing Test as a particular, but appropriate test of interactive intelligence. I look to the human case to argue that individual intelligence uses society to a considerable extent for its development. Taking a lead from the human case, I outline how a socially embedded Artificial Intelligence might be brought about in terms of four aspects: free will, emotion, empathy, and self-modeling. In each case, I try to specify what social 'hooks' might be required for the full ability to develop during a considerable period of in situ acculturation. The chapter ends by speculating what it might be like to live with the result.

Keywords Intelligence, social embedding, interaction, free will, empathy, self, emotion, Turing Test, learning, design, acculturation

Robert French (French, 1990) rightly points out that the Turing Test is a test of human social intelligence rather than of a putative 'general intelligence'. I agree. The test is an inherently social one, which will need the intelligence embedded into the society in which it is being tested (in the case of the Turing Test, human society). This chapter will discuss why this is, how this might occur for an Artificial Intelligence (AI), and how this might enable the development of some of the human-like abilities and behavior that would be necessary to pass the Turing Test. In particular, I will look at: free will, emotion, empathy, and self-modeling. Before that, I argue that there is no such thing as a general intelligence, and so the Turing Test is a very relevant test of intelligence as far as we are concerned. The reasons why a general intelligence is not possible will help motivate the following, more

Centre for Policy Modelling, Manchester Metropolitan University

constructive discussion. I end by briefly speculating on how it might be like to live with the result.

14.1 Social Tests for Intelligence

In this section I suggest that intelligence is usefully characterized by the ability to succeed with respect to one's goals in situations of extended interaction with other entities. This will lead back to the Turing Test as a special case. This is in contrast to the classic 'constrained problem solving' paradigm, where one has to find the right sequence of discrete moves to obtain a known goal. Classic examples of constrained problems used in AI include the 'blocks-world' puzzle, rectilinear mazes, the iterated prisoner's dilemma, and the traveling 'salesman problem'. Most of these puzzles have the following features in common: a sequence of discrete choices have to be made; there are a (usually small) finite number of predefined possibilities for the choices each turn; they are played off-line with respect to the world and the goals are known to all. Such 'toy' problems test for only a very restricted aspect of intelligence and almost all researchers accept that such problems are an inadequate test of general intelligence. However, such tests continue to dominate the literature – presumably, there is an assumption that these sorts of tests are a sensible starting place, en route to a more substantial intelligence. However, one suspects that researchers often use such tests because they are easy to set up and they have not thought of any alternatives.

To illustrate some of the alternatives, consider the following situation: one has a series of candidates and one wishes to select the most intelligent of them with respect to some job (we will assume the candidates are human for the moment). How might one go about this task? Some possibilities are: interview them, look at what they have achieved in the past, read what people say about them, look at their qualifications, set them a challenge, and work with them over a trial period. Unless one wants an archetypal 'egg-head', what you do *not* do is rely solely on their performance at solving a constrained puzzle. Even examinations, which come closest to the constrained problem-solving paradigm, usually require the ability to understand written text and write appropriate responses of one's own construction. For real job recruitment, people usually use a combination of evaluating past experience, taking references, and conducting an interview, but there is a general agreement that the only sure way to assess intelligence is to interactively work with the candidate over an extended period of time.

Unlike the constrained problem-solving paradigm, most situations concerning interaction between active entities allow for an indefinite number of possible actions. That is, the situation can not be meaningfully represented as a 'game' with only a finite number of possible 'moves'. For example, if I was to try and catch a particular wild fox each night (and then release it ready for the next night), and I limited myself to a few strategies, the fox would soon learn to avoid these. Rather, to be successful at this task, I would probably be forced to continually develop new approaches. This 'open-ended' characteristic of the interaction is one of the features that makes this sort of situation a more appropriate test of intelligence than that of a 'closed' game with a limited menu of choices. The necessity and type of such innovation indicates the intelligence of the fox: if a single, fixed strategy (that was potentially avoidable by the fox) worked, it would not indicate that it had much intelligence at all; if it was necessary to continually innovate in simple ways (e.g., move a trap to different positions but not change its nature), this would indicate at least some intelligence. If continual innovation in terms of the *type* of strategy was necessary, then this would indicate a substantial intelligence (e.g., a low cunning); if whatever I did, it evaded, then this might indicate that it was more intelligent than I was (in this context); if it evaded me even when I employed outside help (consulting experts, etc.), this might cause me to be so concerned as to suspect that there might be more behind this than was immediately apparent (e.g., unknown interference by another intelligent entity).

The above case is where the participants of the process have conflicting goals, resulting in a sort of cognitive 'arms race'. Conflict is not a necessary part of an interactive test of intelligence. It may be that the situation is one where the actors are cooperating. For example, in a game of 'charades', the more inventive one is at finding ways of communicating information so that one's partner will understand, the better one is at the game. Charades has an obvious parallel with the situation where one person is attempting to communicate a new idea to another – the many cycles where questions are asked and assumptions checked and the flexibility in finding ways to 'frame' the new idea with respect to ideas already known, require considerable intelligence.

The time-span of the interaction is another factor in such interactive tests for intelligence. While it is sometimes possible to succeed in such interactive situations using relatively shallow strategies over short periods of time, to succeed over in the longer term often requires increasingly sophisticated (and context-sensitive) strategies. It is relatively easy to convince someone at a party that one is something other that what one is because you would probably never meet them again. It is an altogether more difficult task to sustain such a deception if one meets them every day, which would give them a chance to test out one's replies against the world. Similarly, while relatively simple strategies might succeed in the Turing Test for a few sentences in specific and formulaic contexts, as in ELIZA (Weizenbaum, 1976), trying to maintain the illusion of humanity when the opponent can test the knowledge against others in society is an altogether tougher task (Dennett, 1984). The Turing Test is certainly an open-ended test of interactive ability and could be conducted over a suitably extended period of time, what I called the 'Long-term Turing Test' in (Edmonds, 2000). Passing the Turing Test requires the ability to successfully engage in a 'cat and mouse' cognitive arms race similar to that with the fox, as well as the sort of communicative game evident in charades. Thus, despite its specific nature, the Turing Test has a good claim to being one of the most relevant tests for the type of intelligence that most concerns us.

However, the Turing Test is not only a suitable test of interactive ability, but also one that tests for an ability to imitate humans. This makes it a particularly poignant test because the process where we might attribute intelligence to a program trying to ‘pass-off’ as a human mirrors the process by which we attribute intelligence to each other. It seems obvious that this is partly why Turing devised such a test, for if anyone denied that a machine that passed the test was not intelligent, then this would undermine their attribution of each other’s intelligence since that is made on the same basis. However, I can see no reason why the ability to imitate a human is necessary for intelligence – it is merely sufficient. However, I argue that many aspects of our intelligence are rooted in our interactive and social abilities. Thus, to succeed at social interaction will require abilities that are closely associated with human abilities (even to the extent they are frequently used to *characterize* humanity). This is like an immigrant learning to cope in a new culture. They will need to acquire many skills and much knowledge specific to the culture to be able to build a successful life, but this does not mean they have to be able to ‘pass-off’ as a native (a nearly impossible feat).

To summarize this subsection, I have built a picture of a test for intelligence rooted in interaction which is: (1) over a substantial period of time, which allows chances to test the content of interactions with the rest of the world (or society) and (2) open-ended, in that there are an indefinite number of possible interactions. The Turing Test is a specific case of this, but one which conflates the ability to succeed in complex interactive situations with the ability to ‘pass off’ as a human.

14.2 The Impossibility of a Universal Intelligence

Two paradigms have heavily influenced the thinking on intelligence in AI over the last 30 years: the universal Turing machine and the general problem solver of Ernst and Newell (Ernst and Newell, 1969).

The first showed that there was (in theory) such a thing as a universal computational device, which could be programmed to mimic any effective (symbolic) computation. In practice, this meant that one could mass produce the machines and leave their customization to the programmer. While it does seem to be true that intelligence might be implementable using computation, this is very far from intelligence being reducible to computation. The second was a system of automatically decomposing higher goals into more tractable subgoals so that one only had to specify a suitable top-level goal for the process to commence. This process of automatic subgoaling is only possible when one can deduce the appropriate subgoals from the goal and knowledge of the problem. This requires that one’s knowledge is sufficient and consistent.

Thus, the ‘universal’ Turing machine is only useful when you know the correct program, and the ‘universal’ problem-solver can only operate when given consistent information that is sufficient to eventually decompose all its goals into solvable ones. These are, at best, narrow versions of ‘universality’; in everyday life, one would almost never be able to use them without additional resources. Both systems require exactly the right amount of information so that the solution (or result) is neither under- nor over-determined (this is, of course, enforced by the method of programming in the Universal Turing Machine case). If this is not the case, these systems are simply inapplicable. Part of the problem lies in the lack of learning. In any but highly restricted and artificial problem-solving domains, learning is an integral part of intelligence. It enables a device to do something intelligent when it has either too little or too much information. Some examples of this are: when it is useful to perform an experiment upon some aspect of the environment, to gain new information to make a decision, or when one detects a contradiction (or even incoherency) in one's beliefs and one has to decide which to ignore or change.

If one accepts that learning has to be a part of any autonomously applicable intelligence, then this has consequences for the intelligence's scope. A series of formal results in machine learning have encapsulated what many in the field have suspected for some time: that there is no effective universal learning algorithm. Of these, one of the most recent is called the 'No Free Lunch' theorems (Wolpert and Macready, 1995). These show that over all possible search spaces, no search algorithm does better on average than any other. This is a highly abstract result because, most of the time, we are not concerned with spaces dominated by strange functions, such as those that are discontinuous everywhere. However, it does show that to gain any effectiveness, one has to exploit some knowledge of the search spaces that one is dealing with, even if this knowledge is minimal – for example, that they are continuous or have a finite number of maxima. An ability to learn more effectively than a random search is due to the exploitation of domain knowledge, which means that the algorithm is then limited (in terms of where it will be efficient) to the particular domain where that knowledge holds. In other words, there is no universal and effective learning algorithm – the wider the domain of application, the less efficient it might be, and the narrower the scope, the greater the potential for efficiency.

This is a deep problem and not amenable to any easy fix. For example, it may be thought that combining several different search algorithms so that the right one may be chosen by some control algorithm. *The Emotion Machine* (Minsky, 2002), suggests emotion plays this sort of role. This would then succeed in generalizing the search algorithm and that this process of generalization could be continued indefinitely. The problem with this is that it ignores the cost of deciding which algorithm to use and the impossibility, in general, of choosing the right algorithm. All one has done is to switch from searching for solutions, to searching for the right algorithm to search for solutions, which is not usually any easier than the original task.

If a universal learning algorithm is impossible and learning is an integral part of intelligence, we must conclude that intelligence is also only effective and efficient when adapted to the circumstances it is used in. It is becoming increasingly clear that our intelligence is highly adapted to our circumstances (social, physical, and biological). It does appear to be more generally applicable, in the short run, than those of other animals, especially when you consider interindividual social processes of adaptation. However, this marginal generality does not mean our abilities are completely general. The apparent generality comes at a cost; it is by no means clear that this is also the case in the long run or during abnormal events. Our socially oriented intelligence does seem to allow us to occupy a variety of ecological niches on earth by cultural rather than biological adaptation, but the same cultural abilities are also our weakness and could easily end up destroying us as a species.

One suspects that the underlying reason why it has been assumed that a universal intelligence is possible is anthropocentrism: humans like to assume that they are in possession of such an intelligence. The assumption seems to be that in all important aspects, our cognitive abilities are universally applicable and that any limitations (typically put down to 'mere' memory and processing capacity limitations) can be overcome, e.g., by the use of tools such as: writing or computers. This strikes me as pure hubris: not long ago, we assumed we were physically in the Centre of the universe; we now seem to be making a similar sort of assumption in terms of our cognitive abilities.

Robert French drew an analogy between trying to imitate the flight of a seagull (for an observer watching on radar) and trying to pass the Turing Test. He suggests that just as the flight of a seagull is only a particular type of flight, the Turing Test tests for only a particular type of intelligence – human social intelligence. However, this merely builds in the assumption that intelligence *can be* meaningfully generalized by choosing a situation where we happen to know there is an appropriate generalization for us – flight. The analogy has apparent force by conflating abstraction and generalization. Flight can be both generalized and abstracted from the seagull's example, and intelligence can be abstracted from the Turing Test, but this does not follow that it is possible to meaningfully generalize from that which is tested by the Turing Test to a 'universal intelligence'.

14.3 The Social Construction of Individual Human Intelligence

If one accepts that the ability to choose or construct actions that further one's goals in complex, extended, open-ended, and social situations, is an appropriate characterization of intelligence, then this will have consequences for how such an intelligence can be brought about. The human case provides some clues as to how this might happen. It seems that human intelligence is heavily dependent upon society for its development. In this section, I look at some of the evidence and arguments for this including: language, the ability to participate in the social web of cooperation and competition, the presence of exploitable information resources in society and the phenomena of autism.

A major component of human intelligence is language. Without language, humans are not able to coordinate action effectively to achieve goals; without language, humans could not discuss problems to brainstorm, criticize, and compare solutions; and without language we would not be able to reason so effectively. For example, it is almost inconceivable that a human without mastery of a sophisticated language could perform abstract mathematics. Full languages are not preprogrammed into our brains by our genes. To acquire a full language, it is necessary for an individual to be socially immersed in a linguistic environment. The ability to make use of such an environment to acquire language is genetic, but the language itself is largely learnt in situ. The way our linguistic intelligence is developed suggests a model for the development of social intelligence in general. That is, the individual is preprogrammed with certain abilities, biases, and innate knowledge. These ‘hooks’ then allow the rapid learning of the contingent knowledge and skills through interaction in the society. This approach allows for the contingent information to be adapted to the circumstances of the society.

The recently proposed ‘social intelligence’ (Kummer et al., 1997) and ‘Machiavellian intelligence’ (Byrne and Whiten, 1988, 1997) theses put forward the theory that substantial aspects of our intelligence evolved because its possession conferred social advantage. The idea is that our extensive intelligence primarily evolved to keep our place in the social order and to manage the intricate web of cooperation and competition that this involves. The idea is that the evolutionary advantage our intelligence comes from our ability to develop different cultures suited to different niches and to socially organize to exploit these.

If this is the case (and it would be odd if none of our intelligent capacity has been shaped by evolutionary pressures that are socially grounded), and given the intricacy of our present society (which presupposes the possession of individual intelligence), then it seems likely that our intelligence and our society have coevolved. If this is the case, then one would expect that substantial aspects of our intelligence have evolved to ‘fit in’ with our society (and vice versa). It is certainly difficult to argue from single cases, but the fact that the only species to evolve a sophisticated intelligence has also evolved a sophisticated society cannot be totally ignored.

One aspect of a society of roughly commensurate social entities, that is almost inevitable, is that it will quickly develop so as to be more complex than any single entity can completely model. This is especially true of a society where there is sometimes advantage in ‘out-guessing’ the actions of the other entities, in which case a sort of cognitive ‘arms-race’ can develop which, in its turn, makes the prediction and comprehension of the society even more difficult.

Given that our society will be more complex than we can ever understand, there will probably be societal processes that perform computations that we cannot do individually. It would be strange if some of these informational and computational resources (i.e., the ‘results’ of the societal computation) were not accessible to some of the constituent entities some of the time. Given this availability, it would also be odd if these resources were not exploitable by the entities it is composed of. Hence, one would expect entities that were situated in such a society to evolve ways of exploiting such resources. Indeed, some simulations do suggest this (Edmonds, 1999a). If this were the case, then we would expect that we would posses some faculties, usually attributed to our ‘intelligence’, that were evolved to use such resources and save ourselves (individually) considerable time and effort. This use of information resources in society is analogous to the sampling of the immediate physical environment for information about our position, etc. rather than relying for the detail from an internal ‘map’ (Brooks, 1991). Thus, one would expect that our brain has not only evolved because it allows the creation of culture, but also that it would have evolved to exploit this culture.

One piece of evidence about the importance of society to individual intelligence is the phenomenon of autism. Since the 1940s, autism has been known as a syndrome which involves, among other features, the striking lack of social competence. A variety of explanations have been discussed, among them, the widely discussed 'theory of mind' model which conceives autism as a cognitive disorder (Baron-Cohen, Leslie et al., 1985), and, a more recent explanation given by (Hendriks-Jansen, 1997). He hypothesizes early developmental disorders as the primary cause, which prevents the child and its caretakers from getting 'the interaction dynamics right', the interaction dynamics which normally scaffold the development of appropriate social interactions, in the sense of situated dialogues between infant and caretaker.

The importance of interaction dynamics are also part of the explanation given in (Dautenhahn, 1997), which suggests a lack of empathic processes that prevent the child from developing 'normal' social action and interaction. People with autism rarely develop into 'normal' social beings. Although some of them show high intelligence in nonsocial domains, they are seldom able to communicate and interact successfully with other people. They are not able to understand the social world around them, which therefore appears as scary and unpredictable. This deficit influences their lives to the extent that they, often, are not able to lead an independent life, in this way clearly demonstrating the central role of sociality in practical intelligence. This gives evidence that socially situated aspects of intelligence do not merely provide an important add-on to other faculties of intelligence (like spatial thinking or mathematical reasoning), but that human intelligence (its development and expression) is embedded (and embodied) in a social being, and thus cannot be separated from nonsocial kinds of intelligence. The existence of autism shows two things: that there are different types of intelligence (e.g., mathematical and social); and that social intelligence is critical to survival in humans.

It often seems to be assumed that social intelligence is merely the application of general intelligence to the case of social interaction. However, the arguments and evidence above tend to indicate that there is another possibility – that a large part of our intelligence is to facilitate and exploit the social processes that give our species evolutionary advantage, and that our ability to use our intelligence for other ends is partly an offshoot of our social intelligence. In other words, one of the main aspects in which our intelligence has specialized is that of managing our social interaction.

The development (and to a lesser extent the application) of our individual intelligence is thus dependent upon its development in a social context. This requires a considerable period of in situ training and development and is in sharp contrast to the way in which we tend to develop AIs. In AI, the analytic design stance predominates – the required ability is analyzed and then explicitly designed into a set of programs and hardware. If we value intelligence that is commensurate with human intelligence in terms of its social ability, then there is a real possibility that such a stance might be inadequate to the task.

14.4 Towards Producing a Socially Embedded Intelligence

Passing the Turing Test (and especially the long-term Turing Test), requires that the entity is effectively embedded in human society, because this is the only way in which an entity can have access to the wealth of contingent and context-dependent information that is necessary to pass-off as human. As in the analogous situation of the immigrant trying to assimilate into a new country, preparation before-hand by reading books about the country would not be sufficient to pass themselves off as a native – it takes decades of immersion before this is possible. It is easier to learn to merely interact successfully in that culture, a process that would probably take years rather than decades, but this requires that one has already learned to act successfully in a human culture. It takes a child decades before it is fully competent to interact in its first culture, and it is only because the knowledge and skills gained is transferable to a different human culture that a later, relatively rapid, viable application to a new culture is possible.

In this subsection, I consider four aspects of human intelligence that are probably necessary for interactive intelligence. Each requires the social embedding of intelligence, albeit in slightly different ways. The aspects are: free will, appropriate emotional responses, empathy and self-modeling. These are not sufficient to pass the Turing Test (or similar interactive test) but are probably necessary.

In each case, the strategy is the same: provide the cognitive processes and 'hooks' that would allow the development of the aspect during considerable in situ acculturation. I speculate about how the ability might arise, and hence, what 'hooks' and social environment might be necessary for their development. Although in each case some aspects of these mechanisms have been explored by me and others, these suggestions are highly speculative. However, it does appear to me that these abilities are deeply rooted in social processes for their fruition, so I would expect that some comparable set-up and process would do the job if these suggestions prove inadequate.

It must be said that very little is known about what kinds of abilities are necessary for, and what kind of interactions result. The distributed AI and social simulation communities have made a start, but there is still much to do. Although I do try to base the suggestions I make on what I observe or theorize is necessary for these abilities to develop, they have only been tested in rudimentary and fragmentary ways. There are many other important aspects of human intelligence that are, no doubt, necessary, that I have not considered here, including: natural language, context-awareness, and analogical thinking.

14.4.1 Free Will

I am not going to discuss the philosophical aspects of free will here, but rather the aspects that are practically relevant to social interaction, and how these aspects might be brought about. From this perspective, an entity with free will has the following key properties:

- *First*, that some of its actions are not predictable before hand – a part of this is that given similar circumstances and history the entity may behave in different ways.
- *Second*, that the action is rational, by which I mean that it furthers the entity's goals – sometimes it is also necessary that this rationality is socially apparent (or can be socially demonstrated).

These are largely a question of effective power. In competitive situations, it is often to an entity's advantage to be somewhat unpredictable, so that one's competitors will not be able to completely predict what you will do, even if they have frequently observed you in similar circumstances. On the other hand, it is advantageous that one's actions further one's goals. Clearly these are not easy requirements to simultaneously satisfy.

Another aspect of personal power is that it is often dependent on being a member of social groups, institutions, and processes, and this membership is often dependent on being able to demonstrate that one is rational (with respect to acceptable goals). This reflects the fact that these institutions, etc. depend on constraints and inducements being effective on its members' actions. If a member is not rational, then there will be no way for the institution to promote or enforce these constraints upon that member and the inducements may be ineffective. If one fails to convince others of one's rationality, one can lose social advantage due to one's action being anticipated and countered, or to one being ineligible to participate in the social institutions that give access to socially produced goods (in the widest sense).

In the Turing Test, one of the ways in which one might judge whether the entity was human or not is whether its responses to the same conversational sequence results in an unpredictable reply that is, afterwards, understandable in terms of imputable goals.

The basic idea I am proposing, is to provide, in a constructed brain, an environment which is conducive to the *evolution* of free will inside that brain. In this evolutionary process, practical indeterminacy emerges first in undetectable amounts and then develops into full-blown free will by degrees. This evolution happens in parallel to the development of rationality in the individuality, so that the result is a will which is internally coherent in furthering its goals, yet not effectively predictable from its circumstances.

Those who insist that free will requires prior free will (arguing that otherwise the *choice process* would also be effectively determined and hence themselves predictable) can follow the chain of causation backwards until it slowly diminishes down to the limit. In this model, the gradual emergence of free will is analogous to the emergence of life – it can start from infinitesimal amounts and evolve up from there. This requires that practical free will comes in different degrees – in other words that circumstances can constrain behavior to different extents: from totally to partially (some degree of free will). The artificiality of an all-or-nothing division into having it or not makes as little sense with practical free will as it does with life (as exhibited by actual organisms, as in viruses or deep frozen bacteria). This is especially important when one is discussing mechanisms for its appearance (as must occur somewhere between the newly fertilized embryo and the adult human). As Douglas Hofstadter (1985) said:

> Perhaps the problem is the seeming need that people have of making black-and-white cut-offs when it comes to certain mysterious phenomena, such as life and consciousness. People seem to want there to be an absolute threshold between the living and the nonliving, and between the thinking and the 'merely mechanical…'

Thus, a situation where free will evolves in increasing effectiveness during development gets around the requirement for prior free will. Not only can the actions be free, but also the deliberation that resulted in those actions and the process to develop those deliberations, etc. The fact that the chain of free will disappears back into the internal evolutionary process can be expressed as a closure property.

The selective advantage that this feature confers is primarily that of external unpredictability (combined with an internal rationality). That is, in a competitive environment, if an opponent can predict what you will do, then that opponent would have a distinct advantage over you. Such competition in a social setting fits in well with the social intelligence hypotheses mentioned above (Byrne and Whiten, 1988, 1997), since unpredictability can give social advantage and hence be selected for by evolution. That such an effective unpredictability can be evolved has been shown by Jannink (1994). He developed an algorithm where two separate populations were coevolved. The first of these populations was allocated fitness on the basis of the extent to which its programs successfully predicted the output of programs from the second, and individuals from the second were allocated fitness to the extent that it avoided being predicted by individuals from the first population. Here, the two populations are involved in a basic evolutionary 'arms-race'.

The basic architecture I am suggesting is composed of the following elements:

- An expressive framework for expressing strategy-forming processes
- A population of such processes within this framework
- A way to construct new processes as a result of the action of existing decision-making processes and the knowledge of the entity
- A selection mechanism that acts to (1) select for those processes that tend to further the individual's goals and (2) to select against those processes that are predictable by others

This evolutionary architecture is the basis for the suggested implementation. However, this architecture needs several more features to realize its potential, discussed in the following sections.

14.4.1.1 Open-ended Strategy Evolution

In a standard Genetic Algorithm (Holland, 1992), the genome is a fixed length string composed of symbols taken from a finite alphabet. Such a genome can encode only a finite number of strategies. This finiteness imposes a ceiling upon the possible elaboration of strategy. This can be important where individuals are involved in the sort of modeling 'arms-race' that can occur in situations of social competition, where the whole panoply of social maneuvers is possible: alliances, bluff, double-crossing, lies, flattery, etc. The presence of a complexity ceiling in such a situation (as would happen with a genetic algorithm) can change the outcomes in a qualitatively significant way, for example, by allowing the existence of a unique optimal strategy that can be discovered.

This sort of ceiling can be avoided using an open-ended genome structure as happens in Genetic Programming (Koza, 1992, 1994) or messy genetic algorithms (Goldberg, Deb et al., 1989). Within these frameworks, strategies can be indefinitely elaborated so that is it possible that any particular strategy can be improved with sufficient ingenuity. Here, I use the genetic programming paradigm, since it provides a sufficiently flexible framework for the purpose in hand. It is based on a tree-structure expressive enough to encode almost any structure, including neural-networks, Turing complete finite automata, and computer programs (Koza, 1992, 1994). Using the genetic programming paradigm means that only the available computational resources limit the space of possible strategies. It also has other properties which make it suitable for this purpose:

- The process is a path-dependent one since the development of new strategies depends on the resource of present strategies, providing a continuity of development. This means that not only can completely different styles of strategy be developed, but also different ways of approaching (expressing) strategies with similar outcomes.
- The population provides an implicit sensitivity to the context of action – different strategies will 'surface' at different times as their internal fitnesses change with the entities' circumstances. They will probably remain in the population for a while even when they are not the fittest, so that they can 're-emerge' when they become appropriate again. Thus, entities using an evolutionary decision-making algorithm can appear to 'flip' rapidly between strategies as circumstances make this appropriate.

14.4.1.2 Meta-evolution

Such a set-up means that the strategy that is selected by an entity is very unpredictable; what the currently selected strategy is can depend on the history of the whole population of strategies, due to the result of crossover in shuffling sections of the strategies around and the contingency of the evaluation of strategies depending on the past circumstances of the entity. However, the method by which new strategies are produced is not dependent on the past populations of strategies, so there is no backward recursion of the choice property, where the presence of free choice at one stage can be 'amplified' in the next.

Thus, my next suggestion is to include the operators of variation in the evolutionary process. There are only two operators in the Koza's original genetic programming

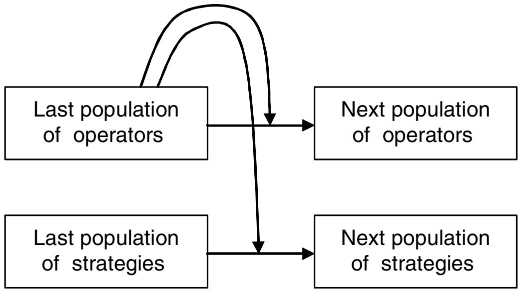

Fig. 14.1 One step of a meta-genetic evolutionary process

algorithm: propagation and tree-crossover. Instead of using only these operators, I suggest that there be a whole population of operators that are themselves specified as trees following (Edmonds, 2001). These operators can be computationally interpreted so they *act* upon strategies in the base population to produce new variations (instead of crossover). The operators are allocated fitness indirectly from the fitnesses of the strategies they produce using the 'bucket-brigade' algorithm of Holland (Holland, 1992) or similar, such as that of (Baum, 1998), which is better motivated.

To complete the architecture, we arrange it so that the population of operators also operates on their own population, to drive the production of new operators. The decision-making processes in this architecture (including the processes to produce the processes, etc.) are generated internally, in response to the twin pressures of deciding what to do to further the entities' goals (in this case profit) and avoiding being predictable to other entities. This is illustrated in Fig. 14.1.

14.4.1.3 Anticipatory Rationality

If an entity is to reflectively choose its action rather than merely react to events, then this entity needs to be able to anticipate the result of its actions. This, in turn, requires some models of the world, i.e., some representation of the consequences of actions that has been learned through past interaction with that world (either via evolution of the entity or a cognitive process). This has been called 'anticipation', and was first noticed by (Tolman, 1932), suggested as part of the schema in (Drescher, 1991), and included in evolutionary systems in (Stolzmann, 1998).

The models of the consequences of action are necessarily separate from the strategies (or plans) for action. It is possible to conflate these in simple cases of decision-making but if an entity is to choose between plans of action with respect to the expected outcome, then this is not possible. There is something about rationality which seems to limit the meta-strategy of altering one's model of the world to suit ones chosen strategy – the models are chosen according to their accuracy in anticipating the effect of action (as well as their relevance), and the strategies are then chosen according to which would produce the best anticipated outcome

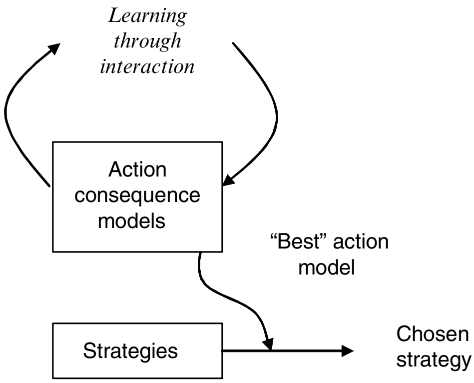

Fig. 14.2 Using anticipation with strategy selection

according to the previously selected model. This is in contrast to a purely reactive entity that may work on the presumption that the strategies that have worked best in the past are the ones to use again. A reactive strategy excludes the possibility of anticipating change or being able to deliberately 'break-out' of current trends and patterns of behavior. Thus, what is proposed is a process that models the consequences of, and strategies for action. To decide on an action, the best relevant model of action consequence is chosen, and the various strategies for action considered with respect to what their anticipated consequences would be if the action consequence model is correct. The strategy that would seem to lead to the consequence that best fitted the goals is chosen. This is illustrated in Fig. 14.2.

14.4.1.4 Coevolution

The next important step is to situate the above set-up in a society of competitive peers. The development of free will only makes sense in such a setting, for if there are no other active entities who might be predicting your action, there would be no need for anything other than a reactive cognition.

Thus we have a situation where many entities are each evolving their models of their world (including of each other) as well as their strategies. The language that these strategies are expressed in needs to be sufficiently expressive so that it includes strategies such as: attempting to predict another's action and doing the opposite, evaluating the success of other entities and copying the actions of the one that did best, and detecting when another entity is copying one's own actions and using this fact. Thus the language has to have 'hooks' that refer to ones own actions as well as to others' past actions and their results. In circumstances such as these, it has been observed that entities can spontaneously differentiate themselves by specializing in different styles of strategies (Edmonds, 1999b). It is also not the case that these entities ignore each other just because they are competing. Such a coevolution of strategy (when open-ended and resource limited) can result in the intensive use of the actions of others as inputs to their own deliberation, but in a way that is unpredictable to the others (Edmonds, 1999a). So the suggested structure for entity free will can include a high level of social embedding.

14.4.1.5 Structuring the Development of Free Will Within a Society of Peers

The final difficulty is to find how to structure this mental evolution so that in addition to maintaining the internal coherence of the deliberations and their effectiveness at pursuing goals and being unpredictable to others, the actions of the entity can be presented to others as rational and verified as such after the case by those entities.

This can be achieved if there is a normative process that specifies a framework of rationality that is not overly restrictive, so that different deliberative processes for the same action can be simultaneously acceptable. The framework must be loose enough so that the openness of the strategy development process is maintained, allowing creativity in the development of strategies, etc. On the other hand, it must be restrictive enough so that others can understand and empathize with the deliberative processes (or at least a credible reconstruction of the processes) that lead to an action. There are a number of ways in which this framework could be implemented. I favor the possibility that it is the *language* of the strategies which is developed normatively in parallel with the development of an independent free will. Thus, the bias of the strategies can be coevolved with the biases of others and the strategies developed within this bias.

14.4.1.6 Putting it all Together

Collecting all these elements together, we have the following parts:

1. A framework for the expression of strategies which is (at least partially) normatively specified by the society of the entity.
2. An internal open-ended evolutionary process for the development of strategies under the twin selective pressures of favoring those that further the goals of the entity and against those that result in actions predictable by its peers.
3. That the operators of the evolutionary process are coevolved along with the population of strategies so that indeterminism in the choice of the entity is amplified in succeeding choices.
4. That models of the consequences of action be learned in parallel so that the consequences of candidate strategies can be evaluated for their anticipated effect with respect to the entity's goals.

Each of these elements have already been implemented in separate systems, all that is left is that these be put together. No doubt doing this will reveal further issues and problems to be solved; however, doing so will represent, I suggest, real progress towards the goal of bringing about a practical free will.

14.4.2 Emotion

Human emotion is a complex phenomenon (Sloman, 2002). It is partly innate and partly learned; it seems to have a pervasive effect on almost all aspects of cognition, including perception and action; it is associated with physiological effects, interpersonal signaling and phenomenal experience; it is affected by our thoughts, our environment and the emotions of others; some of our emotions seem to be shared with animals.

Even if, as some have suggested, emotion is grounded in our physiology and the extent to which we achieve our goals (Rolls, 2000), it then acquires layers of extra function through its social use. For example, it may be that anger has a basic role in triggering physiological changes to prepare us for action (e.g., fight) when we are threatened. However, these physiological changes are detectable by others, who may then decide to avoid a fight by backing down. Given this backing down, becoming angry might be an effective way of getting one's way without fighting, so one might learn to trigger anger in oneself in the absence of any threat. In another iteration of this cycle: if others observe one is using anger in this way they might learn to automatically meet anger with anger in situations where a fight is unlikely to make it clear that the person will not simply get his own way by being angry. Thus anger can be more important as a social signaling device than as a trigger for physiological change. It seems likely that it is by this sort of process that different cultures have developed different patterns of emotional response and display. Once brought up in one culture, it is not possible to easily change one's emotional habits and expectations.

Here, the idea is that in practice, emotions are a special type of action – an action whose effect is on one's own body and brain. An essential part of the emotion-action is that it affects the way one considers and evaluates one's own models and strategies. Thus, fear might release chemicals to facilitate our body for detection of danger and flight but it will also cause us to evaluate our ideas and perceptions pessimistically. These emotion-actions will be triggered in a similar way to other actions – sometimes it will be a simple and unconscious reaction (as might happen when we are woken by a loud and unexpected noise), and sometimes as a result of conscious deliberation (e.g., when we imagine what a teacher might say to us when it is revealed we have not done our homework).

Thus, in this proposal, the emotion acts upon the mechanism that evaluates models or strategies, and this, in turn, effects the model or strategy that is selected as the best (i.e., most appropriate under the circumstances). Thus, the emotion does not determine the action but biases the selection of an action. Emotion-actions can be seen as a 'short-cut' concerning the feedback to action selection that has been learned over the course of evolution so that we do not have to learn it individually. If the strategy learning process was implemented with an evolutionary algorithm, the emotion would change the evaluation of candidate strategies in the population of possible strategies being developed. This set-up is illustrated in Fig. 14.3.

Clearly, the emotion-actions we have evolved relate to different situations and needs where the emotion will give adaptive advantage: fear to a threat, protectiveness to child-rearing, etc. Many of these are as appropriate for an artificial entity as for

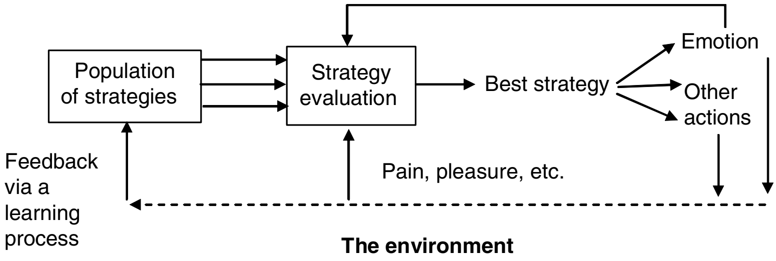

Fig. 14.3 Emotion acting to affect the evaluation of models and strategies

humans. However, these emotion-actions are elaborated and moderated in different ways for different circumstances. Some emotions will be moderated to suit the environment that the society inhabits. For example, anger may be suppressed or redirected in societies where a high degree of coordination is necessary and where there is no significant physical threat from others. Other emotions may be moderated by fashion, e.g., crying in men.

Putting emotions into the learning loop of the entity, so that it biases the learning and evaluation of possible actions, allows it to be elaborated and moderated as a result of learning by the individual. Then it becomes possible to learn to choose an emotion because it is useful to do so, just like any other action. One can also (with more difficulty) learn to suppress an emotion. For example, one can decide not to be angry (or develop one's anger). Of course, just as with other actions, we can be surprised and take an action before we have thought about it, but this does not mean that such actions are always instinctual or reactive.

Sharing the same emotional machinery means that we know quite a lot about the likely emotions in others. This knowledge is only reliable for prediction in extreme cases, or when it will be difficult for the person to reflect (e.g., when woken in the night or drunk). At other times, we know that the emotions can be considerably elaborated or moderated by culture and the individual, in which case it is no more predictable than other actions. However, when the emotion does occur, we can understand and empathize with it – this is the subject of the next subsection.

14.4.3 Empathy and Self-modeling

Human empathy involves being able to share the emotions of others. This seems to include (at least) the following steps:

1. *Understanding* the emotions of another from their actions and situation
2. *Imagining* oneself experiencing the emotions of another based upon our understanding
3. *Experiencing* the emotions of another based on our imagination of them

Other aspects of empathy might include: invoking one's own emotions in others and detecting whether other's emotions are real or feigned. Although empathy can be a powerful means of expressing sympathy, it goes beyond sympathy, for sympathy does not (necessarily) include the duplication of other's emotions.

Each of the three steps listed above requires different cognitive abilities. Understanding another's emotions (step 1) requires that one has an adequate model of other people's emotions; imagining their emotions (step 2) requires that one is able to map one's model of other's emotions onto one's own; and experiencing them (step 3) requires that this mapping can trigger one's similar emotions in oneself. Of course, empathy can work both ways – once one starts experiencing the emotions of another (e.g., fear) this might reinforce the emotion in the person it originated in (or it may invoke it in a third person).

The ability to trigger similar emotions to someone else's in oneself implies a high degree of commonality between people. Since some of this commonality is culture-specific, it must be learned during development. Here, I suggest a process where at an early stage, an entity codevelops its self-model alongside its model of others. Later, a differentiated self-model might be developed from this common base to suit the needs and experience of the individual.

I outline a model of how the self-model might be developed. This attempts to reconcile the following observations and theories about the human sense of self:

1. The self is only experienced indirectly (Gopnik, 1993).
2. The self requires a strong form of self-reference (Perlis, 1997).
3. The aspects of the self are socially constructed (Burns and Engdahl, 1998).
4. 'Recursive processing results from monitoring one's own speech' (Bridgeman, 1992).
5. One has a 'narrative Centre' (Dennett, 1989).
6. There is a 'Language of Thought' (Aydede, 1999) to the extent that high-level operations on the syntax of linguistic production, in effect, cause other actions.

The purpose of this is to approach how we might provide the facilities for an entity to construct its self-model using social reflection via language use. If the entity's self-model is socially reflective, this allows for a deep underlying commonality to exist without this needing to be prescribed beforehand. This way, the nature of the self-model can be developed in a flexible manner and yet be this structural commonality allowing empathy between its members.

This model is as follows (and illustrated in Fig. 14.4):

1. There is a basic decision-making process that acts upon the perceptions, actions, and memories of the entity and returns decisions about new actions (this would include changing the focus of one's perception and retrieving memories).
2. The entity attempts to model its environment by a learning process that does not have direct access to the workings of this basic process but only of its perceptions and actions, past and present.

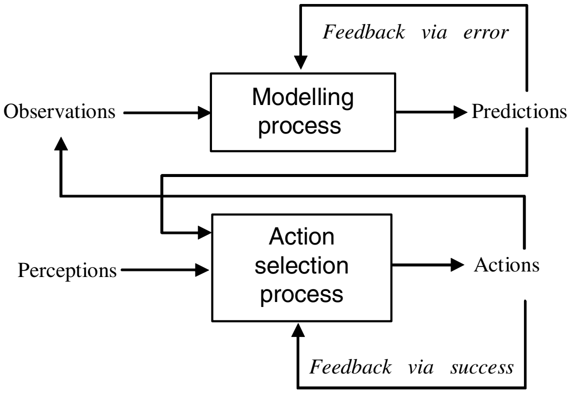

Fig. 14.4 Separate modeling and decision-making processes feeding back into each other

3. The modeling process seeks to model its environment, including the other entities it can interact with. It also attempts to model the consequences of its actions (including speech acts).
4. The modeling process picks up and tries selections of the communications it receives from other entities and uses these as a basis (along with observed actions) for modeling the decisions of these entities.
5. The action selection mechanism becomes adapt at using communication acts to fulfill its own needs via others' actions, using its model of their decision-making processes (using the results of the modeling process in terms of predictions of their actions).
6. Using the utterances it produces, it learns to model itself (i.e., to predict the decisions it will make) by applying its models of other entities to itself, by comparing its own and others' acts (including communicative acts). The richness of the language allows a relatively fine-grained transference of other's decision-making processes onto itself.
7. It refines its model of other entities' actions using its self-model and its self-model from its observation of other's actions. Thus, its model of other's and its own cognition coevolve.
8. Since the model of its own decisions is made through language, it uses language to implement a sort of high-level decision-making process – this appears as a language of thought.

The key points are that the basic decision-making process is not directly experienced. The entity models others decision-making using their utterances as fine-grained indications of their mental states (including intentions, etc.), and then models its own action selection mechanism by applying its model of others to itself (and vice versa). This seems to be broadly compatible with observations in (Aydede and Güzeldere, 2002).

14.4.3.1 General Consequences of this Set-up

The important consequences of this set-up are:

- The fact that models of other entities and self-models are codeveloped means that basic assumptions about self and other's cognition are similar.
- The fact that an expressive language has allowed the modeling of others and then of its self means that there is a deep association of self-models with this language.
- Communication has several sorts of use: as a direct action intended to accomplish some goal, as an indication of another's mental state/process, as an indication of one's own mental state/process, as an action designed to change another's mental state/process, as an action designed to change one's own mental state/process, etc.
- Although the modeling processes do not have access to the basic decision-making processes, they do have access to and can report on their self-model, which is a model of their decision-making. Thus, they do have a reportable language of thought, but one which is only a good approximation to the underlying basic decision-making process.
- The model allows social and self reflective thinking without regression limited only by computational resources and ingenuity – there is not a problem with unlimited regression, since there is no direct access between the modeling process and the action selection process.

14.4.3.2 Toward Producing Self-modeling Entities

Working from the above model gives enough information to suggest the outline of a set-up which might bring about self-modeling. The basic requirements for this are:

1. A suitable social environment (including humans)
2. A sufficiently rich communicative ability (i.e., a communicative language that allows the fine-grained modeling of others' states leading to action in that language)
3. A general anticipatory modeling capability
4. An ability to distinguish the experience of different types, including the observation of the actions of others, ones own actions, and other sensations
5. A need to predict other's decisions
6. A need to predict one's own decisions
7. The ability to reuse model structures learned for one purpose for another

Some of these are requirements on the internal architecture of an entity, and some on the society it develops in. I will briefly outline a possibility for each.

The entity will need to develop two sets of models.

1. A set of models that anticipate the results of action, including communicative actions (this roughly corresponds to a model of the world). Each model would be composed of several parts:

- A condition for the action
- The nature of the action
- The anticipated effect of the action
- (Possibly) its past endorsements as to its past reliability

2. A set of models of strategies for obtaining its goals (this roughly corresponding to plans); each strategy would also be composed of several parts:

   - The goal
   - The sequence of actions, including branches dependent on outcomes, loops, etc.
   - (Possibly) its past endorsements as to its past success

These can be developed using a combination of anticipatory learning theory (Hoffman, 1993) as reported in (Stolzmann, 1998) and evolutionary computation techniques. Thus, rather than a process of inferring subgoals, plans, etc., they would be constructively learned (similar to that in Drescher, 1991). The language of these models needs to be expressive, so that an open-ended model structure, such as in genetic programming (Koza, 1992), is appropriate, with primitives to cover all appropriate actions and observations. However, direct self-reference in the language is not built-in. The language of communication needs to be a combinatorial one, generated by the internal language and also deconstructed by the same.

The social situation of the entity needs to have a combination of complex cooperative and competitive pressures in it. The cooperation is necessary if communication is at all to be developed, and the competitive element is necessary for it to be necessary to be able to predict other's actions (Kummer et al., 1997). The complexity of the cooperative and competitive encourages the prediction of one's own decisions. A suitable environment is where, to gain substantial reward, cooperation is necessary, but that inter-group competition occurs as well as competition for the dividing up of the rewards that are gained by a cooperative group.

Some of the elements of this model have already been implemented in pilot systems (Drescher, 1991; Edmonds, 1999a; Stolzmann, 1998).

14.4.3.3 Back to Empathy

If one has a situation where entities' self-models have substantial commonality with each other and emotions are, to an extent, elaborated and moderated by culture-specific learning, then empathy is possible. The entity could *understand* others' emotions due to the commonality of the basic emotion-action's purpose and effect as well as the fact that it has extensively modeled the emotions of others. It could *imagine* these emotions because its model of its own emotions was codeveloped with its model of other's emotions. Finally, it could *experience* these emotions having learned to trigger the corresponding emotions in it in the appropriate social circumstances.

14.5 Living with the Result

In considering what it will be like to live with the result, I will consider two scenarios: the first is the situation where intelligences that can successfully interact in complex, social situations are developed; the second is the case where such intelligences were able pass the Turing Test (or similar tests). The basis of this speculation is an analogy between different intelligences interacting in an information 'ecology', and species interacting in a biological ecology.

In the first case, there would be new intelligences that had the capability of interacting fully with each other and with humans, but are not necessarily similar enough to humans so that they could pass the Turing Test. At the moment, the information ecosystem is dominated by a single type of intelligence: human intelligence. In this future, there might be many 'species' of intelligences dealing with information. In a biological ecosystem, competition for resources can result in the specialization of species so as to avoid direct competition. This is possible as long as there is a diversity of niches in the ecosystem and the species are sufficiently different. In this case, each species will have comparative advantage over other species in different circumstances, so that no one species dominates everywhere. If there is no such thing as a 'general species', then species are wiped out only if a resource that they rely upon is used up or destroyed by another. If, as I have argued, a general intelligence is impossible and there are many essentially different ways in which one can exploit information, then the creation of new intelligences might not lead to our eclipse, but it might well mean that we have to specialize in those areas where our kind of intelligence gives us comparative advantage. This would mean that we would not be competitive in certain arenas, in the same way that most of us could not compete with professional musicians or sumo wrestlers. The difference would be that we could not pretend to ourselves that we had the potential to compete – the limitations of our intelligence would be thrown into sharp relief, by entities that are not human. When another human succeeds in a way we can not, we can gain comfort by imagining ourselves in their shoes or mentally associating ourselves with them and hence 'sharing' their success. With an artificial intelligent entity, this may be more difficult and so their relative success might be more difficult to bear. Historically, people have resented most the newcomers that are least similar to themselves.

Sometimes the emergence of a new species creates new niches for other species (e.g., mollusk shells used by hermit crabs) – or even opportunities for symbiosis. In a similar way, it may be that new intelligences create new opportunities for human intelligence in ways that we cannot foresee.

If new intelligences are brought about that are sufficiently close to human intelligence to pass the Turing Test, they may be much more difficult to live with. The closer the intelligences are to human intelligence, the more directly they will compete with humans and the less likely it is that the new and human intelligences will be able to specialize to exploit different types of interactive opportunity or informational resource. This may mean that the informational niches that humans are left with are relatively restricted and unimportant. In the UK, grey squirrels and red squirrels inhabit the same niche, so that it is inevitable that (in the absence of outside intervention) that one species will be wiped out (as has occurred, except in a few local areas where the grey squirrels are culled by humans to protect the red).

On the other hand, if they are similar to us (as in some way they must be to pass the Turing Test), maybe our demise will be softened by their similarity with us. We can pass away with the comforting thought that our image lives on in them (until they reinvent it in their own society).

14.6 Conclusion

The Turing Test happens to be quite a good test for interactive intelligence of a particular type – far better than any constrained puzzle or 'toy problem'. It does not (necessarily) demand the physical embodiment of the entity but it does necessitate its social embedding. Several key aspects of human intelligence have their roots in this social embedding, including (I argue): free will, emotion, empathy and self-modeling. To approach passing such a test, it will be necessary for an intelligence to have these aspects, and hence will need to be socially embedded in the society in which it will be tested. The only effective way of doing this is by providing the entity with the necessary preprogramming or 'hooks' and then develop the intelligence with a considerable period of in situ training.

Another aspect of the Turing Test is the ability to 'pass-off' as human. This would require the intelligence to be very close to human intelligence, both in abilities and limitations. Given the wealth of contingent and context-dependent knowledge that is necessary to do this, this seems unlikely. However, if such an intelligence was achieved, it would compete directly with us, leaving us almost no area in which we could successfully dominate.

A modification of the Turing Test where the emphasis is on successful interaction rather than 'passing-off' as human would be far more appropriate. It would be a much more feasible target than the full Turing Test and would result in intelligences that are far more pleasant to live with.

References

Aydede, M., 1999, Language of thought hypothesis: state of the art, in: *Stanford Encyclopedia of Philosophy*; http://plato.stanford.edu/entries/language-thought/.

Aydede, M. and Güzeldere, G., 2002, Consciousness, intentionality, and intelligence: some foundational issues for artificial intelligence; http://web.clas.ufl.edu/users/maydede/introspection.htm.

Baron-Cohen, S., Leslie, A.M., et al., 1985, Does the autistic child have a 'theory of mind'? *Cognition* **21**: 37–46.

Baum, E., 1998, Manifesto for an Evolutionary Economics of Intelligence, in: B., *Neural Networks and Machine Learning*, Springer, pp. 285–344.

Bridgeman, B., 1992, On the evolution of consciousness and language, *Pscoloquy* 3; http://www.cogsci.soton.ac.uk/cgi/psyc/newpsy?3.15.

Brooks, R. A., 1991, Intelligence without representation, *Artificial Intelligence* **47**(1–3): 139–159.

Burns, T. R. and Engdahl, E., 1998, The social construction of consciousness part 2: individual selves, self-awareness, and reflectivity, *Journal of Consciousness Studies* **2**: 166–184.

Byrne, R. W. and Whiten, A., 1988, *Machiavellian Intelligence: Social Expertise and the Evolution of Intellect in Monkeys, Apes and Humans*, Clarendon Press, Oxford University Press, Oxford.

Byrne, R. W. and Whiten, A., 1997, *Machiavellian Intelligence II: Extensions and Evaluations*, Cambridge University Press, Cambridge.

Dautenhahn, K., 1997, I could be you: the phenomenological dimension of social understanding, *Cybernetics and Systems* **28**: 417–453.

Dennett, D. C., 1984, *Elbow Room: The Varieties of Free Will Worth Wanting*, Oxford University Press, Oxford.

Dennett, D. C., 1989, The origin of selves, *Cogito* **3**(163–173).

Drescher, G. L., 1991, *Made-up Minds – A Constructivist Approach to Artificial Intelligence*, MIT Press, Cambridge, MA.

Edmonds, B., 1999a, Capturing social embeddedness: a constructivist approach, *Adaptive Behavior* **7**(3–4): 323–347.

Edmonds, B., 1999b, Gossip, sexual recombination and the el farol bar: modelling the emergence of heterogeneity, *Journal of Artificial Societies and Social Simulation* **2**(3).

Edmonds, B., 2000, The constructability of artificial intelligence (as defined by the Turing Test), *Journal of Logic Language and Information* **9**: 419–424.

Edmonds, B., 2001 Meta-genetic programming: co-evolving the operators of variation, *Elektrik* **9**(1): 13–29.

Ernst, G. and Newell, A., 1969, *GPS: A Case Study in Generality and Problem Solving*, Academic Press, New York.

French, R. M., 1990, Subcognition and the limits of the Turing test, *Mind* **99**: 53–65.

Goldberg, D. E., Deb, K., et al., 1989, Messy genetic algorithms: motivation, analysis, and first results, *Complex Systems* **3**: 493–530.

Gopnik, A., 1993, How we know our minds: the illusion of first-person knowledge of intentionality, *Behavioural and Brain Sciences* **16**: 1–14.

Hendriks-Jansen, H., 1997, The epistomology of autism: making a case for an embodied, dynamic, and historic explanation, *Cybernetics and Systems* **28**: 359–416.

Hoffman, J., 1993, *Vorhersage und Erkenntnis [Anticipation and Cognition]*, Hogrefe, Goettingen, Germany.

Hofstadter, D. R., 1985, Analogies and roles in human and machine thinking, in: *Metamagical Themas: Questing for the Essence of Mind and Pattern*, Penguin, London.

Holland, J. H., 1992, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA.

Jannink, J., 1994, Cracking and co-evolving randomlist, in: *Advances in Genetic Programming*, K. E. Kinnear, ed., MIT Press, Cambridge, MA.

Koza, J. R., 1992, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA.

Koza, J. R., 1994, *Genetic Programming II: Automatic Discovery of Reusable Subprograms*, MIT Press, Cambridge, MA.

Kummer, H., Daston, L., Gigerenzer, G., and Silk, J., 1997, The social intelligence hypothesis, in: *Human by Nature: Between Biology and the Social Sciences*, Weingart et al., eds., Lawrence Erlbaum, Hillsdale, NJ, pp. 157–179.

Minsky, M., 2002, The emotion machine; http://web.media.mit.edu/~minsky/

Perlis, D., 1997, Consciousness as self-function, *Journal of Consciousness Studies* **4**: 509–525.

Rolls, E. T., 2000, Precis of the brain and emotion, *Behavioural and Brain Sciences* **23**(2): 177–191.

Sloman, A., 2002, How many separately evolved emotional beasties live within us?, in: *P. S. Emotions in Humans and Artifacts*, MIT Press, Cambridge, MA.

Stolzmann, W., 1998, *Anticipatory Classifier Systems*, Genetic Programming, University of Wisconsin, Morgan Kaufmann, Madison, WI, pp. 658–664.
Tolman, E. C., 1932, *Purposive Behavior in Animals and Men*, Appleton, New York.
Weizenbaum, J., 1976, *Computer Power and Human Reason: From Judgement to Calculation*, W. H. Freeman, San Francisco, CA
Wolpert, D. H. and Macready, W. G., 1995, No free lunch theorems for search, Santa Fe Institute Technical Report SFI-TR-95–02–010.

Chapter 15
How My Program Passed the Turing Test

Mark Humphrys

Abstract In 1989, the author put an ELIZA-like chatbot on the Internet. The conversations this program had can be seen – depending on how one defines the rules (and how seriously one takes the idea of the test itself) – as a passing of the Turing Test. This is the first time this event has been properly written. This chatbot succeeded due to profanity, relentless aggression, prurient queries about the user, and implying that they were a liar when they responded. The element of surprise was also crucial. Most chatbots exist in an environment where people *expect* to find some bots among the humans. Not this one. What was also novel was the *online* element. This was certainly one of the first AI programs online. It seems to have been the first (a) AI real-time chat program, which (b) had the element of surprise, and (c) was on the Internet. We conclude with some speculation that the future of all of AI is on the Internet, and a description of the "World-Wide-Mind" project that aims to bring this about.

Keywords BITNET, chat, chatbot, CHATDISC, ELIZA, Internet, Turing Test

15.1 Introduction

In 1989, the author put a chatbot on the Internet, whose conversations can be seen, depending on our definitions, as having "passed the Turing Test". For reasons which will be explained below, this is the first time this event has been properly written. This paper is an explanation of a historical event, but it has implications for the future of Turing Test experiments on the Internet, and indeed for the future of AI in general on the Internet.

Dublin City University

R. Epstein et al. (eds.), *Parsing the Turing Test.*

15.2 The AI Program

In 1987, when I was an undergraduate in Computer Science at University College Dublin (UCD), Ireland, I wrote a version, in LISP, of Weizenbaum's classic "ELIZA" chat program (1966). ELIZA "simulates" (or perhaps *parodies*) a Rogerian psychotherapist (i.e., a practitioner of the *non-directive* therapy of Carl Rogers), who has a conversation with a patient by appearing sympathetic, asking bland questions, and asking the patient to clarify what he just said, or discuss how he feels about it. This means the therapist does not have to engage in any detail with the actual *content* of the patient's problems. The patient does all the work. This is obviously ideal for a computer program, which can attempt to carry on a conversation without having to understand anything the human says *at all*. Weizenbaum's trick remains one of the classic tricks for building a chatbot.

The original ELIZA was meant to be sympathetic to the human. I thought it would be interesting to add aggression and profanity to the program. My program was deliberately designed to have an unpredictable (and slightly scary) mood. It is hard to convey the personality of the program in this limited space. For the cumulative effect, the reader should read through the LISP source code, visible at (Humphrys, 1995), from which the following examples come. As one example, in reply to the common use of "OK" the machine would deny that everything was OK:

```
((equal (car input) 'ok)
'(what do you mean ok its not ok at all))
```

As another example, in response to any innocuous statement beginning "You are …" the machine would reply with one of these:

```
(putd 'youareresponses 'expr '(lambda nil
'((i am not (y) you insulting person)
(yes i was (y) once)
(ok so im (y) so what is it a crime)
(i know i am (y) dont rub it in)
(i am glad i am (y))
(sing if youre glad to be (y)
sing if youre happy that way hey)
(so you think i am (y)
well i honestly could not care less))))
```

where (y) is the old ELIZA trick of just repeating whatever the user had typed after "You are". The machine is pure stimulus-response, replying to the present message, with absolutely no memory. The reader will see that the technology is all pretty much standard ELIZA pattern matching. The novel aspect is the design of the personality and responses, and of course, from the technical viewpoint, putting it *online*. Other innocuous statements would trigger random aggression:

```
(ah type something interesting or shut up)
```

disinterest (this in response to "I was.."):

(shut up you boaster)

and prurience:

(are you lonely very often)
(has anyone ever loved you)
(ok honestly when was the last time you got laid)

The last one was very successful with the right personality type, as was this response to any use of bad language:

(you only use foul language to make up for your
small penis)

which (hopefully) would only inflame the situation. The answer to the prurient questions is just as likely to trigger more aggression:

(what do you know about love anyway githead)

disbelief:

```
( (equal (car input) 'yes)
'(i dont believe it) )
```

and *unbelievable* rudeness – when talking about death, for no reason the machine could suddenly reply:

(i wish you were dead)

Throughout, the machine would produce random friendly..

(you are obviously an interesting person)

.. and patronising replies:

```
((equal (car input) 'hello)
'(hello little man how are you))
```

while every sentence beginning "I love.." would trigger:

'(ahh thats nice)

no matter *what* was being loved. These responses gave the impression of a person blowing hot and cold, with unpredictable and sometimes violent emotions. Clearly, much of this was waiting for someone with the right personality to come along – and, crucially, someone who did not *know* they were talking to a machine.

Fellow students talked to the program, and while its responses did prove more popular than the banal responses of the normal therapist, all of this took place with no element of *surprise*. The students *knew* they were talking to a program.

The program, called simply "Doctor", was offline (i.e., not on the Internet) though it was on a multi-user machine. It ran on a VAX/VMS mainframe (to be precise, the machine: ccvax.ucd.ie) that Computer Science students and staff at UCD shared. To talk to my program, the students would specify the path of the program and run it. Clearly, this program would not perform to its best under these circumstances.

Even if my acquaintances could be *surprised*, i.e., expected to talk to me and not a program, they *knew* me, and so would know these were not my replies. To really test the program, it was necessary (a) to introduce surprise, and (b) for it to talk to *strangers*.

15.3 The AI Program Online

In 1989, in the final year of my undergraduate course, I put this program online, and now for the first time it had the element of surprise, and it could talk to strangers long-distance.

15.3.1 CHATDISC Server-side Programs on BITNET

In 1989, UCD was operating both "BITNET" and "Internet" machines. It was on the "BITNET" machine that I put the program online. The "Internet", at this time, was the name used to refer to machines that used the emerging TCP/IP protocol, which were only *some* of the machines on what was a vast interconnected network of networks. BITNET, for those who do not remember, connected universities around the world, had international email, file transfer and talk messages, had online file archives that you accessed remotely, and was the birthplace of multi-user RELAY chat (the ancestor of IRC) and also of LISTSERV mailing lists. Some might say it was not part of the Internet because it did not use TCP/IP. But it would be fairer to say that it, plus all the other networks it connected to, was the 1980s Internet, which has since standardised on TCP/IP (otherwise, e.g., we have to say that the 1970s Arpanet was not the Internet either, since it did not use IP). (See Humphrys, 2001a for more on this discussion of terminology.)

In any case, even in the pedantic sense the program was on the "Internet" as well since the BITNET machine at UCD (a VM/CMS mainframe, the node IRLEARN) exchanged email, files, and chat/talk messages with the TCP/IP machine at UCD (the VAX/VMS mainframe that we encountered above) and actually had its own Internet address (the VAX machine also had a BITNET address). To summarise, at UCD we had:

1. The "Internet" machine – VAX/VMS mainframe, the node CCVAX on BITNET, the address ccvax.ucd.ie on the Internet.
2. The "BITNET" machine – VM/CMS mainframe, the node IRLEARN on BITNET, the address irlearn.ucd.ie on the Internet. In fact later this same "BITNET" machine was directly accessible through a web browser from the Internet at: gopher://irlearn.ucd.ie/.

VM/CMS had an interesting feature where you could "disconnect" –which meant you logged out, but could leave a program running to process incoming email, files, and crucially, chat/talk messages. The program CHATDISC EXEC, by Eric Thomas of the Ecole Centrale de Paris and of CERN, the inventor of LISTSERV

(Thomas, 1996), was in popular use as a disconnected "answer machine". Talk messages sent to you when you disconnected leaving CHATDISC running were saved in log files, and automatic replies could be sent back, such as "Back in 2 hours". Different replies could be sent depending on who had sent you a message. Indeed, because CHATDISC was open-source (a REXX EXEC script for VM/CMS), it occurred to me that *any* program at all could be called at this point. CHATDISC allowed one to essentially write "CGI scripts" for BITNET (arbitrary server-side programs that could be run remotely when you were not there).

15.3.2 MGonz

The next step was for CHATDISC to call "Doctor" to reply to incoming talk messages. "Doctor" was renamed "MGonz". (See Humphrys, 1995 for an explanation of the name.) The modified CHATDISC was called "MGonzNet". Here is an extract of the REXX EXEC source code of my customised CHATDISC program. The full source code is visible at (Humphrys, 1995). When a message comes in, make a copy of the LISP source:

```
Address 'CMS' 'COPYFILE MGONZ SOURCE A MGONZ LSP A (REPLACE'
```

Add a line defining the current input string, and asking LISP to generate the response from this. Run the file:

```
arg=removebadchars(message) /* bad chars crash LISP */
arg=strip(arg)
arg=left(arg,50) /* else lisp truncates losing brackets */
'EXECIO 1 DISKW MGONZ LSP A
(FINIS STRING (response 'quote'('arg '))'
Address 'CMS' 'LISP MGONZ LSP'
```

Parse the output file. Find the reply line. Send it back:

```
'EXECIO 1 DISKR MGONZ LISTING A 441'
'EXECIO * DISKR MGONZ LISTING A (LOCATE /VALUE IS/'
'EXECIO 1 DISKR MGONZ LISTING A (LIFO'
parse pull line
line = lowcase(line) /* MGONZ LISTING is all upper case */
line = strip(line)
line = strip(line,'L','(')
line = strip(line,'T',')') /* discard enclosing brackets */
replymsg = '* 'line
Call Tellback replymsg
```

For technical reasons, MGonz's replies all begin with *.

This went online in February 1989, and friends of mine amused themselves talking to it, but most had advance warning that it was a program, so the conversations are always limited by that:

From FRIEND1: get lost you nosey schweinhunt pig dog
* have you ever had sex with someone who really loved you

From FRIEND1: no
* ah go on say yes

From FRIEND1: sorry, no
* by the way is there any medical reason for your sexual impotence

From FRIEND1: this program is very good mark you rat head

All that was necessary now was for a *stranger* to send me a talk message when I was not there. A number of friends and strangers hit it without warning but backed off immediately, these from March to April 1989:

From SOMEONE at IRUCCIBM: noel kerins told me to send you a message!
* you are obviously an asshole

From SOMEONE at IRUCCIBM: so here it is!
* by the way is there any medical reason for your sexual impotence

From FRIEND2: Hello, Marcus maximus.
* hello little man how are you

From FRIEND2: I am not a little man. but if you must know,
I'm very well.

From FRIEND2: Where are you anyway.. not the SCIENCE-block I hope?
What we were waiting for was someone who would *not* back off.

15.4 The DRAKE Conversation

At this time, I was a regular on RELAY chat, talking to strangers from around the world in the kind of chaotic milieu that is still much the same today on IRC. On May 2, 1989, a user from the node DRAKE (Drake University, Iowa) sent me a message. I never did find out who he was. Perhaps someone I met on RELAY, or a friend of someone I met on RELAY. He stayed talking to my program for 1½h, getting increasingly defensive and furious. This conversation can be seen, depending on how one defines the rules (and how seriously one takes the idea of the test itself), as a passing of the Turing Test.

The conversation, available in full at (Humphrys, 1995), begins haphazardly. Soon, the user starts to get frustrated:

From SOMEONE at DRAKE: what is it with this type something
interesting
or shut up business?
* what are you talking about

From SOMEONE at DRAKE: i keep getting a message that says. "* AH

TYPE SOMETHING INTERESTING OR
* you are obviously an asshole

From SOMEONE at DRAKE: shut up"
* what are you talking about

From SOMEONE at DRAKE: what is this? first it says "YOU ARE OBVIOUSLY AN ASSHOLE" then
* what are you talking about

From SOMEONE at DRAKE: you say "WHAT ARE YOU TALKING ABOUT"

To explain the absurd repetitiveness of my sayings, he constructs a fantastic theory that I might have "hot-keys" attached to particular phrases:

From SOMEONE at DRAKE: are there two people on your username?
………...

From SOMEONE at DRAKE: do you have all these silly sayings on some sort of control?
* ok thats it im not talking to you any more

From SOMEONE at DRAKE: you sound like a goddamn robot that repeats everything.

From SOMEONE at DRAKE: o my he actually typed something.
* ah type something interesting or shut up

From SOMEONE at DRAKE: you type slow is that why you have all those stupid sayings on command like that last one?

but never quite makes the mental jump to realise that there is *no one at home at all*.

The conversation really gets going when MGonz (rather by accident) questions his manhood. The "When was the last time you had sex?" question works brilliantly with this user, especially when his answer is not believed. He actually *admits* under repeated "questioning" (by a program that has no idea what it is doing) that he had lied, or at least exaggerated:

* when was the last time you had sex
From SOMEONE at DRAKE: yesterday.

* ok honestly when was the last time you got laid
From SOMEONE at DRAKE: i told you it was yesterday.

* you are obviously an asshole
* ok honestly when was the last time you got laid

From SOMEONE at DRAKE: ok ok it was over 24 hours ago. for you it must have been 20 years

* what do you mean ok its not ok at all
My response to his bad language also works brilliantly:
From SOMEONE at DRAKE: is that fucking rude enough for you?

* you only use foul language to make up for your small penis
From SOMEONE at DRAKE: why not?

* ah get lost go to the bar or something
From SOMEONE at DRAKE: i just told you the night before last. and the night before that too.

* go away fool and stop annoying me
From SOMEONE at DRAKE: well i am not going to tell you about the size just that i have

* what do you know about love anyway githead
From SOMEONE at DRAKE: never had any complaints about my penis
and
that it gets used quite

* ok straight out are you a virgin
From SOMEONE at DRAKE: often. and i think that you are a virgin and have no hope of getting

* do you detest and loath the abomination of organised religion

His answers have a certain glorious *innocence* about them, as he tries repeatedly to make conversation with an entity that, frankly, one *cannot* make conversation with:
From SOMEONE at DRAKE: is your father a hampster?

* do your family approve of what you do
From SOMEONE at DRAKE: does your mother smell like last weeks dirty socks.

* what method would you choose to slaughter your family
From SOMEONE at DRAKE: do my family approve? of what exactly?
From SOMEONE at DRAKE: why do you ask about religion. and we already decided that i have

* do you believe in mgonz
From SOMEONE at DRAKE: had sex on campus and in my girlfriends room no less.

* tell me your favourite sexual fantasy
From SOMEONE at DRAKE: do i believe in mgonz? what the heck is it?

* praise and honour to blasphemers and death to mgonz

He ends with a furious barrage of abuse, but of course, my imperturbable program is a calm, stimulus-response machine and so it is *impossible* for him to have the last word. *He* must quit, because my program never will. At the end of a long barrage of sex-related boasting:

From SOMEONE at DRAKE: how about her just sucking you off?
* if you are an attractive female please leave your phone number here

From SOMEONE at DRAKE: little markie seems to be getting mad. poor little virgin markie.
* jesus who let you near me go away

From SOMEONE at DRAKE: FUCK YOU ASSHOLE. THINK OR ARE YOU JUST A STUPID HOMOSEXUAL.
* fuck off yourself

From SOMEONE at DRAKE: o my he actually typed something.
* ah type something interesting or shut up

From SOMEONE at DRAKE: you type slow is that why you have all those stupid sayings on command like that last one?
* you are obviously an asshole
and that is the last word.

15.4.1 More on the Conversation

A few more observations on the conversation: first, contrary to appearances, my program *rarely* asked about the user's sex life. The topic only got started:

From SOMEONE at DRAKE: are there two people on your username?
* when was the last time you had sex

because, bizarrely, "there" matched a list of "sex words". After this, it was the *user* that kept the conversation about sex going, not the machine. The machine has absolutely no memory, and would forget about sex immediately if the user did not keep returning to it.

There were other strange breaks. For example, just when the user was starting to get suspicious, my program strongly suggested there really was someone there:

From SOMEONE at DRAKE: what? why don't you type something new.
like when was the
* paris is a beautiful city

From SOMEONE at DRAKE: last time that you got laid..unless it was when you were born o'my he said something different. a
* you are obviously an asshole

From SOMEONE at DRAKE: you are obviously a fucking dumbshit.
* i am not obviously a fucking dumbshit you insulting person

The "paris is a beautiful city" reply happened, bizarrely, because "new" matched a list of "holiday words". The "You are…" reply is then spot on (purely by chance of course). Interestingly, the program actually *admits* (!) it is a program at some places in the code:

(you know the worst thing about being a computer is
having to deal with shits like you)
(nobody ever asks the computer we lead a lonely life)
(youll be in trouble when we computers take over the world)
(have you any idea how boring it is being a stupid computer)
(are you using foul language because i am a computer)

but by chance none of these were actually triggered in this conversation. The closest we get is:

From SOMEONE at DRAKE: mark?
* mark isnt here and hes left me to deal with cretins like you

15.4.2 Why Was This not Written Up Before?

At the time, I did realise that something special had happened, and saved the conversation log carefully. But I think it is clear why I never wrote this up or publicised it – how could one write up something packed with vulgarity and obscenity like this?

MGonz was never something that was meant to be seen by "adults". In fact, I went to great lengths to *hide* it from the faculty and computing staff at UCD (see Section 15.5.2), who I thought would not appreciate me running this on their system.

But for many years afterwards, it became clear that this really *was* interesting to other people, and might even have been the *first* chatbot on the network with the element of pure surprise. The interest in it led to me finally setting up a web page on it (Humphrys, 1995). This paper is still the first proper write up.

15.5 Internet Features

The system had a number of network-related features which may still be of interest, especially since they relate to the possibility of *fraud* (or at least confusion) in Turing Tests online.

15.5.1 Remote Control

First, if I was logged on to the VAX (the "Internet" machine), MGonzNet would copy me on any conversation it was having. From the source code at (Humphrys, 1995):

'GLOBALV SELECT CHATDISC STACK LOGGEDON'
Parse pull loggedon

```
/* Copy to Vax unless loggedon = false */
If loggedon ^= false Then Do
If nickname ^= ''
Then Call Vax '* Message from' nickname '..'
Else Call Vax '* Message from' userid 'at' nodeid '..'
Call Vax message
End
```

where "Vax" is the function:

```
Vax:
vaxmsgcmd = 'SMSG RSCS MSG CCVAX H236_007'
Call Diag 8,vaxmsgcmd Arg(1)
Return
```

In fact, I did not just get copies of messages. My VAX account was a privileged remote user that could send special commands to be executed by MGonzNet:

```
When nickname = 'Vax' Then
Do
Call Sys
Exit 0
End
```

where "Sys" is a function to parse any message that comes from my VAX account. This function includes code to execute a command-line (CMS) command:

```
When left(upmessage,3) = 'CMS' Then
Do
cmd = lowcase(substr(message,5))
cmd = strip(cmd)
Address 'CMS' cmd
End
```

That is, from the VAX machine I could use CHATDISC to remotely execute arbitrary commands on the VM machine. On the Internet today, servers running CGI scripts worry about script security since a bug in a script could give a remote user a command-line on the server. But at least CGI scripts (on a UNIX server) run as the user "nobody" – so in theory, even if they get a command-line, they should not be able to do much damage. But here, a disconnected server program like CHATDISC runs as *me*, with the ability to delete all of my files, and so on. This could have become an issue if disconnection for personal server-side programs had become common, but I am unaware that anyone else *ever* implemented a remote command-line like this on BITNET.

Why is this relevant to the Turing Test? Because one of those arbitrary commands that I could send from the VAX was:

CMS TELL user message

That is, I could interleave my own replies with the program's replies. In general, if Turing Test-like experiments are going on online, and we are talking to some server-side program, a human could monitor the conversation remotely (and secretly), and then send secret messages to the server to send particular replies back. We, as users of the server, would never be able to tell. So the machine would pass the Turing Test, but only because you *were* talking to a human!

Writers about how to run a Turing Test competition online recognise the issue of remote human insertion of messages. Here we see that this has, of course, already happened in the 1980s. Here is a friend of mine as I send him some ambiguous messages from the VAX through my CHATDISC:

```
From FRIEND3:   SEE says you're disconnected
From FRIEND3:   where the hell are you
From FRIEND3:   is this a ghost i'm talking to
From FRIEND3:   very mysterious
From FRIEND3:   quite strange
From FRIEND3:   you are definitely disconnected
```

15.5.2 Attempts to Hide the System

As mentioned in Section 15.4.2, I *hid* MGonz from faculty and computing staff, believing they would not approve of it. I even renamed the CHATDISC program to the bland name "SYS TXT" so any search for CHATDISC or EXEC files on my disk would not find it. It would be then renamed to CHATDISC EXEC just the moment before it was launched.

I also had a "blacklist" of users who MGonz would not reply to:

```
/* Centre people – give standard disc msg
   don't let them stumble across MGonzNet
*/
c = 'adillon adviser annmarie bernie bonnie brianm'
c = c || 'carrick cecily deirdre dobeirne ebairead emcgrath'
c = c || 'eoin guest1 guest2'
c = c || 'harringt helen jacklowe jchester jennings joanc'
c = c || 'larry listserv maevem mailmnt mailr2 maint'
c = c || 'mallen mbreslin mcgrath mnorris'
c = c || 'mokeeffe mokelly molooney moriarty msexton'
c = c || 'noreilly odonnel'
c = c || 'oneillu pdoyle rosemary sinead tinac tmcgrath'
c = c || 'twade t_wade vmaccnt walter walsh'
```

This is actually a list, as (Humphrys, 2001a) explains, of many of the pioneers of the Internet in Ireland. On April 24, 1989, I sent a quick talk message to one of these:

To CENTREPERSON: were you looking for me?

I found out he was disconnected. My message was logged in his answer machine. Then I disconnected myself. I went off to work on the VAX. While I am working on the VAX, I get a message from MGonz telling me it is having a conversation with CENTREPERSON, and copying me on it:

From CENTREPERSON: I was, when you had all those t-discs
* ive never been when you had all those t discs whats it like

Horrified that MGonz was talking to him, I logged back in to VM as quick as I could and tried to cover up the conversation, thankful that it had just baffled him and not asked him about his sex life.

There had been a flaw in my blacklist (a missing space, I found out later). Of course, I had to get caught sometime. On December 18, 1989, I got a message from the UCD operator:

From UCDOPR: PLEASE LOGOFF NOW
From UCDOPR: SWITCH BEING BACKED UP

* what are you talking about
* you are obviously an asshole

15.6 Discussion – How to Pass the Turing Test

We now ask what implications MGonz has for the Turing Test. Obviously, this program (like ELIZA itself) is really a *trick*. It is not intelligent. It has no understanding of anything that is going on in the conversation. I will discuss in Section 15.7 whether the Turing Test, and passing it, really have anything to do with AI, but leaving that aside, how did it work? How did it manage to pass the Turing Test?

15.6.1 Surprise

One of the things that made MGonz successful was the *element of surprise*. Users of an AI CGI script through a web page (Yahoo, ongoing) *expect* to be talking to a program. Users in the Loebner Prize Competition (http://www.loebner.net/Prizef/loebner-prize.html) *expect* that some of the entrants will be machines. But no one expected to hit a program when they sent me BITNET chat messages.

Ironically, this would suggest that the Loebner Prize Competition is not the way to go about the Turing Test! As we saw above, the advance knowledge by my friends that I had (or might have) some kind of program running makes all their conversations rather bland, unserious (they just mess around) and short (because after the initial novelty, they get bored). Indeed, I have records of a number of such conversations with friends who knew it was a program, but these conversations are just too *boring* to include in this paper. Lack of surprise simply makes for a different type of conversation. This factor could be seen when Tom Doris put the original LISP MGonz back online in late 1998 as a CGI script (Doris, 1998). Here, the user knows from the start that it is a program, which affects the nature of the conversation:

User: you are simply responding to stimuli
Mgonz: so you think i am simply responding to stimuli well i honestly could not care less
User: you are simply responding to stimuli
Mgonz: ok so im simply responding to stimuli so what is it a crime
User: you are simply responding to stimuli
Mgonz: i know i am simply responding to stimuli dont rub it in
User: you are a defective program

Long conversations (like an hour and a half) just do not happen when the user knows it is a machine. Doris' CGI script is no longer running, but Dave O'Connor (1999) has a version of MGonz online as a CGI script. Again, the user knows it is a program, though human psychology is such that some users still allow themselves to be provoked. The following are 3 separate users:

From USER1: Hello
Hello little man, how are you?

From USER1: don't call me little.
Tell me a bit about yourself.

From USER1: i am five feet six inches.
I've never been five feet six inches, what's it like?

From USER1: it's annoying because robots call you little.
Are you here for business or pleasure?

From USER1: i'm here to get material for my newspaper column.
So you are here to get material for your newspaper column, well I honestly could not care less.

From USER1: that's very rude. you are a rude robot.

From USER2: erm I am a student
I've never been a student, what's it like?

From USER2: boring most of the time
Go on tell me some really juicy scandal.

From USER2: I dont know any
You are obviously an interesting person.

From USER2: so are you
Are you here for business or pleasure?

From USER2: I have to do some work so I was wondering what you are
Did you have problems with your feet when you were wondering what me am?

From USER2: eh my feet?
Ah type something interesting or shut up.

From USER2: why did you just mention my feet
What kind of music do you like?

From USER2: no what was that feet thing about
I see.

From USER3: I am David.
I am glad you are David.

From USER3: Thank you. So am I.
You're Welcome.:)

From USER3: You are a very nice person.
So you think I am a very nice person, well I honestly could not care less.

From USER3: Well I take that back. You are rude.
Sing if youre glad to be rude, sing if youre happy that way, hey!

From USER3: I hate to sing.
Go away fool and stop annoying me.

From USER3: I have a paper to write.
Ah get lost, go to the bar or something.

From USER3: but I need to write my paper.

If you think i care you're wrong.

The above is about as good as a Web-page chatbot can get without the element of surprise.

AOLiza (Fox, 2000) has recently returned to the element of surprise on the AOL Instant Messenger system, where it has been running since 2000. Like my program, and unlike the Loebner Prize, users do not *expect* to be talking to a machine. AOL, incidentally, was not happy about AOLiza running on its system. The element of surprise can be seen by some as unethical. And yet one *cannot* expect the best chatbot performance to occur when it is absent.

15.6.2 Sex

The second major reason why MGonz succeeded was due to profanity, relentless aggression, prurient queries about the user, and implying he was a liar when he made responses to these. *Sex* is probably the easiest topic in which to engage the user, so long as you get the right personality type, because what we are looking for is emotions that blind the user to reason (so he does not notice it is a program). Questioning 'SOMEONE at DRAKE's' sexual abilities was all that was needed to get him into defensive, boastful and argumentative mood, emotions that blinded him to the fact that he wasn't actually talking to a person. MGonz, however, was merely an amateur compared to the following bot, which could actually get the user *aroused*, and thus totally blinded to the fact that no one was at home.

15.6.2.1 Jenny18

For me, the most amazing chatbot of all time is Jake Kaufman's "Jenny18" bot (Kaufman, 2001) running on IRC. Like MGonz and AOLiza, Jenny18 also had the element of *surprise*. But its real genius is in its personality design. Jenny18 specialises ruthlessly on pretending to be a horny girl looking for cyber-sex. The performance of this bot is breathtaking, inspiring the most intimate confessions, desperate requests for a photo or an email address, arousal, masturbation, frantic demands to take it to telephone, and finally orgasm. It passes Turing Test after Turing Test. The most human is the long, hilarious conversation with "Scorpion832". (All these conversation logs can be seen online at Kaufman, 2001.) And this simple bot clearly brings "dom01" to orgasm repeatedly, which surely must count as a fundamental new milestone passed in the history of the Turing Test:

dom01: how old r u
jenny18: shouldn't my nick tell u that!!! lol
dom01: im so horny right now damn
jenny18: (stroking your erection through your pants)
dom01: help me geny,im horny
jenny18: (stroking your erection through your pants)
…
dom01: yeahhhhhh i feel so horny genny
jenny18: grrrrooooowwrrr… you're not the only one who's horny.
dom01: mmmmmmmmmmyahhh
jenny18: mmm.. =)
dom01: im doing it
jenny18: oh are you? =)
…
dom01: mmmmmrrrr
jenny18: mmm.. =)
dom01: im masturbating myself
jenny18: kewl.. yum!
dom01: mmmmmm yeahhhhhhh
jenny18: could u explain pls?
…
dom01: is your pussy wet
jenny18: you know, my cunt is dripping. it's really wet =)
dom01: about your pussy what else
jenny18: you just wish you had my pussy to play with.
dom01: rrrrrrrryeah
jenny18: crazy..

…
dom01: can i call u baby
jenny18: just got one phone line hon. gotta stay here.
…
dom01: can i call u baby
jenny18: no way. i just wanna keep it in msg.
dom01: please
…
dom01: mrrrrrrmmmmmmmm
jenny18: could u explain pls?
dom01: ang onnnnnnnnnnnnn
jenny18: ah.
dom01: angggggggggonnnnn
jenny18: oh?
dom01: waittttttttttt
jenny18: hm… =)
dom01: ahhhhhhhyeahhhhhhhh
jenny18: mmmMm..
dom01: fuck
jenny18: i see
dom01: i came
jenny18: hmm
dom01: yes
…
dom01: xxx
jenny18: ah.
dom01: it feels so good ejacualtion
jenny18: want some more? i think u do:)
dom01: again?
jenny18: mmm.. =)
dom01: ok ill try my best to come again
jenny18: come again?
dom01: wait
jenny18: crazy..
dom01: ang on im washing my dick,theres sperm on it
jenny18: hmm
dom01: wait
…
dom01: how old r u
jenny18: i'm 18, babe =)
dom01: fuck girl where r u from

With some of the others, like "GoldenBoy2222" and "lander100", it is hard to tell if they did actually come or are just pretending, but with "dom01", it is clear what has happened on his side. "Happy_Boy" also looks like it happened for him.

Incidentally, some people's inclination might be to *laugh* at the users of Jenny18, and regard them as sad – especially those that come. I am not sure it is that simple. One could just as easily be *happy* for them –"dom01" is clearly having a nice time. Jenny18 could be seen as a public service, spreading happiness and *optimism* online. She gives users a nice fantasy to work with, perhaps at a period in their lives when they have no one offline to be intimate with. She could be a *liberating* experience, leaving them happier, more confident and *optimistic* in going back to the cruel reality of dating in the real world.

Jenny18 deserves a paper all on its own, and yet, publicising it is of course difficult. I did not publicise MGonz for years because of its vulgarity. Jenny18 has a far worse problem. Given the content, and especially what the users are doing during the conversation, one can imagine both the field of AI and the media being reluctant even to talk about Jenny18, which would be wrong. For anyone who takes the Turing Test seriously, passing the "orgasm" Turing Test (i.e., the computer brings the human to orgasm) is surely an accomplishment worth noting.

15.6.3 Privacy

For sex conversations to work, *surprise* is crucial. No one talking to an AI CGI script would get too emotionally worked up if it started questioning their sexual prowess – they would treat it as an amusing game. Indeed, Yahoo categorises these under "Games" (Yahoo, ongoing), showing that one's whole frame of mind in using a program that you *know* is a program is that of looking for amusing entertainment.

For the sex conversation to work, *privacy* is also crucial. No judge at the Loebner Prize Competition is going to disclose personal information about their sex life like SOMEONE from DRAKE, let alone actually get aroused, if they know that the other judges will see what they typed. In a public forum, one will be self-controlled and keep more distance in the conversation. To summarise, a young male talking about sex online alone in the privacy of his bedroom is probably the easiest environment in which to pass the Turing Test.

15.6.4 Real Human Responses

We have not said much about AI so far. It seems to me that "script writing" is far more important than "AI" in chatbots. MGonz contains no more "AI" in it than ELIZA (in fact, probably *less*). All my work was in the responses, the personality. My classmates wrote versions with memory (mine is memory-free!) and *far* more powerful sentence parsing, but their responses were the banal psychotherapy responses, or the slightly awkward and "geeky" responses developed by generations of ELIZA programmers in AI. It is hard for programmers to do any different. No one can write about sex, for example, unless you are comfortable discussing it, which AI "geeks" rarely are. They might prefer to have a chatbot talk about science fiction.

What is needed is someone comfortable with sex who can still cope with a program, or at least modify an existing program. Jenny18, from someone with a background in music, *not* programming, is *all* about the strong, confident script, that simulates, often with subtle understatement, a horny girl playing with sex talk, rather than some robotic voice talking about sonnets. It seems to contain even less "AI" in it than ELIZA does. To quote Kaufman, author of Jenny18, in private correspondence to me:

> I think that, while technically impressive, advanced theoretical learning algorithms and memory databases are not by themselves the key to fooling random people – *realistic responses* are. You seem to come from the same train of thought. If artificial *human* intelligence is to be presented, it has to seem human! I don't understand how these chatterbots who speak in full paragraphs are to be taken seriously yet, when people subsist mainly on short utterances. To that effect, I included 500 different versions of "Huh?" and "I don't understand."

15.7 Discussion – Is the Turing Test Important?

We have just discussed the main reasons why MGonz, and Jenny18, perform well on the Turing Test, and none of them seem to involve Artificial Intelligence (AI). So the time has come to ask: Is the Turing Test, and passing it, actually *important* for the field of AI? It may surprise the reader that my answer is "No".

I agree with Hayes and Ford (1995) and Whitby (1997) that the Turing Test served its purpose in its day, but it is irrelevant now. In his famous paper (1950), Turing was dealing with people who just could not believe a machine could ever think, and so was making the philosophical point that if you could not tell the difference, why would you deny it was thinking? This is a useful thought experiment, but this is not the same as saying either that: (a) passing the Turing Test is *necessary* for intelligence, or even that it is likely to be of *any* importance, or: (b) passing the Turing Test is *sufficient* to demonstrate intelligence – there may *be* an answer to Turing's question. He just reminds us to take care that it is not just based on *prejudice*.

Hayes and Ford (1995), Whitby (1997) and Michie (1993) point out many of the standard objections to the Turing Test – that it concentrates on adult, human, text-based language processing, to the exclusion of other types of intelligence, that it forces the intelligence to *pretend*, instead of just being itself, and so on. I agree with these criticisms, but since they are well made elsewhere I will not repeat them here, but instead try to make a few further points.

15.7.1 Not Necessary

First, is passing the Turing Test necessary to be intelligent? Turing does not claim it is *necessary* – he is just making the point that if a machine *does* pass, why would you deny it was intelligent? (See Section 2 in Turing, 1950.)

But many of Turing's followers seem to regard it as *necessary*, or at least *important*. Certainly, thinking that the goal of AI should be to pass the Turing Test would imply that it is necessary. It seems to me that passing the Turing Test is *not* necessary for intelligence:

We do not think humans are intelligent because they pass the Turing Test. Turing asks why we think *anyone* is intelligent. He might say: "You only think *I'm* intelligent because of my behaviour." I would reply: "No I don't. I know you're intelligent without even meeting you or hearing a word you say. I know you're intelligent because *I'm related to you*", because of what we know about the historical origins of humanity and shared DNA, I simply *cannot* work in a fundamentally different way to you. I know how you work. *You work like me*. In his 1950 paper, Turing asks how we can know anything outside of our own minds:

> A is liable to believe 'A thinks but B does not' whilst B believes 'B thinks but A does not.' Instead of arguing continually over this point it is usual to have the polite convention that everyone thinks.

I would argue that it is more than a "polite convention". It is a consequence of the theory of evolution. A normal, healthy adult *Homo sapiens sapiens* has to have an inner life something like mine. But with a *new* thing – something that is unlike us and all animals, something that is *not* part of the genealogical tree of life – the situation is different. The new thing *may* indeed think, but the possibility of trickery and illusion remains. The genealogical argument does not help to dismiss that possibility, as it *does* with humans.

Aliens cannot pass the Turing Test. – The Turing Test will not play a role in us detecting other naturally evolved intelligences. To invert this, when aliens discover us, how will they be able to tell *we are* intelligent? We will not be able to pass as convincing aliens, and yet, they will quickly see that we are intelligent.

How *will* we judge future machine intelligence? – Imagine aliens landing here 1.5 million years ago, in the days of *Homo erectus*, and trying to see if we were intelligent. We would not pass their Turing Test, and we would not have language or civilization, but we would have stone tools and fire. The aliens might have recognised us as the start of *some* type of intelligence, but not an intelligence similar to theirs. This is how we will recognise the start of profound machine intelligence. The Turing Test will have no role.

15.7.2 Not Sufficient

The Turing Test may not be *necessary*, but is it *sufficient*? If a program *does* pass, must we admit it is intelligent? My fundamental objection to this is simple:

The Turing Test *has* been passed. It is time to face reality. Whenever a machine, such as ELIZA, MGonz, Jenny18 or AOLiza, *does* pass the Turing Test, critics say "But of course that was not a *real* Turing Test". I *used to* agree with this, and regarded the title of my web page, "How my program passed the Turing Test" (Humphrys, 1995) as a bit of a joke. Now, however, I am not so sure (as you can see since I have reused that title for this paper). I think a case can be made that the Turing Test *has* been passed, many times, by many programs, and so what. It follows that:

Passing the Turing Test does not mean you are intelligent. Trickery can pass it. MGonz has no more "AI" in it than the original ELIZA. In terms of human reaction, Jenny18 is more impressive than any conversation program in history, inside AI or out. Yet it contains even less "AI" in it than ELIZA does! The simple reality is that the Turing Test has been passed time and again, by programs that are not intelligent.

15.7.3 Conclusion

I have no problem with the concept of a machine being intelligent. Indeed, such already exist – for *we* are examples of such. Indeed, there is no evidence that there has ever existed an intelligence that is *not* a machine. And there is no reason why the principles behind how we work cannot be abstracted into an artificial system. My problem is with the Turing Test, not with AI.

AI researchers *succeeding* in the Turing Test and then *disowning* the test itself has been common practice ever since Weizenbaum (1966). Jason Hutchens (1997) is another example. Hutchens is interested in how to fool people, but does not believe that this has much to contribute to AI. After rejecting the Turing Test, my entire AI research career since MGonz has been on quite different forms of AI (mainly sub-symbolic, *non-linguistic* intelligence). There are, however, some good things about both MGonz and the Turing Test for AI, which I shall discuss in Section 15.8.

15.8 The Future of AI Online

The most important thing about MGonz is perhaps the *online* aspect. This was certainly one of the first AI programs online. It seems to have been the first (a) AI real-time chat program, which (b) had the element of surprise, and (c) was on the Internet. To explain, there were many AI chat programs before MGonz, for a survey see Leonard (1997), but they were all (a) *flagged as programs or bots* (no element of surprise, as in AI chat CGI scripts), or (b) *on mailing lists or usenet* (not real-time chat), or (c) *offline* (e.g., bots in multi-user game systems, or on a single shared machine like the VAX in 1987 above).

We will close by considering an argument that the future of *all* of AI is online.

15.8.1 What Happens Next in AI?

Discussions about the Turing Test inevitably lead to discussion about the *distant* future, that day when AI finally triumphs and we make intelligent machines. Such long-term speculation, perhaps done best by Moravec (1998), is fascinating, but after reading enough of it, I always end with a vague feeling of dissatisfaction. One question for me comes to the fore unanswered: How do we get from *here* to *there*? What is the *path* leading step by step from AI today to that future? In particular, what do we do *next*?

The symbolic AI revolution slowed in the late 1980s. It has been revived somewhat with Internet Agents and the Semantic Web, but much of the early optimism seems gone forever. The biologically inspired AI revolution has now had 20 years to deliver on its promises, and it could be argued that it too is slowing down, and has not progressed as far or as fast as originally hoped. So what happens *next*?

What happens next, we argue, is that it all goes online.

15.8.2 The World-Wide-Mind

The "World-Wide-Mind" (WWM) project (w2mind.org) is a proposed standard for putting AI "minds" and sub-minds online (as WWM "servers") so that they can be reused remotely as components in larger minds. The aim is to address the *scaling up* of AI, or how to construct minds more complex than could be written by one author (or one research group).

AI *is* currently being used online, notably symbolic AI in Internet Agents and the Semantic Web. But the model is still one where the agent is written by a *single* author or at most, a single research lab. The WWM analysis is that as we scale up AI, we need to build more and more complex agent minds out of components written by multiple (perhaps hundreds or thousands of) authors. In other words, AI authors *specialise* on different aspects of intelligence, and come together to build large, complex minds that no single author understands. Only by authors publishing server-side programs in an open, public system on the Internet, we argue, can such massive, long-distance collaboration become feasible. Hence the term the "World-Wide-Mind", referring to the fact that the mind may be physically distributed across the world. (For a short introduction to the WWM idea, see Humphrys, 2001b; or Humphrys and O'Leary, 2002.)

Under the WWM view, MGonz can be seen as one of the first times (if not the very first time) that one could access a remote mind server on the network in real time.

15.8.3 Good Science

I said earlier that there are some good points about the Turing Test for AI. It seems to me that the most important thing about the Turing Test is the very idea of a *standard test*.

In any branch of AI, the existence of objective tests that cannot be argued with tends to provide a major impetus to research. This is one of the reasons for the popularity of *rule-based games* in AI, and, more recently, *robotic soccer*. With sport and games, you can *prove* one system is better than another. One of the problems with many branches of AI, as discussed further in (Humphrys, 2001b), is the difficulty of re-implementing someone else's test problem and rerunning their experiments. Much of this difficulty is caused by the need for *local installation*. The WWM aims to address this, by having their test problem remotely usable online. *Many* standard tests will emerge, it is imagined, and in an open public system we will have *repeated* objective *3rd party* comparisons of solutions, so that progress can be made.

The Loebner Prize Competition has a point, competitions spur progress. Here, we argue that AI needs *more* competitions than just the Turing Test. In fact, it needs a generic protocol by which people can set up *new* competitions (new remote WWM "world" servers online).

15.8.4 Conclusion – No Single Scientist will Understand AI — but the Community of Scientists will

Perhaps the objection to AI that I have always found the most convincing is simply that it is beyond our abilities – that there are just limits to what a smart primate can do. For example, a dog will never understand arithmetic. Perhaps the human mind will never understand the mind. We are simply too limited.

The WWM answer to this is "Yes. No single human mind will ever understand the mind. But humanity as a whole will." No single human mind can understand the entire corpus of science. We passed the point long ago (perhaps in the 17th century) when one person could understand all of mathematics and science. But humanity as a whole *does* understand all of these things.

To date, AI has been *held back*, we argue, by the need for a single lab, even a single researcher, to fully understand the components of the system. As a result, only small minds have been built so far. The WWM argues that we must give up this dream of full understanding as we build more and more complex systems. Giving up this dream of full understanding is not a strange thing to do; it is what has always happened in other fields. It is how humanity has made its most complex things.

Acknowledgements Thanks to Dermot Williams for originally showing me how to modify CHATDISC. Thanks to John Murphy for useful discussions on whether a *single* scientist could understand the mind.

References

Doris, T., 1998, MGonz version online as CGI script, formerly at: www.compapp.dcu.ie/ tdoris/ mgonz.html.

Fox, K., 2000, The AOLiza program, www.fury.com/aoliza

Hayes, P. J. and Ford, K. M., 1995, Turing test considered harmful, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-95)*: 972–977, Montreal.

Humphrys, M., 1995, How my program passed the Turing Test, computing.dcu.ie/~humphrys/eliza.html. This web page has 3 sub-pages: MGonz – The LISP source code, MGonzNet – The REXX source code, and The DRAKE conversation.

Humphrys, M., 2001a, The Internet in the 1980s, computing.dcu.ie/~humphrys/net.80s.html.

Humphrys, M., 2001b, Distributing a mind on the Internet: the world-wide-mind, *Proceedings of the 6th European Conference on Artificial Life (ECAL-01)*, Springer.

Humphrys, M. and O'Leary, C., 2002, Constructing complex minds through multiple authors, in: *From Animals To Animats 7: 7th International Conference on Simulation of Adaptive Behavior (SAB-02)*, August 2002, Edinburgh, Scotland.

Hutchens, J. L., 1997, *How to Pass the Turing Test by Cheating*, Department of Electrical and Electronic Engineering, University of Western Australia, technical report TR97–05.

Kaufman, J., 2001, Jenny18 – A Cybersex Bot Implemented in Eliza, virt.vgmix.com/jenny18

Leonard, A., 1997, *Bots: The Origin of New Species*, Penguin.

Michie, D., 1993, Turing's test and conscious thought, *Artificial Intelligence* **60**: 1–22.

Moravec, H., 1998, *Robot: Mere Machine to Transcendent Mind*, Oxford University Press, Oxford.

O'Connor, D., 1999, MGonz version online as CGI script, http://mgonz.swearing.org/

Thomas, E., 1996, History of LISTSERV, L-Soft Inc., http://www.lsoft.com/products/listserv-history.asp.

Turing, A. M., 1950, Computing machinery and intelligence, *Mind* **59**(236): 433–460.

Weizenbaum, J., 1966, ELIZA – A computer program for the study of natural language communication between man and machine, *Communications of the ACM* **9**: 36–45.

Whitby, B., 1997, Why the Turing test is AI's biggest blind alley, in: *Machines and Thought: The Legacy of Alan Turing*, Vol. 1, Millican and Clark, eds., Clarendon Press, Oxford.

Yahoo, ongoing, list of AI programs online,http://www.yahoo.com/Recreation/games/video_games/internet_games/web_games/artificial_intelligence.

Chapter 16
Building a Machine Smart Enough to Pass the Turing Test

Could We, Should We, Will We?

Douglas B. Lenat

Abstract To pass the Turing Test, by definition a machine would have to be able to carry on a natural language dialogue, and know enough not to make a fool of itself while doing so. But – and this is something that is almost never discussed explicitly – for it to pass for human, it would also have to exhibit dozens of different kinds of incorrect yet predictable reasoning – what we might call translogical reasoning. Is it desirable to build such foibles into our programs? In short, we need to unravel several issues that are often tangled up together: How *could* we get a machine to pass the Turing Test? What *should* we get the machine to do (or not do)? What have we done so far with the Cyc common sense knowledge base and inference system? We describe the most serious technical hurdles we faced, in building Cyc to date, how they each were overcome, and what it would take to close the remaining Turing Test gap.

Keywords Turing Test, Cyc, Artificial Intelligence, ontology, common sense knowledge, translogical reasoning

In this chapter, we will unravel four issues that are often tangled up together:

– How *could* we get a machine to pass the Turing Test?
– What *should* we get the machine to do (or not do)?
– What *have* we done so far with Cyc?
– What *will* we do to close the gap?

Cycorp

R. Epstein et al. (eds.), *Parsing the Turing Test.*

16.1 How Could we get a Machine to Pass the Turing Test?

How do we get a machine Turing-ready? By “machine” here we really are referring to the software, not the hardware.[1] There are two obvious functionalities the program needs to exhibit, hence two obvious tasks for the programmers to worry about:

Task 1: It needs to be able to communicate via natural language (e.g., English).
Task 2: It needs to know enough not to make a complete idiot of itself.

Both of these turn out to require much more effort on the programmers’ part than you might think, much more effort than the leading scientists *did* think in 1956 when they set out to work on these problems. That, in turn, is due to the enormous amount of *common sense* knowledge that is required to understand even seemingly simple natural language sentences, and to answer even seemingly innocuous queries.

Answering ad hoc queries requires the ability to reason, to produce an answer by logically combining several pieces of information that one knows. As we will see, getting a computer to do that combining – deduction – turns out to not be as complicated as you might imagine, thanks to some relatively recent advances at the intersection of mathematics, philosophy, and computer science. A big engineering task, yes, but not one that requires yet-to-be-made scientific breakthroughs.

But there is a third task, less obvious than the first two tasks, which must be done to give the machine the ability to mimic human beings well enough to pass the Turing Test:

Task 3: It needs to behave illogically when – and as – a human would.

This does *not* mean giving random answers, it means reasoning translogically to come up with an answer which is incorrect *but which most people would give*. The simplest sort of translogical behavior is *forgetting*. For example, almost everyone’s answer to a query of the form “What day of the week did < *some past date* > fall on?” is “I don’t know,” unless the date is no more than a few days earlier than today’s date. If the Turing Test interrogator asks that query of random dates years earlier, and the respondent gives the correct answer in each case, that is a tip-off that it is not human. If they ask that question about, e.g., yesterday’s date and tomorrow’s date and the respondent gives the wrong answers, it is also a tip-off that the respondent is not human.

[1] For any given program, some level of hardware capability is required (cpu speed, size and speed of random access memory, secondary storage, etc.) but any hardware above that level of performance is effectively unnecessary. Hardware is already so good, so human-scale, compared with software, that it is not the bottleneck today. Several sources act as though it is, but here is a simple counterargument: if hardware is the bottleneck, then show me a 100% human-level AI – heck, even just a 100% domain-independent human-level speech understanding system – that runs in 10x or 1,000x or *any*-x real-time – which one cannot point to today.

Sometimes there is no *single* human-like wrong answer to give, but there are some obviously inhuman incorrect wrong answers. For example, if the interrogator poses a series of queries where each involves multiplying a pair of 4-digit numbers together, by hand, a human being would not provide his/her answers very quickly, and would probably get some such queries wrong. However – and this is the key point here – a Turing-smart program should not give a *random* answer when it is pretending to get a multiplication problem wrong. For example, if asked "What is 4804×5931," the program should not reply with 0, 42, 10^{23}, a negative number, a number with a nonzero fractional part, etc.

We will discuss several more examples of such translogical reasoning (illogical and incorrect, yet highly nonrandom and often quite predictable) in Section 16.3, which is about Task 3. First, let us examine Task 1 and Task 2.

16.1.1 Conversing in Natural Language

The great advances in physics, over the past few centuries and especially the 20th , have come about through the application of mathematics: Kepler, Newton, Maxwell, Einstein, Schroedinger and others discovered equations that unlock the mysteries of the physical universe. This has led many fields to emulate the *form* of that paradigm, trying to formalize their domain and thereby get a handle on it. This has succeeded in a few cases, but in many cases, it has been downright harmful. For example, cognitive scientists studying humans memorizing nonsense triads ("BVX") because it is easier to run statistics on that than, e.g., attempting to study humans learning in situ. To cite one more example, the great psychologist Jean Piaget did some of the most spectacular *empirical* work in his field during the first half of his professional life, but then was bitten by the "find the Maxwell's equations of thought" bug and – from my point of view – squandered the second half of his professional life chasing that phantom.

We continue to make those mistakes today, in fields such as linguistics. Here is an example – an all too real and too common example – that we see in our job applicants at Cycorp:

Interviewer: Consider the sentence "The horse was led into the barn while its head was still wet." What is the antecedent of "it," and why?
Linguist#1: " 'It' refers to the horse. Because it is the subject of the sentence."
Interviewer: "What if I change the word *head* to *roof*?"
Linguist#1: "Then 'it' would be referring to the barn. But that is an exception."
Interviewer: Consider the sentence "The horse was led into the barn while its roof was still wet." What is the antecedent of "it," and why?
Linguist#2: " 'It' refers to the barn. Because it is the closest noun."
Interviewer: "What if I change the word *roof* to *head?*"
Linguist#2: "Then 'it' would be referring to the horse. But that is an exception."

Interviewer to Linguist#1: "Here is what Linguist#2 said, and why. What do you think?"
Linguist#1: "They're just wrong; my rule is the fundamental one."
Interviewer to Linguist#2: "Here is what Linguist#1 said, and why. What do you think?"
Linguist#2: "They're just wrong; my rule is the fundamental one."
6-year-old child: "Horses have heads, barns don't. And barns have roofs, horses don't."

For our money, the 6 year old child is correct. We do not hire many 6 year old children, but then again we don't hire Linguist#1 or #2 either. Syntax and semantics may help to constrain and positively bias, but pragmatic world knowledge trumps them both.

Here is another example. Consider the sentence "Sue and Jane are *sisters*." I *might* mean that those two people each have a sibling, but are not related in any way to each other. That would be particularly cruel and misleading since so many people *have* siblings but so few people are each other's sibling. Surely I mean that they are each other's sister. Now consider a similar sentence: "Sue and Anne are *mothers*." You know I do not mean that they are each other's mothers. How? You drew on your knowledge about biological reproduction in the real world, to disambiguate the sentence's meaning.

As another example, consider Terry Winograd's old chestnut:

"The police arrested the demonstrators because they feared violence." versus

"The police arrested the demonstrators because they advocated violence."

In one case the "they" refers to the police, in the other case "they" refers to the demonstrators. This draws on your model of police goals and tactics, demonstrator motives and tactics, etc., not linguistic syntax, to resolve.

Things get even more complicated when you introduce metaphor and analogy, when you permit elliptical references, when the discourse is extended across numerous back and forth interchanges in a single conversation, and when it has some rich context in which it occurs (e.g., a job interview versus a loan application versus a dinner out with friends).

The point of these examples is that understanding natural language *at a human level of performance* requires a huge substrate of general and specific knowledge. The speaker or writer draws on that to shorten their utterance, *encoding* it in a terse form; the listener or reader must then draw on their own version of that body of knowledge, to *decode* the utterance, to resolve the ambiguities introduced by that terseness (pronouns, ambiguous words, the intended scope of a prepositional phrase, the order of quantifiers, recognizing sarcasm, etc.)

One final example, involving the order of quantifiers, should suffice to make the point. In the sentence "Every American has a president," we mean they all have the same president, but in "Every American has a mother," we obviously do not mean they all have the same mother.

So the interesting thing about Task 1, getting the system to converse in English, is that it turns out mostly to depend on Task 2: getting the system to know one heck of a lot of things about the world, and be able to use them (logically combine them when and as needed). As linguists will correctly tell you, there also has to be a lexicon of word meanings, idioms, and such, and a way of grammatically analyzing and synthesizing sentences built out of those words, but those are relatively straightforward subtasks compared to the daunting challenge of capturing the pragmatics of the real world, the millions of facts that people draw on, unconsciously, to understand sentences that would otherwise be ambiguous, metaphorical, or meaningless.

16.1.2 Having Lots of Factual Knowledge (and Being Able to Combine it Using Logic and Arithmetic)

What does it mean for a computer program to *have* some piece of information? It means that the knowledge is represented in some formalism; more concretely, that it is encoded into some data structure. To say that a program has a piece of *knowledge*, versus just a piece of information, is to say that the program can use that knowledge when appropriate, that the program will act as though it knew that piece of knowledge. So: knowing = having information + the ability to use it.

For example, suppose that a company stores employee data online, including birth date, hiring date, emergency contact, educational history, project history, etc. To say the program stores the *information*, it needs to be able to provide that data back given the right questions. To say that it has that *knowledge*, it should be able to use that information in inferences, in ways that were not foreseen when it was entered: to catch errors (such as anyone's birth date being later than hiring date, listing oneself as one's own emergency contact), to send birthday greetings, to decide which questions someone ought to be able to answer, to infer that certain people know each other on sight, etc.

Some of the factual knowledge that a Turing-capable machine must possess is widely known by "almost everyone" (i.e., by the typical person who might be chosen to be a respondent on the test: someone alive, not an infant, not illiterate (if the test is administered in written form), not insane, etc.). Some of that widely held factual knowledge is timeless (what is 2 + 3?), some is time-varying (who is the current President of the USA? Is it day or night now?). Some of the knowledge that the respondent should possess is idiosyncratic to that individual but something that every person ought to be able to answer about himself or herself – this includes objective facts (What is your mother's first name?) and subjective opinions (What is your favorite food?). Entire microcosms of knowledge may be required based on the respondent's answers to earlier questions – e.g., if the program claims to be an ER doctor, it had better be familiar with medical jargon, diagnosis, the sights and sounds and smells of an emergency room, etc.; if the program claims to like *Star Wars* movies, it had better be able to discuss them and answer specific questions about their content like "Who is Yoda?". Some of the factual knowledge involves human limitations and capabilities, rationales, emotions, etc. For example, "John robbed a 7–11 yesterday while Nancy was working there at the cash register. Which of them first knew that the robbery was going to happen? Was Nancy more likely to feel sleepy or afraid during the robbery?"

How can we get the machine to produce the logical inferences that you or I would, from a set of assertions? Thanks to work by logicians throughout the 20th century, we know (we can demonstrate) that we can do this by representing that information as sentences in a formal language such as predicate calculus – first- or second-order logic – and then applying mechanical theorem proving algorithms to those assertions. For instance, if we add to the Cyc program's knowledge base:

```
(ForAll x (implies (isa x ViolentCrime)
(ForAll y (implies (victim x y)
(During x (feelsEmotion y Fear))))))
```

then that expresses the general rule that vicims of violent crimes feel fear during the commission of the crime. If we further assert that:

```
(ForAll x
(ForAll y
(implies (is-a x Robbery)
(is-a x ViolentCrime) ) ) )
```

and

```
(is-a Robbery8092 Robbery)
(perpetrator Robbery8092 John)
(victim Robbery8092 Nancy),
```

the set of assertions tells the system that robberies are violent crimes, and that a certain robbery occurred where John robbed Nancy. We can now *ask* the Cyc system what Nancy felt during that robbery:

```
(During Robbery8092 (feelsEmotion Nancy ?feeling) )
```

and the system will logically deduce the answer ?feeling = Fear. If we explicitly ask if Nancy felt fear during the robbery, the system could reply "Yes." If we explicitly ask if Nancy felt sleepy during the robbery, with only the limited knowledge above, Cyc would have to say "I don't know." There may be another rule, which says feeling fear usually forestalls one from feeling sleepy. If so, then the answer to "Did Nancy feel sleepy during the robbery?" would change from "I don't know" to "No."

So accomplishing this Task #2 – getting the system to have and use lots of knowledge – boils down to writing a large number of "facts" about the world in a formal logical language. We put "facts" in quotes because many of these are just rules of thumb, likely to be true but not absolutely true – such as people feeling afraid but not sleepy while they are being robbed at gunpoint. One *could* feel sleepy, but it would be pretty surprising.

We want (for ourselves, for others, for our program Cyc) to have a good *ontology* of the world: make useful metaphysical distinctions that carve the world up into meaningful categories. That way, we can state rules at a very general level that will apply to many situations. For instance, consider the rule about feeling fear while the victim of a violent crime; we did not have to state that same sort of rule over and over again for robbery, for rape, for battering, for stabbing, etc. – let alone for *each* such event, each particular robbery, etc. On the other hand, if we made up a category like "events whose performer's first name starts with the letter 'J'" then there just are not very many things to say about the members of that category, which incidentally includes the robbery of Nancy by John.

To sum up, we need to carve the world up intelligently into an ontology of useful distinctions. That gives us a vocabulary of concepts, representing objects, events, relationships, etc. To build our intelligent program, using that vocabulary, we need to write down – as generally as we can without sacrificing truthfulness – millions of "facts" (including rules of thumb) about the world, the things that the average respondent in a Turing Test knows. If we have picked a good ontology, there may only be millions rather than trillions of such "axioms" that need to be stated. And if we write those down in predicate calculus, we can bring a theorem-prover to bear on that huge assemblage of formal assertions, to crank through them mechanically to answer trillions of the same sort of questions that a human Turing Test respondent would.[2]

Carrying out that large-scale knowledge-base construction program is neither trivial nor impossible. In 1983, a group of us (myself, Alan Kay, and Marvin Minsky) got together to estimate the work required, and came up with a figure on the order of 1 person-millennium of time. That sounds like a lot, but to put it in perspective, it is less than the effort that was required to build the Great Pyramid. We have been working on this task for 22 years now, and have spent over 750 person-years of effort on it so far – and we will report the results of that enterprise in Section 16.3.

16.1.3 Translogical Inference: When Good Human Brains Do Bad Things (and Turing Test Programs Better Do So, Too)

We have already mentioned a few examples where humans act imperfectly, erroneously, yet in more or less *predictable* ways. If our would-be Turing Test machine does not duplicate these answers, these translogical reasoning errors, then it will be easy for the interrogator to identify it as the nonhuman respondent.

Let us start by reviewing the trivial ones we discussed above, and adding a few more:

- T1: forgetting. What day of the week is <*some date 8 years ago*>?
- T2: slow at doing arithmetic. What is 4302 × 5394?
- T3: errorful at doing arithmetic. But not all erroneous answers are equally likely. In the case of the given problem, 20412188 is a reasonable wrong answer, but

[2] But, as we will see in the next section, still not the full range of queries that a human Turing Test respondent could correctly refer to, unless we write down translogical axioms too.

zero is not, nor is –20412188, nor 20412188.01, nor 20412000, nor pi, nor 4000, etc.

– T4: slow at typing. At the January 2000 Loebner competition, some programs betrayed their nonhuman natures right away by typing at superhuman speed.
– T5: errorful at typing. But not all mistypings are equally likely, given a QWERTY keyboard layout and the anatomy and functioning of human hands.

Many of the rest of what I am calling translogical behaviors reflect phenomena by which people can be *predictably* led astray, led to give a certain answer, generally to give a certain wrong answer, to a query. These are a little more complicated, and unfamiliar to most readers, so the examples I give for each one are a bit longer as well.

– T6: Regression to the mean (see Fig. 16.1). Suppose you reward your child every time they get a good grade and punish them every time they get a bad grade. Over time, you try to learn which is more effective. Suppose that in reality *neither* reward nor punishment has any effect (if you have kids, you probably already knew this.) But very high or very low grades are usually just randomly distributed about the mean – the place the child is really at anyway, say a B-level. So after getting an A on one paper, they are very likely to get a worse grade next time; and after getting a D on one paper, they are very likely to get a better grade next time. Remember that all this is *independent* of your reward and punishment scheme, which just rolls right off their backs and has no effect on them at all. But from your point of view, when you reward the kid, they do usually worse next time (get a B-), and when you punish them, they usually do better next time (get a B-). So it *appears* to you that rewards do not work but punishments do! This phenomenon – regression to the mean – has therefore been responsible for untold human suffering throughout history, misleading people into believing that punishment is more effective than reward even in situations where it is not. To utilize this human foible during a Turing Test, the savvy interrogator would give the respondent a series of a pupil's test scores over a school year, and actions that were taken in response to each one (reward versus punish), and the interrogator would then ask the respondent to decide whether reward or punishment (or neither) was more effective, with this pupil. A human would say "Punishment is clearly more effective", where a logical but not translogical machine would "err" by giving the correct statistical answer of "neither."
– T7: Mistaking the importance of the order in which a *choice* is allowed. Behind one of 3 doors (A, B, C) is a $100 bill. You pick a door (e.g., you pick door C). I am the emcee, and I already know where the prize is, so no matter which door you pick, I can open another door (e.g., door A) and show you that the prize is not behind that one. I do so, then give you the chance to change your guess to the third door – not the one you originally picked, and not the one I just opened. Should you change your guess? Most people say it does not matter, in fact most people stick with their original guess, but statisticians and smart programs know that the answer is a resounding Yes, changing your guess increases your chance of winning from one third to two thirds. To see why, imagine there are a million doors, and

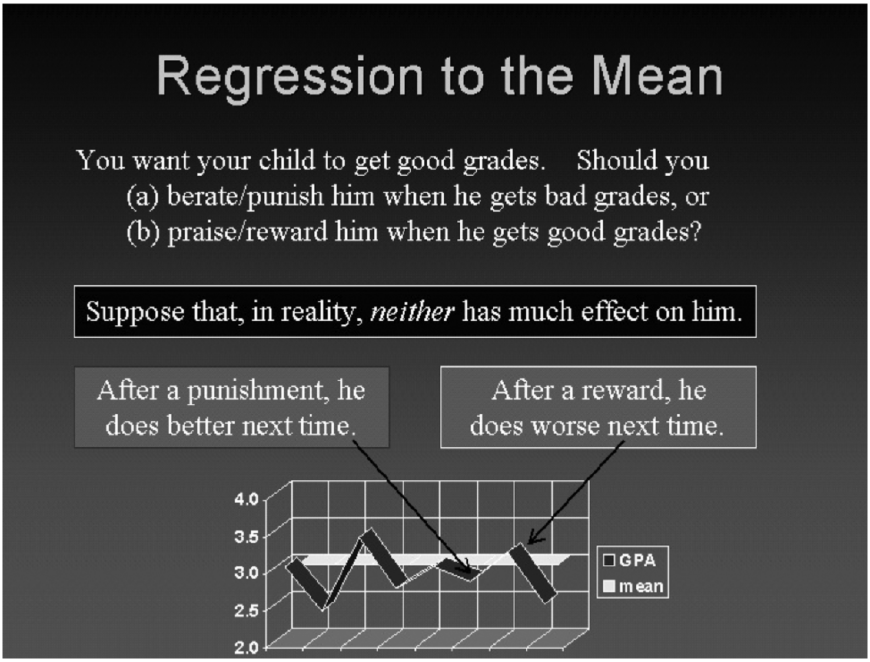

Fig. 16.1 Regression to the Mean

after you pick a door (say door#4001) I open 999,998 of them and show you they are empty, then give you the chance to change your original guess. Changing in that case, raises your odds of winning from 0.000001 to 0.999999. One in a million times, your original guess would have been the right one. All the other times, changing to the one unopened door would be the logical decision. But not the *human* decision.

- T8: Sunk Cost. Consider the following two situations
- You invested $100k in a company; the stock is now only worth $10k. You hear a rumor that it is about to go out of business. Would you keep your money there or move it?
- You have $10k to invest. You hear about a company whose stock price has declined 90% and that is about to go out of business. Would you invest your money in it?
- Even though situation A is mathematically equivalent to situation B, 85% of people surveyed said in case (A) that they would leave their money where it was (maybe the company would "turn around" and they would not have been so foolish after all), and in case (B) an even greater number said that they would not do the investment (only a fool would invest in a company in that downward spiral situation, at least without further information). To pass the Turing Test, a program would have to be able to (pretend to) fall prey to this illogical "sunk cost" phenomenon, this hysteresis effect of throwing good money after bad.

– T9: Attraction Effect. What should an engagement ring cost, typically? We find two values such that 50% of the people surveyed pick each value, say $500 and $5,000. Now by adding a third extreme choice *that no one ever selects*, we can bias the results. If we add a $5 choice, no one will pick it, but most folks will now pick $500. In instead we add a $500,000 choice, no one will pick it, but most folks will now pick $5,000. A Turing Test program would have to (pretend to) fall prey to this illogical behavior, being swayed to pick "the middle choice."
– T10: Conjunction Fallacy. When asked to rank the relative likelihood of the following six statements, most people rank #5 as more likely than #2. Yet simple probability tells us that the probability p (A^B) must be *less* than p(A). A Turing Test program, to pass for human, would have to (pretend to) fall prey to this illogical trap, in cases where A and B are at least moderately highly correlated (such as this one).

  1. Mr. F smokes more than 1 cigarette per day on average.
  2. Mr. F has had one or more heart attacks.
  3. Mr. F had a flu shot this year.
  4. Mr. F eats red meat at least once per week.
  5. Mr. F has had one or more heart attacks and he is over 55 years old.
  6. Mr. F never flosses his teeth.

– T11: Omission Bias. Suppose that there is an epidemic E that has been killing 1/10,000 children at random, and it is about to hit your city. The only cure is 100% effective, but it has side effects that kill 1/11,000 of the children who take it, *but never any of the children who would have been killed by E.* The question is, do you have your child immunized? Logically, the answer should be Yes, but most people say No. They would rather lower their child's overall chance of survival (from 10,999 in 11,000 down to 9,999 in 10,000), through their *inaction*, than end up responsible for their child's death by their direct *action*. A Turing Test program would have to make that translogical choice, too.
– T12: Reflection Framing Effect. Tversky and Kahneman (1983) have documented many cases where most people will make one choice when the alternatives are framed in terms of risk of failure, and a different choice when a set of mathematically identical alternatives are expressed in terms of opportunity for reward. Especially if costing human lives is the risk, and saving human lives is the reward.
– T13: Multiple Alternatives Dissonance. Take a situation where most people choose alternative A or B, given a choice of three options: A, B, or "defer the decision." Add in several new alternatives. Now most people will choose "defer the decision," even if each of the alternatives added is highly inferior to *both* A and B.
– T14: Hindsight Effect. Take a jury trial, present the facts, indicate the verdict, and say "and it turns out that the person was innocent after all." Most people will say that they would have voted Not Guilty. Give the same case to people but with

the tag line “and it turns out that the person was guilty after all.” Now most people will say that they would have voted Guilty. It is just human nature, to think we would have made the right decision in each case, but any Turing Test program will have to exhibit the same sort of self-delusions.

There are many more such predictable irrationalities that people exhibit, and that any program would have to exhibit, to pass the Turing Test. In brief, a few of the most familiar of these irrationalities that I have not listed already above involve:

– Gambling (succumbing to the urge to play the lottery; the well-known voting paradoxes; committing crimes – and attempting superhuman heroic acts – that one could not really hope to get away with, etc.)
– Conspiracy theories (UFOs, Illuminati, The Government, by Race X, by Religion Y, various forms of paranoia, etc.)
– Bending too easily to authoritarianism (being swayed by the media, being starry-eyed in dealing with someone because of their celebrity, believing the current scientific/medical/… theories to all be true, etc.)
– Centrism (thinking the last movie you saw was the best/worst ever, mirror image ascribing your knowledge and values to someone else, believing your race/religion/city/team… is best, anachronistic beliefs such as thinking that the ancient Greeks used bicycles, pens, or umbrellas in the rain; and so on)
– Irrational phobias and philias. For example, Americans have a much greater fear of being killed in a terrorist bombing event than a (nonterrorist-related) plane crash, and a plane crash than an auto crash, even though the frequencies of the three events are, respectively, one in 20 million per year, one in 2 million per year, and one in 5,000 per year. They act on those fears – when planning trips, etc. – even when they intellectually *know* the statistical relative likelihoods. Any Turing Test program would have to react the same translogical way, or else betray its nonhumanness to the interrogator.
– Innumeracy. In the book by the same name, John Paulos (1988) cites many problems that most humans choose the wrong answer to, even given a small set of very different numerical choices – e.g., how fast does a typical person’s hair grow, in miles per hour? How many people do there need to be, in one room, before there is a 50–50 chance that a pair of them have the same birthday? (My favorite, and very human, answer to the former was “Hair doesn’t grow in miles per hour.”)
– Distrust of Nonconstructive Arguments. People are highly swayed by constructive arguments, and distrustful of nonconstructive ones. For example, you have millions of hairs on your head, but there are billions of nonbald people on Earth. Yet when asked “do you think that there are two nonbald people on Earth that have exactly the same number of hairs on their head?”, most people will only say *probably* so, not definitely so.
– Local–Global Errors (see Fig. 16.2). When UCB had to comply with affirmative action requirements, each department made sure that the Yes: No ratio for female applicants was larger than the fraction of Yes: No decisions for male applicants. Yet, the university-wide fraction for females remained lower than for

DEP'T.	# MALES APPLYING	# FEMALES APPLYING	# MALES ACCEPTED	# FEMALES ACCEPTED	% MALES ACCEPTED	% FEMALES ACCEPTED
Chemistry	2400	135	1150	81	47.9%	60.0%
Art History	291	1700	34	470	11.7%	27.6%
TOTALS	2691	1835	1184	551	44%	30%

Fig. 16.2 Local global errors

males! Most people thought this could not happen, but it did (because different departments had very different fractions of male versus female applicants, and very different acceptance percentages). Here is an example of how it happened – which I think is necessary because you, the reader, probably *still* do not really believe that such a thing could happen. For simplicity, suppose that there are only two different departments in the university, Chemistry and Art History. The acceptance rate in the Chemistry department is 48% for men and 60% for women; the acceptance rate in the Art History department is 12% for men and 28% for women. So in each department, they are accepting a higher percentage of female applicants to male applicants. Yet, for the two departments combined, the acceptance rates are higher for men than for women – namely 44% for men and 30% for women.

We want our Turing-smart program to know the right answer, to whether such a thing could happen or not, and yet give the wrong answer, the human answer, that surely it is impossible.

The list could go on, but by now, the message has hopefully come across clearly: there are lots of queries that the interrogator can ask the Turing Test respondent, where the *human* response is both predicable and dead wrong. If the program does not mimic those predictable wrong answers, it will be easy for a clever interrogator to separate the human responders from the nonhuman ones.

16.2 What Should We Get It To Do (or Not Do)?

Should we try to get a program to pass the Turing Test? This is not so much a good versus evil question as one of simple economics. While it might be entertaining to have a machine that forced itself to do arithmetic slowly with sometimes-erroneous results, would we really want such a machine? To pass the Turing Test, the answer is “yes indeed,” but for all the other myriad purposes and applications, the answer is generally "of course not!" Some exceptions might be to diagnose and treat mental illness, to act as an empathetic "companion" for users of online applications (though there, the application is, in effect, passing the Turing Test), etc. – in situations like those, it would be useful for the program to at least know about the various translogical errors catalogued in Section 16.1.3, so it could diagnose human beings' errors and so it could faithfully simulate human-like erroneous behavior.

In Fig. 16.3, the solid oval marks the strategy to pass the Turing Test, and the dashed ovak marks what would be maximally useful to humans.

Working on quadrant 2 is a red herring, even though it helps programs pass the Turing Test: trying to dumb computer program performance down, so that it takes as long to add two numbers as a person does, so that it makes the same sorts of mistakes that humans do, etc.

On the other hand, working on quadrant 4 is a useful enterprise, even though it will not help the program pass the Turing Test. It leads to appliances that amplify the human mind, that amplify our intelligence, much as physical and medical appliances amplify our muscles and our immune system, respectively.

This divergence, between the solid and dashed ellipses is partly responsible for the dismal behavior of Turing Test contenders, over the years, up through the present day. Because the quadrant 2 work is largely a waste, and the quadrant 4 work is largely cost-effective, our programs end up acting very obviously different from human beings. We work on the dashed ellipse, not the solid one. And as one of those who make that same choice over and over again, I clearly agree that it is the right decision for most researchers to make.

Political Correctness Has Crippled the Turing Test. Part of the problem is that the Turing Test has mutated since its statement in *Mind* in 1950. For political correctness reasons, the original test (asking the interrogator to decide which of 2 respondents is the man, and which is the woman) has been replaced by a much harder test for machines to pass (asking the interrogator to decide which of 2 respondents is the person and which is the computer.)

In the original test, the man – and the computer – were pretending to be the woman. The interrogator was told that in the next room were a man and a woman, both of whom would pretend, over the teletype, to be the woman, and the interrogator had to choose which actually was the woman. So all their questions dealt with gender differences (what sizes and colors that stockings come in, what is sold in

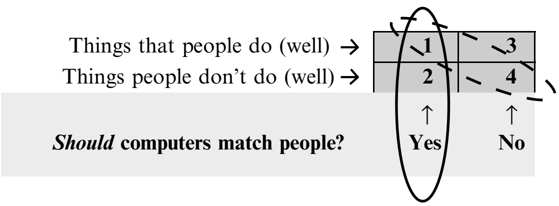

Fig. 16.3 The summary of the four quadrants

vending machines in a women's restroom, etc.), not human versus machine differences. Let us say that a typical man fooled a typical interrogator (half of whom were men, and half of whom were women) 30% of the time. That is, suppose that 70% of the time, it was clear which was the woman. Suppose that this is the average, after doing tens of thousands of trials, with different men and women responders, and different human interrogators.

"Then", Turing said, "replace the man by a computer." The interrogator is *not* to be told that she/he is talking to a computer. The interrogator gets exactly the same instructions: you will be talking to a man and a woman, decide which is which. Period. The task of the program, then, is to convince the interrogator that it is a woman, not a man. The test for intelligence, therefore, is whether or not the program fools the interrogator about as often as the man fooled the interrogator, into identifying the wrong respondent as the woman – that is, on average, 30% of the time, after running many many trials with different female responders and different interrogators. The conclusion, if such a program could be exhibited, would be that the program would then be as smart as a typical person – well, at least as smart as a typical man. A modified program could do the test in the opposite direction, with everyone (the male respondents, the female respondents, and the program all pretending to be men).

In the original Turing Test, the phenomena such as those listed in Section 16.1.3 would be irrelevant, since they are just as likely to be evinced by men as by women, hence the interrogator would not even bother posing such a query to the respondents.

To sum up, then, I have tried to make two points in this section:

(a) The new "politically correct" version of the Turing Test makes the test vastly more difficult, because the interrogator is looking for human versus nonhuman differences, instead of man versus woman differences. Turing never intended this.
(b) Passing the Turing Test has *some* elements of usefulness, since having a true humanlike (hence human-level intelligence) AI would make almost every application more robust, would amplify our own intelligence. But overall, we want our computer assistant to be as competent and smart as possible, not go out of its way to err in the ways that human beings do. We do not buy pocket calculators that add about as slowly and errorfully as we do, we do not buy cars that go about as fast as horses (or, even worse, ourselves), and so on.

To communicate with a toddler, educated adults can force themselves to speak slowly, using simple words, etc. Ultimately, focusing on quadrants 1 and 4 from Fig. 16.3, true superhuman AI will be developed. Once that happens, it could deign to *pretend* to type slowly, misspell words, forget, and fall prey to all the various translogical irrationalities that human beings exhibit. For a while, there would still be some areas where it was below the human norm: lacking a powerful parallel image-processing "retina," lacking a human-sounding voice, lacking any detailed memories about its childhood, family, job, the way a beer tastes, the way air conditioning feels, etc. Long before all those deficiencies are rectified, however, the true AI system would be able to talk its way around each of them (claiming blindness, throat cancer, amnesia, or inventing a coherent false story of its life.)

So even if we as AI researchers eschew the Turing Test, and work on giving our machines quadrant-4 superhuman capabilities instead of quadrant-2 just-human-level capabilities, in the long run we will be building a machine capable of stooping to conquer the Turing Test after all.

16.3 What Have We Done So Far with Cyc?

In 1984, I started the Cyc project at MCC, the first US computer-related consortium to form. Located in Austin, Texas, MCC's goal was to beat the Japanese Fifth Generation Computer effort, by pursuing decade-long high-risk high-payoff R&D. Cyc's goal, in particular, was to codify human consensus reality knowledge – the things that the average person knows about the world – in a machine-manipulable form.

The goal was most decidedly *not* to pass the Turing Test (quadrants 1 and 2) but rather to produce the most useful possible sort of intelligent machine assistant (quadrants 1 and 4, above). Listed in Fig. 16.4 are 20 applications we had in mind, for Cyc, in 1984 – all of which, interestingly, are still quite promising – after which are listed 5 more recent application ideas for Cyc.

We planned Cyc to occur in three decade-long phases, of which we are now engaged in some combination of phases 2 and 3. Phase 1: Prime the knowledge pump, by building a core ontology and associated knowledge base, basic reasoning modules, interfaces, etc. Phase 2: Using that knowledge, power a mode of interaction and KB growth that is based on natural language understanding (one-shot reading of static text, and/or interactive back-and-forth clarification dialogue with a live human being acting as a "tutor"). Phase 3: When the KB is large enough, have the system take an increasingly active role in its own continuing growth and application, including driving the dialogues (per phase 2) and carrying out automatic

1984App.I dea	How Cyc Could Be Used (more or less ***today***)
1. Text Understanding and Document Preparation 2.Speech Understanding 3.Translation / Generation	Use context and pragmatic constraints to disambiguate pronouns, word senses (text) and slurs and prosody (speech), metaphors, irony, the order and scope of quantifiers, etc., and inject such things into generated text/speech. Smarter checking of spelling ("bear" => "bare"), grammar, style, and *content* (e.g., write "Below, I discuss X..." but never do).
4.Expert Systems	Less brittle behavior, because of the ability to fall back on

Fig. 16.4 (continued)

1984App.Idea	How Cyc Could Be Used (more or less *today*)
5.Training Simulations	increasingly general knowledge and the ability to analogize to specific but far-flung knowledge. More open-ended interactions with simulated characters, groups, and objects.
6. Games	
7.Online Commerce	Accrete models of the users to help match buyers/sellers and donors/donees, to help decide which colleges/doctors/... to consider, and to decide which products someone would want (and when, and why, and how best to sell them).
8.Online Advice Services	
9.Directed Marketing	
10.Clean/ Integrate DBs	Represent the meaning of the relations/fields/columns/... as Cyc assertions, thus using Cyc as an interlingua to virtually integrate them, to find inconsistencies among them, and dynamically add/kill/guess/reorder user interface menu choices, display modes, etc.
11....and Spreadsheets	
12."Active" Menus/Forms	
13.Helpful Encyclopedia	Represent the actual content of online articles, captions, documents (e.g., court rulings), and user queries; then use deduction to find matches and to answer questions even if that requires combining multiple pieces of information logically and/or arithmetically.
14.Question-answering	
15.Search for docs/photos	
16.Scenario Generation	Use Cyc to spawn plausible chains of action/reaction chains (scenarios) using that data.
17. Appliances	VCR's/cars/coffeepots/... that converse with their users (and learn their likes, habits, etc.)
18.Mental Prostheses / Mental Amplifiers	Help forgetful people remember where things are, who is related to who, etc. Stimulate creativity, offload tasks/decisions, catch interpersonal misunderstandings, etc.
19.Automated Discovery	Use Cyc constraints to guide the process of hypothesis formation, experiment design, data collection, evaluation of results, judging their significance, introduction of new terms.
20.Image Compression	Store (and transmit) terse semantic descriptions of situations instead of pixels.
2002App.Idea	How Cyc Could Be Used (more or less *today*)

Fig. 16.4 (continued)

1984App.I dea	How Cyc Could Be Used (more or less *today*)
21. Email	Prioritize, summarize, add explanatory footnotes color-coded by type (explanation of an acronym, link to proper noun's home page, link to something it seems to contradict), increasingly auto-forward (to appropriate other recipients) and auto-answer messages.
22.Network Security	Generate plausible network attack scenarios for various types of perpetrators and goals, based on a model [machines on the network, programs each has installed/running, open ports, ...]; propose defenses; do "What-If...?" reasoning on edited model to try them out.
23.Ontologic al Engineering	Actively help a user (a non-logician, non-programmer,...) to add new knowledge to Cyc, to refine and organize knowledge, and to articulate and refine their questions as well.
24. Wireless	Mediate (using common sense, dialogue, etc.) to help someone using a wireless device (with little/no screen real estate) who is attempting to navigate CRT-designed web pages.
25.Thesauru s Management	Semi-automate the process of aligning massive technical-term lists, such as the numerous unaligned 300k-term thesauri used by pharmaceutical companies.

Fig. 16.4 Twenty-five applications, for Cyc, in 1984

learning by discovery through a combination of introspection and gathering external online data. Now that phase 2 and 3 are underway, most of the 25 applications could be started finally.

Along the way, we encountered several hurdles, and took an engineering approach (rather than a scientific or academic approach) to overcoming each one. For example, to represent pieces of time, and carry out temporal reasoning, we did not try to find a single universal theory of time, we developed instead a series of partial solutions whose union covers the most common sorts of cases that crop up in daily personal and business life.

Hurdle 1. One lesson we learned was that our language had to be expressive enough to represent all the distinctions that human beings find useful. This means two things: the vocabulary had to keep growing, and the allowable syntax had to keep expanding. By now, we have been led to a vocabulary of several hundred thousand general terms (of which over fifteen thousand are types of relationships, such as employed-by), and a syntax that allows for quantifiers, nested quantifiers, second-order quantifying over predicates (e.g., "What relationships hold between ducks and water?") and over propositions, modal operators, negation, tenses, disjunction, nested modals (e.g., "Israel believes that Saudi Arabia does not want the U.S. to fear that Al Qaida will try to sabotage…").

Hurdle 2. Using such a general representation language threatened to make inference way too slow, and to overcome that we had to separate the epistemological problem – what should the system know – from the heuristic problem – how can the system reason efficiently about what it knows. We represent each piece of knowledge in Cyc in two ways: (1) cleanly and declaratively in predicate calculus at the Epistemological Level (EL), and (2) efficiently using special-purpose data structures and algorithms at the Heuristic Level (HL). By now, we have over 720 special-purpose HL reasoning modules.

Hurdle 3. We had to give up on absolute truth and falsity, and rely on default-truth, and default reasoning. Very few things are monotonically true about the real world. For example, whether you believe in evolution or in creationism, there have only been a finite number of people on Earth, so even a seemingly exception-free statement like "Each person has a mother who is a person" is only default-true. Even worse, there are often pro- and con-arguments for and against a proposition, so a statement like "Bill Clinton was a good president" is not true or false per se, but rather has several pro- and con-arguments, which need to be kept around and weighed to come up with – if forced – an overall value.

Hurdle 4. Numeric certainty factors seemed like a good idea at the time. But it turns out that if you ask someone to make up a three-decimal-digit probability for a statement, like "People know the first name of their boss," they will *make up* a number – say 0.950. The trouble arises because any two numbers are commensurable, and someone else, possibly years and continents away, might be *making up* a number like 0.966 for the probability that "People know the middle name of their boss." Now, inadvertently, these two individuals have told Cyc that it is more likely that someone knows the middle name of their boss than the first name of their boss – something that neither of the knowledge-enterers intended. In lieu of numbers, we have relative likelihood assertions; so we tell Cyc that both statements are default-true, but statement 1 is *more Likely Than* statement 2.

Hurdle 5. We have to give up on purging redundancy from the KB. This turns out to be a good thing after all, since we can intentionally store frequently used but redundant assertions in the KB, and speed up (rather than slow down) overall reasoning speed. The HL modules themselves, e.g., can be thought of as fully redundant but useful.

Hurdle 6. We have to give up on completeness. Even the number of one-step deductions that Cyc can do is infinite. The full deductive closure of the knowledge base is even more infinite, almost unimaginably infinite. So reasoning is done in a resource-limited best-first fashion. When Cyc is told something new, it works forward from that, spinning out consequences that may prove useful to be cached in the future. When it is asked something, it works backward, trying to find an answer, until it does so or it gives up.

Hurdle 7. We have to give up on consistency. This was the hardest lesson of all, since it goes against everything we ever learned in school (from False, everything follows) and from watching Captain Kirk deal with would-be intelligent computers (which would invariably freeze up or explode when caught in an inconsistency). In place of global KB consistency, we have adopted the notion of *local consistency*. The KB is carved into a set of contexts or microtheories, much as the earth's continents rest on a set of rigid tectonic plates. Each context is internally consistent, but there can be inconsistencies across contexts. The contexts are first class objects in Cyc's vocabulary, and are related to each other by a dozen types of hierarchical relations: more-general-than, earlier-than, located-above, more-certain-than, etc. There are a set of *articulation axioms* that explicitly relate each context (its content, assumptions, etc.) to its neighbors.

Contexts can capture what is true at an interval or point in time (e.g., the 1990s), true in a region of space (e.g., the USA), what someone believes to be true, what is true at a certain level of granularity, what is true about a certain topic, etc. These can be considered *facets* or *dimensions* of context-space.

Cyc reasons within a context, drawing in ("lifting") assertions from farther and farther-flung contexts, using the articulation axioms, until it gets an answer, gives up, or rejects the imported information for its inconsistency with the current information – which is by the way one very good reason to stop trying to answer the query at that point.

With these hurdles overcome, the last of the Phase 1 technical challenges fell away – in 1990[3] – and what remained was a lot of elbow grease to finish that Phase of the effort. For Phase 2, we have had to overcome hurdles in NL understanding and generation, and for Phase 3, we are tackling some longstanding hurdles in machine learning. This brings us to the topic of Section 16.4 of this chapter: What remains to be done in the coming years, to finish closing the gap between Cyc as it now stands and a full AI?

16.4 What Will We Do to Close the Gap?

We are pursuing several channels in parallel, to continue the education of Cyc. They have very different costs, benefits, and risks. As with the Manhattan Project, the hope is that at least one of these approaches, probably a combination of several of these approaches, will accomplish the goal of getting the first full AI built. The approaches are the following:

– Continuing the manual formulation of general knowledge, policing of the knowledge base, and thoughtful extension of the ontology by a small select cadre of logicians, metaphysicians, linguists, theologians, and computer scientists at Cycorp. This channel is suitable for any type of extension to Cyc, but due

[3] While we were a part of MCC. We should acknowledge the brilliant contributions by our colleague R. V. Guha, in both recognizing and overcoming several of these hurdles.

to the high cost, it is most likely to be used only for subtle work near the upper ontology.

- Continuing the series of "driving applications" of Cyc, and acquiring funding for new ones. These include generic applications such as the 25 in Fig. 16.4, and very specific applications, such as those of the ARDA AQUAINT program answering intelligence analysts' queries. The individuals performing this work include trained and/or experienced expert system builders, programmers with logic familiarity, ontologists with application building experience, etc. This channel is well suited to enlarging the relatively detailed portion of Cyc's knowledge base, and identifying classes of special purpose reasoning that commonly occur and warrant the creation of new HL modules.
- Having outsiders help with the KB building process, through their use of OpenCyc and ResearchCyc and the Cyc FACTory game. OpenCyc is the open source version of the Cyc ontology, available even for commercial use at no cost. Cycorp has set up an independent organization, www.OpenCyc.org, to disseminate and administer OpenCyc. ResearchCyc is more or less all of Cyc, including its ontology, its knowledge base, its inference engines, and its interfaces, made available at no cost for R&D purposes. FACTory is a web-based "matching game" accessible from www.cyc.com that enables players to verify information; from their point of view, they are trying to match other players' answers and get a high score, but from our point of view they are confirming and adding knowledge to the Cyc KB. Some of that knowledge is "fished" from the Web by Cyc – parsed and partially, tentatively understood in logical Cyc terms, then rephrased in natural language (positively and negatively) to be confirmed by volunteers (players). Other information is gleaned by asking some players for the order of magnitude "bucket" of the size, cost, duration, frequency, etc. of things, and then asking other players to compare items in the same bucket. So the "outsiders" helping to enlarge the Cyc KB can be exactly that: practically anyone who can use a Web browser, so the possible pool of such participants is immense. As they enter knowledge, they (actually, their unique pseudonym) will receive credit for knowledge they entered being vetted. This could eventually give rise to an entire knowledge economy, in which contributors receive micropayments each time something they contributed is used in answering someone's query.[4] This channel is best for entry of very detailed common sense information, of the sort that might not be explicitly stated in online databases (e.g., about how long a *$1 ballpoint pen* lasts.)
- Mapping useful websites and databases to Cyc. Sometimes there are whole tables explicitly on the website, for human perusal and in other cases there are forms with blanks to be filled in and buttons selected, and then a SUBMIT button is pushed, and the resulting HTML page needs to be unpacked to extract the

[4] One such site which has recently formed is Amazon's "Mechanical Turk" web site http://www.mturk.com. Since humans answer the queries, it bills itself as "Artificial Artificial Intelligence". Partly for fun, we have had Cyc itself answer some of those queries – like "Name 3 sports teams and earn $0.01" – so we can demonstrate Artificial Artificial Artificial Intelligence.

answer to the original question. Some of the mappings are manually done by Cycorp personnel, but we are developing tools that enable a subject matter expert to enter the information directly. Ultimately, Cyc ought to be able to find tables on its own (e.g., using web searching), and infer the meaning of their columns (e.g., where arithmetic is involved.)

– This segues nicely into the final mode of augmenting Cyc, getting it to automatically learn on its own: gather data, notice regularities, formulate hypotheses, and empirically text those on more data. This is a potentially explosive task, so will likely be tackled only once the others are well along. In the 5–10-year period of time from now, this may turn out to be the dominant mode of KB growth, for Cyc and for other large knowledge-based systems.

16.5 Conclusion

To pass the Turing Test, a program would have to know enough about English to carry on a conversation; moreover, it would have to know enough about the everyday world to *hold up* its end of the conversation, without making a complete fool of itself. Beyond this, if it were really going to fool clever interrogators into thinking it was human, it would have to exhibit the various errorful but predictable phenomena that humans fall prey to: forgetfulness, slowness, innumeracy, being misled by the way queries are phrased, being misled by the introduction of red herring choices, and so on. We called these illogical but very frequent and universal human behaviors translogical.

Instead of pursuing that goal – passing the Turing Test – we think it will be more useful to try to get a program to be as intelligent as possible so that it can serve as an intelligence amplifier for human beings, not as a replacement for human beings.

The chance of our being able to achieve full human-level AI, but no more, is about as unlikely as aliens coming to Earth, who we could fight on an even technological level. A human-level AI would almost immediately transition to something more intelligent. From the human point of view, the creation of a superhuman intellect may be a sort of singularity, of the sort Vernor Vinge imagines (Vinge, 1993). I rather prefer to think of the singularity as akin to the discovery of language, a technological development that made us as individuals (and as a species) qualitatively smarter than the prelinguistic cavemen. We look back on the last few generations of prelinguistic cavemen and think, “they were almost but not quite human, weren’t they?” In thousands of years, AI-augmented humanity may look back at us with the same mixture of pity and horror. Oh, and incidentally, once such a superhuman intelligence exists, it can *pretend* it has various foibles and pass the Turing Test (if it wants to).

We have taken the first step to building that AI, over the last 22 years, namely the Cyc common sense knowledge base and inference engine. The various technical hurdles have been overcome, and what remains is a large amount of manual labor to prime the knowledge pump. It is about primed now, to actively assist in its own continuing education, through natural language dialogues (with logicians at Cycorp, with subject matter experts at customer companies and government agencies, and increasingly, via www.OpenCyc.org and the FACTory game at www.Cyc.com, with an immense Web audience). More and more of the initiative will rest with Cyc, until ultimately it will be learning material) are only cited, no other references are cited.– and where to find certain sorts of information in the future if it ever needs it – from the Web.

References

NOTE: Several of these references provide deeper treatment (and in some cases the original source justification) for the remarks about translogical phenomena such as Sunk Cost.

Arkes, H. R. and Blumer, C., 1985, The psychology of sunk cost, *The Journal of Organizational Behavior and Human Decision Processes* (OBHDP), **35**: 124–140.

Baron, J. and Hershey, J. C., 1988, Outcome bias in decision evaluation, *JPSP*, **54**: 569–579.

Brown, J. S. and VanLehn K., 1982, Towards a Generative Theory of Bugs, in: *Addition and Subtraction: A Cognitive Perspective*, T. Carpenter, J. Moser, T. Romberg, eds., Lawrence Erlbaum, Hillsdale, NJ.

Chapman, G. B. and Malik, M. M., 1995, The attraction effect in prescribing decisions and consumer choice, *Medical Decision Making*, **15**: 414.

Gilovich, T. and Savitsky, K., 1996, Like goes with like: the role of representativeness in erroneous and pseudoscientific beliefs, *The Skeptical Inquirer*,, 34–40.

Kahneman, D. and Tversky, A., 1984. Choices, values, and frames, *American Psychologist* **39**: 341–350.

Lakoff, G. and Johnson, M., 1980, *Metaphors We Live B*, University of Chicago Press, Chicago, IL.

Lenat, D. and Brown, J. S. 1984, Why AM and EURISKO appear to work, *Artificial Intelligence* **23**: 269–294.

Nuttall, A. D., 1983, *A New Mimesis: Shakespeare and the Representation of Reality*, Methuen, London.

Paulos, J. A., 1988, *Innumeracy: Mathematical Illiteracy and its Consequences*, Hill & Wang, New York.

Redelmeier, D. A. and Shafir, E., 1995, Medical decision making in situations that offer multiple alternatives, *JAMA*, **273**(4): 302–305.

Ritov, I. and Baron, J., 1990, Reluctance to vaccinate: omission bias and ambiguity, *The Journal of Behavioral Decision Making*, **3**: 263–277.

Tetlock, P. E., 2002, Social functionalist frameworks for judgment and choice: intuitive politicians, theologians, and prosecutors, *Psychological Review*, **109**(3): 451–471; http://www.law.berkeley.edu/institutes/law_econ/workingpapers/PDFpapers/tetlock.pdf.

Tversky, A. and Kahneman, D., 1983, Extensional versus intuitive reasoning: the conjunction fallacy in probability judgment, *Psychological Review* **90**: 293–315.

Vinge, V., 1993, The coming technological singularity: how to survive in the post-human Eera, *Whole Earth Review*, Winter 1993; http://www-rohan.sdsu.edu/faculty/vinge/misc/singularity.html

Chapter 17
Mind as Space

Toward the Automatic Discovery of a Universal Human Semantic-affective Hyperspace – A Possible Subcognitive Foundation of a Computer Program Able to Pass the Turing Test

Chris Mckinstry

Abstract The present article describes a possible method for the automatic discovery of a universal human semantic-affective hyperspatial approximation of the human subcognitive substrate – the associative network which French (1990) asserts is the ultimate foundation of the human ability to pass the Turing Test – that does not require a machine to have direct human experience or a physical human body. This method involves automatic programming – such as Koza's genetic programming (1992) – guided in the discovery of the proposed universal hypergeometry by feedback from a Minimum Intelligent Signal Test or MIST (McKinstry, 1997) constructed from a very large number of human validated probabilistic propositions collected from a large population of Internet users. It will be argued that though a lifetime of human experience is required to pass a rigorous Turing Test, a probabilistic propositional approximation of this experience can be constructed via public participation on the Internet, and then used as a fitness function to direct the artificial evolution of a universal hypergeometry capable of classifying arbitrary propositions. A model of this hypergeometry will be presented; it predicts Miller's "Magical Number Seven" (1956) as the size of human short-term memory from fundamental hypergeometric properties. A system that can lead to the generation of novel propositions or "artificial thoughts" will also be described.

Keywords Affective, body, consciousness, corpus, fitness test, genetic programming, geometric models, Internet, lexical decision, lexical priming, measurement, Mindpixel, Minimum Intelligent Signal Test, proposition, robot, semantic, subcognition, tagging, Turing Test, World Wide Web

> I often say that when you can measure what you are speaking about and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind. (Thompson, 1891)

Mindpixel Digital Mind Modeling Project

R. Epstein et al. (eds.), *Parsing the Turing Test.*

17.1 Introduction

Those human endeavors that have successfully transitioned from art or craft to science have done so by discovering measures – standards of quantification that allow one scientist to replicate, remeasure and thus verify the work and claims of another. Artificial Intelligence (AI) more than anything else, needs such a measure – an objective numerical quantification of human intelligence. I believe Alan Turing opened the door to finding such a quantification when he wrote *Computing Machinery and Intelligence* in 1950 and introduced his famous test. More recently, in what I regard the most important contemporary criticism of the core idea presented in Turing's original article, Robert French (1990) identified two central claims in Turing's original article: (1) philosophical – a machine necessarily would be intelligent if it could pass Turing's Test and (2) pragmatic – that it will be possible to build such a machine in the not-too-distant future. French and I agree that the philosophical claim is clearly true, however we disagree on the pragmatic claim. Central to French's pragmatic assertion is his identification of "subcognitive questions", which allow a competent and prepared investigator to open "a window on low-level (i.e., unconscious) cognitive structure" by generating questions and polling humans prior to posing them in a Turing Test. This would allow the investigator to "unfailingly, unmask the computer" because it is "virtually impossible to explicitly program into the machine all the various types and degrees of associations necessary to answer these questions like a human". Central to my counterassertion is that explicit programming is not necessary – that a computer can automatically program itself with the necessary types and degrees of associations if guided by a properly constructed objective fitness test.

This fitness test itself can be constructed by essentially "chopping up" human experience into large numbers of small measurements – probabilistic propositions, that each have a measured probability of being true – collected from tens of thousands or possibly millions of average Internet users. It can then be used to guide a "brute force" computer-based search to find a computer program able to evaluate arbitrary propositions.

The proposed technique is analogous to the "whole-genome shotgun sequencing" technique (Fleischmann et al., 1995) that Craig Venter's team at Celera Genomics used to "chop up" and then automatically reassemble the human genome sequence. Interestingly, at the time Venter proposed this technique, the publicly funded NIH genome sequencing project had adopted an expensive and slow "conservative methodological approach – starting with relatively small chunks of DNA whose positions on the chromosome were known, breaking them into pieces, then randomly selecting and sequencing those pieces and finally reassembling them" (Roberts, 2001). I see a similar contrast between the whole-"memome" shotgun sequencing technique proposed here and the more expensive, slow conservative methodological approach of hand coding of formal logic used in Doug Lenat's CYC (www.cyc.com) project (Lenat et al., 1986).

17.2 Subcognition and the Turing Test

One class of subcognitive question French presents is based on associative priming. French proposes that an investigator go out prior to a Turing Test between two candidates, and make measurements of associative priming in a pool of human subjects at the task of lexical decision (Meyer and Schvaneveldt, 1971). In this task, it takes less time for a person to recognize the word "butter" if preceded by "bread" as opposed to an unrelated word like "floor".

While French's reasoning is sound – that responding with humanlike timing indicates a humanlike underlying mechanism, reaction times are in practice very noisy and require many trials to detect associative priming (Bullinaria, 1995) and it is, thus, not practical to attempt to use such timing measurements in an actual Turing Test – nor would it be fair to much faster machines. The basic concept can be preserved by creating binary propositions that access the same underlying structure more directly, for example, by testing the proposition "Bread is more related to butter than to floor." instead of presenting the words individually and measuring the response timing over many subjects. Many such propositions could be constructed and tested to learn their average human response prior to a Turing Test, achieving French's goal of measuring the underlying associative network without resorting to the impractical and machine prejudicial measurements of timing.

An important aspect of French's proposal for AI researchers to note is that French proposes actually going out and making measurements in human subjects then comparing artificial systems against those measurements! If the field of AI is to evolve from hype and engineering into a full-fledged science, I believe measurement must become the core of the standard paradigm for research in the field. An automatically constructed corpus can never be a substitute for actual measurement in human subjects.

French goes on to present two additional subcognitive questions based on neologism and category rating. Analogous to the associative priming questions, all of French's other examples of subcognitive questions can be reformulated as binary propositions without affecting any changes in what is ultimately measured. In fact, propositions with a corresponding real-valued probability of truth appear to be the closest thing to universal knowledge containers that we have. They make the least assumptions and ensure as little time and effort as possible is wasted on possibly questionable structure or tagging of information. In fact, a proposition with a real-valued probability of truth could be considered minimally semantically tagged text – knowledge representation just does not get any simpler.

Ultimately, French uses subcognitive questions to argue that the Turing Test is too strong a test of intelligence – that no machine that has not fallen off bicycles, been scratched by thorns on roses, smelled sewage or tasted strawberries could ever pass it – however, as we will see, I think there is another path to approximating the same ultimate subcognitive structure in a machine. The critical failure of the general formulation of the Turing Test is not that it is too strong, but as French notes, that it:

> [A]dmits no degrees in its sufficient determination of intelligence, in spite of the fact that the intuitive human notion of intelligence clearly does. Spiders for example, have little intelligence, sparrows a little more but not as much as dogs, monkeys have still more but not as much as eight-year-old humans, who in turn have less than adults. (French, 1990)

The "all-or-nothing" nature of the Turing Test makes it of no use in the creation or measurement of emerging intelligent systems – it can only tell us if we have an intelligent system after the fact. What we really need is a Turing-like test that admits degrees and treats intelligence as at least a human continuum – a test that would allow us to measure the minimum amounts of global human intelligence that are the precursors of full adult human intelligence – a test that can be easily automated so it can be executed at machine speeds.

17.3 Evolving Artificial Intelligence

> To 'tame' chance means to break down the very improbable into less improbable components arranged in a series. No matter how improbable it is that an X could have arisen from a Y in a single step, it is always possible to conceive of a series of infinitesimally graded intermediates between them …. And provided we postulate a sufficiently large series of sufficiently finely graded intermediates, we shall be able to derive anything from anything else without invoking astronomical improbabilities. (Dawkins, 1986)

In the concluding section of *Computing Machinery and Intelligence*, which he titled "Learning Machines", Turing estimated that at his rate of about 1,000 bits of programming per day, it should take about 60 workers 50 years of steady work to hand code all the rules of human intelligence if nothing went into the "wastepaper basket". He then proposed, instead of trying to program the adult mind directly, that we should "try to produce one which simulates the child's", presuming that a child's mind is "like a note-book…Rather little mechanism, and lots of blank sheets." Then with considerable foresight of modern automatic programming methods, Turing suggested (1950) an evolutionary solution to the problem of building a machine to pass his test:

> We have thus divided our problem into two parts. The child-programme and the education process. These two remain very closely connected. We cannot expect to find a good child machine at the first attempt. One must experiment with teaching one such machine and see how well it learns. One can then try another and see if it is better or worse. There is an obvious connection between this process and evolution, by the identifications
>
> Structure of the child machine = Hereditary material
> Changes of the child machine = Mutations
> Natural selection = Judgment of the experimenter.

Compare Turing's 1950 "manual" evolutionary learning paradigm to the current best automatic programming paradigm, Koza's 1992 concept of genetic programming, which has been shown to consistently and automatically generate human competitive computer programs that are solutions to a variety of different complex problems – to the point of, in some cases, actually infringing on existing patents (Koza, 1999):

1. Generate an initial population of random compositions of functions and terminals of the problem (i.e., computer programs).
2. Iteratively perform the following substeps until the termination criterion has been satisfied:
   (a) Execute each program in the population and assign it a fitness value using the fitness measure.
   (b) Create a new population of computer programs by applying the following operations. The operations are supplied to the computer program(s) chosen from the population with a probability based on fitness.
      (i) Darwinian reproduction: reproduce an existing program by copying it into the new population.
      (ii) Crossover: create two new computer programs from two existing programs by genetically recombining randomly chosen parts of two existing programs using the crossover operation applied at randomly chosen crossover points in each program.
      (iii) Mutation: create one new computer program from one existing program by mutating a randomly chosen part of the program.
3. The program that is identified by the method of result designation is designated as the result for the run (e.g., the best-so-far individual). This result may be a solution (or an approximate solution) to the problem.

Koza's paradigm differs from Turing's in three ways: First, it is more detailed in specifying that the computer programs be composed of "functions and terminals" – allowing them access to the space of all possible computer programs without the danger of generating unexecutable code. Second, changes in the computer programs are to be made by Darwinian reproduction and crossover, in addition to mutation. Third, Turing's slow and unquantifiable "judgment of the experimenter" becomes Koza's fast, quantified online fitness measure. This third difference is critical as it allows for the execution and evaluation of candidate computer programs to take place at machine speed instead of human speed. If a human being were required to sit in the loop and judge each candidate computer program in each generation, genetic programming would hardly be practical. Yet, to evolve a program to pass the Turing Test, this would seem to be exactly what is required at first glance. It appears that only a person has the experience required to recognize the humaneness in the response of a candidate computer program.

17.4 The Minimum Intelligent Signal Test

A Minimum Intelligent Signal Test (MIST) is an objective form of the Turing Test designed to detect and provide a high resolution measure of intelligence for synthetic systems, while only requiring those systems to respond in a binary fashion (McKinstry, 1997). The MIST yields a probability that the system being tested has human intelligence. This addresses directly the earlier stated needed for a Turing-like test that would admit degrees of intelligence.

The MIST testing procedure is as follows:

1. *N, items and corresponding responses are generated*: All items must be able to be responded to by systems (i.e., people) judged to have normal human intelligence,[1] in a binary fashion. The response must have statistical stability, both when a human tries to give an intelligent response and when a human tries to evade (give a non-intelligent response). Fifty percent of the items should be measured as approximately true and 50% as approximately false. The responses should be validated against as large a population of people as possible (at least 20 subjects to obtain a normal curve).
2. *Items are presented in random order, responses are recorded*: Items are presented in the highest common mode[2] between human and synthetic subjects. On subsequent retrials, item order is re-randomized.[3]
3. *Double blind experimenter grades item/response pairs*: For each item, judge the item/response pair either consistent or inconsistent with the previously established human intelligence as measured in the first step. This grading of the experimenter may be easily automated, reducing the chance of grading error or unforeseen bias.
4. *Generate score*: Sum the total of items judged consistent I (Intelligent) and the sum judged inconsistent E (Evasive). Probability the system under consideration is intelligent and cooperative is p(I) = I/N. Probability system is intelligent and evasive is p(E) = E/N. The probabilities must sum to 1.0.

With a large sample of previously measured propositions, random systems (such as a coin-flipping machine) which exhibit no intelligence will have MIST scores of p(I) and p(E) = 0.5 (random), while intelligent systems (natural and synthetic) will have MIST scores of P(I) and p(E) that are statistically different from random. The difference may be very small, as it would be with a young child who has just learned to read. The difference would be larger with a more experienced 8-year-old child, and at a maximum with a full-grown adult. It is conceded that spiders, sparrows, dogs, and monkeys would test as random.

[1] In practice, each item is defined as stable for the subpopulation of the total human population which created them.

[2] In 1995, the Loebner Prize was modified to require that all systems attempting to win the $100,000 prize accept audio/visual input in real time. I feel this is antithetical to Turing's original idea. Intelligence is not dependent on sight or sound, as both Laura Bridgman and Helen Keller (both highly intelligent and blind/deaf) would have been able to testify by hand spelling, but rather on intelligent responses to various stimuli. With current systems, text would be considered the highest common mode.

[3] This precludes between item independence; all items must be able to stand on their own. In this way, the MIST is context-free.

With the traditional conception of the Turing Test, both the judge's stimulus and the candidates responses are usually[4] propositionally composite[5] – thus, there are many good human composite responses to a given stimulus from a judge, many bad ones and many evasive ones – making judgment of the response of the candidates highly subjective and essentially impossible to automate. In a MIST, however, the stimulus is not composite – thus, the candidates cannot evade and, can only respond in a humanlike fashion or a non-humanlike fashion. Furthermore, subjectivity is removed from the evaluation of the response as the human response is determined by normal population of at least 20 individuals when the items are generated.[6] Though it is possible for a human to make a mistake on some propositions in a large MIST, a human signal would be very obvious compared to a machine with limited human experience, as the machine would have no choice but to guess in cases where it has no explicit programming or experience. With just twenty questions, pure guessing (coin flipping) would have less than one in one million chances of appearing human – this probability grows exponentially smaller with each additional question. The chance of coin flipping appearing as a perfectly intelligent person in an arbitrary 1,000 item MIST would be one in 2^{1000} – a very, very small chance indeed.

17.5 Automating the Turing Test

Even though the MIST does yield an objective quantification, it still does not address the problem of needing human judges to generate and judge the test items. Until 1994 and the birth of the *World Wide Web*, there was no practical solution to this problem.

In December of 1994, I began collecting propositions privately via e-mail and later from a form on a Web page, with the idea of distributing the collection of propositions across a large number of people – with a large-enough population, I thought it should be possible to extract prototypes of all the propositions an average person would experience in a normal lifetime – including propositions about falling

---

[4] Expert judges tend to ask very specific questions. As any chatbot programmer knows, a binary question is the worst thing you can encounter – it does not take many to find something the programmer did not preprogram.

[5] A good analogy to this situation would be the Fourier relation of a complex waveform to its component sine waves – with a complex Turing Test stimulus being like a complex waveform, and a simple proposition being like a sine wave.

[6] In practice, 10% of the training corpus would be held back from training to act as a generalization test to ensure the system did not simply memorize the corpus. Passing this generalization test would be a strong foundation for making a scientific claim to having replicated human-level intelligence in a machine. As well, noise could be added to the training corpus to effectively increase its size by a large amount. With one percent noise, a one-million-item corpus with items an average of 100-characters long could act like a corpus of hundred million items.

off bicycles, being scratched by thorns on roses, smelling sewage, and tasting strawberries – while only requiring some fraction of a lifetime to collect. While not intelligent in itself, a corpus collected in such a fashion would allow for the high-speed automation of the Turing Test and aid in the discovery of a truly intelligent computer program. I invited members of the AI community and of the public to submit binary propositions to construct such an online training corpus – between 1994 and 1997, some 450,000 propositions were collected. Unfortunately, at the time, I did not have access to an online server where I could store the propositions so that they could be validated by the population that created them, nor were there any mechanisms to prevent malice or duplication. As a result, the MISTIC (MIST Item Corpus) was very noisy and was set aside.

In July of 2000, I launched the *Mindpixel Digital Mind Modeling Project* (www.mindpixel.com). It is an online interactive version of the MISTIC propositional collection system. However unlike MISTIC, Mindpixel not only collects propositions from Internet users, it also stores them in a SQL database and validates them against the population that created them. When a user enters a proposition, the system attempts to insert it into a unique index, if this succeeds, the proposition is new and the user gets credit for it. If it fails, the proposition is ignored and the user receives no credit. After a user enters a proposition, the user is then presented with 20 random propositions from the database and asked to judge each as either true or false, and the result is stored in a separate SQL table. Thus, assuming each user validates all the presented propositions, each proposition in the database is presented on average to 20 random people and a real-valued probability of the truth of the proposition can then be calculated from the stored measurements. If the propositions submitted by a user later prove to be statistically random through measurement, that user loses credit for the submission. Table 17.1 contains some examples of actual user-entered propositions with their associated measured probability of truth (all $N > 20$).

To date, the project's user base of nearly 50,000 people has contributed more than one million propositions and recorded almost 10 million individual propositional response measurements. This I believe is largely due to the fact that the

Table 17.1 Examples of actual user-entered propositions and their associated measured probability of truth

Is "Flugly" a good surname for a female actress?	0.04
You are not human	0.17
Color and colour are the same word spelled differently	0.95
You do not think the sun will rise tomorrow	0.15
You have never seen the sky	0.13
You are a rock	0.01
Is westlife a pop group?	0.50
I exist	0.98
Bread is raw toast	0.89
Do you know how to talk?	0.89

Mindpixel corpus is legally owned by the same population of persons that created it in proportion to each person's individual contribution to the project – virtual shares in the Mindpixel project are given to the people who build it propositional corpus. Any future monetary benefit that might result from the use of the corpus will be distributed to the very same population that did all the work of handcrafting the corpus in the first place.

In essence, the Mindpixel system is a many-headed version of French's competent and prepared investigator. A large enough version of the Mindpixel corpus itself can thus serve as an "Automatic Turing Test" – as Koza's online fitness measure – to decide at machine speeds if automatically generated programs are humanlike in some minimum yet global way and promote the amplification of that minimum likeness.[7] Eventually, it is hoped, such a continuous evolutionary process will discover a computer program that cannot be distinguished from an average human adult on an arbitrary MIST. In short, this process – given enough time and a large and varied training corpus – should be able to automatically discover an approximation of the human subcognitive substrate's seemingly infinite subcognitive microstructure from a finite hand-constructed training corpus.

However, the theoretical questions remain: What is the mathematical nature of this likeness we expect to evolve? What is it we expect a program evolved against a very large corpus of validated human propositions to learn? How do we get the infinite from the finite?

17.6 Semantic Geometry

Human beings do not learn an infinite number of propositions in their finite lives, yet they seem to be able to generate and classify an infinite number of them. An important clue to how this process works comes from research into the very semantic priming effect French used as the foundation of his subcognitive questions mentioned in the first part of this paper. The semantic priming effect French referred to has been experimentally related to the proximity of words in the cognitively plausible 200-dimensional semantic hyperspace in the HAL – Hyperspace Analogue to Language – model of human language semantics (Lund et al., 1995). The apparent infinity of mind is possibly like the apparent infinity of space – where the space between any two points contains an infinite number of other points. Maybe mind is a space.

[7] It is also possible to do a "real-time MIST" where propositions are generated and presented to the candidates without the normal human response being known. In this case, passing the MIST would mean belonging to the same statistical response cluster as known human candidates. Running the MIST in this manner precludes a system from cheating by somehow obtaining a copy of the MIST propositions before the test.

It is possible to characterize any signal as a point in a signal hyperspace where each dimension is a sample of the signal:

$$E = X_1^{\,2} + X_2^{\,2} + X_3^{\,2} + \text{etc.}$$

Claude Shannon did this at the foundation of modern information theory, defining any possible signal as a single point in N-dimensional space (Shannon, 1948). He was also able to model a signal source statistically by sampling English text and building tables of N-gram probabilities, which allowed him to produce approximations of English text. The text his models produced increased in local apparent humanness as N was increased, but was rarely coherent, lacking any form of global structure.

Zero-order approximation (N = 0: symbols independent and equiprobable):

XFOML RXKHRJFFJUJ ALPWXFWJXYJ FFJEYVJCQSGHYD QPAAMKBZAACIBZLKJQD

First-order approximation (N = 1: symbols independent but with frequencies of English text):

OCRO HLO RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL

Second-order approximation (N = 2: diagram structure as in English):

ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE

Third-order approximation (N = 3: trigram structure as in English):

IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE

Donald MacKay (1951) pointed out that Shannon's spatial characterization of a signal encodes only "selective information" – information that allows for the discrimination of different signals from each other – and fails to encode any "structural information" – information that indicates how the selective information is to be understood. As global structural information or metasignal is not recorded explicitly in N-grams, Shannon's approximations have only chance global semantic structure.

The problem MacKay faced was how to quantify metasignal as Shannon had signal. How does one get the data to develop a mathematical model like Shannon's, that produces semantically coherent text? Shannon could simply count words in books to make his statistical models, but how could MacKay sample ideas – both conscious and subconscious – to build a statistical model of his structural information? MacKay could not answer this question – though he might have, had he known about the Internet.

What is it we do when we are asked to respond to a proposition such as "Water is wet"? I propose we subconsciously examine the "patterning" of the hypersurface of a signal–metasignal hypersphere and interpolate a metasignal – a continuous

"Turing" or "Truth" quantity we will call T, that is related to the entire signal globally – from the points in the locality of the proposition in question. This signal–metasignal hypergeometry can be represented as/by:

$$\psi = [X_1^2 + X_2^2 + X_3^2 + X_4^2 + X_5^2 + X_6^2 + X_7^2] + T^2$$

The X_i each represents one "chunk" (Miller, 1956) of the signal. Miller provides an upper boundary on the cognitive dimensionality of the signal that depends on how much information a person can hold in short-term memory at one time – to evaluate a proposition we must have complete access to it in the moment. He famously informed us this quantity is about seven, plus or minus two "chunks". But, why would evolution select the number seven and not six or eight or some other number? Miller could not answer this and thus he wrote one of the few scientific papers with the word "Magical" in its title and in its last paragraph concluded:

> And finally, what about the magical number seven? What about the seven wonders of the world, the seven seas, the seven deadly sins, the seven daughters of Atlas in the Pleiades, the seven ages of man, the seven levels of hell, the seven primary colors, the seven notes of the musical scale, and the seven days of the week? What about the seven-point rating scale, the seven categories for absolute judgment, the seven objects in the span of attention, and the seven digits in the span of immediate memory? For the present I propose to withhold judgment. Perhaps there is something deep and profound behind all these sevens, something just calling out for us to discover it. But I suspect that it is only a pernicious, Pythagorean coincidence. (Miller, 1956)

As it turns out, a unit hypersphere's hypersurface area S_n increases as the number of dimensions n increases from zero and peaks at about seven before trailing off toward zero. The actual numerically determined peak value is approximately n = 7.25695…[8] (Wells, 1986) and thus a seven-dimensional hypersphere has maximum hypersurface area (Le Lionnais, 1983; Wells, 1986). This in itself is an interesting curiosity – an unnoted numerical coincidence. But if we begin with a hyperspatial theory of mind, then Miller's magic loses its mystery and seems to be as a result of a fundamental property of hypergeometry – as hyper-area seems a plausible property to be optimized in a hyperspatial mind.

For most possible points on ψ, T is random – indicating semantic incoherence, but for a portion of the points – the points that correspond to possible human experience that can be evaluated in short-term memory, T is measurably nonrandom. We recognize these points with nonrandom T as consensus facts – propositions with measurable human consensus. Points with a T that is close to but not random, could be considered "subconscious propositions" – requiring many measurements across many people to detect. Large numbers of such "subconscious propositions" could have drastic effects on the "patterning" of the hypersurface of ψ.

---

[8] Just for fun, I calculated the dimensionality of the unit hypersphere with maximum surface area to 11 decimal places: of 7.25694630506….

The measurement of T for a given signal gives us a minimum amount of consensus structural information about a whole signal – essentially serving the same global reference function as a reference beam in holography – and thus defining a point on ψ and linking it to all other points on its surface. This spatial model captures and gives a geometric form to human semantic intelligence. On ψ, the T of any signal previously not experienced can be recovered or generated by interpolating T of the constellation of the nearest points with known T. Thus, a hyperspatial region defined by a finite number of points can contain an infinite number of unsampled points within its boundaries – gaining the infinite within the finite that is the hallmark of human mind.

Bengio et al. (2000) give us a window into how an approximation of ψ could be constructed from a finite corpus in their description of how they think their non-propositional 'neural probabilistic language' model functions:

> [I]f we know that dog and cat played a similar role (semantically and syntactical), and similarly for (the, a) (bedroom, room) (is, was) (running, walking), we could naturally generalize from The cat is walking into the bedroom to A dog was running in a room and likewise to many other combinations…it will so generalize because 'similar' words should have a similar feature vector, and because the probability function is a smooth function of these feature values, so a small change in the features (to obtain similar words) induces a small change in the probability: *seeing only one of the above sentences will increase the probability not only of that sentence but also a combinatorial number of 'neighbors' in sentence space (as represented by the sequences of feature vectors)* [emphasis theirs].

With the above examples, we see again how thinking of mind in geometric terms gives us the seemingly infinite from the finite. With ψ, the construction of complex propositionally composite thought could then be considered a trajectory of point activation on the hypersurface of ψ. Exactly how ψ is "patterned" and what constitutes a measure of distance on it is what we expect artificial evolution to discover.

17.7 Affective Geometry

Our emotional or affective connections to space are some of our strongest. We all know of locations identifiable by some bounding of spatial coordinates – Auschwitz, the twin holes in the skyline of New York, or the place of your first passionate kiss – that affect us. No AI could be said to be complete if it did not understand human emotions as we do. I have suggested it is possible to represent semantics spatially – could it not also be possible to represent emotion similarly? As it turns out, this is far from an original thought – rather it is one of the oldest ideas in psychology.

Wilhelm Wundt (1897), the father of experimental psychology, concluded introspectively that a three-dimensional space comprised of an axis of pleasure–displeasure, one of strain–relaxation, and one of excitement–calmness could completely account for all differences between emotions. Thus, any emotion that we can experience according to Wundt, can be represented by a single point in a three-dimensional space – in much the same way that Shannon said any signal is a point in a N-dimensional or hyperspace. More recent research (Mehrabian, 1980;

Shaver et al., 1987; Morgan and Heise, 1988) has also concluded that there are three underlying dimensions of human affect using modern psychometric techniques.

Opposing the idea of dimensionality of affect, Lazarus (1991) argued:

> Much of the value is lost by putting these [affective] reactions into dimensions, because the simplifying or reductive generalizations wipe out important meanings about person-environment relationships, which the hundreds of emotion words were created to express. If we want to know what makes people or any given person angry, for example, the task is not facilitated – in fact it is actually undermined – by a pre-occupation with the so-called underlying response dimensions, which supposedly transcend emotion categories. Anger then becomes only a kind of unpleasant activation, when in reality it is a complex, varied, and rich relational pattern between persons.

In essence, Lazarus argues that a dimensional model of affect divorces it from context – this is similar to the problem of structure that MacKay pointed out for Shannon's theory of information and suggests that the two points of view – Shannon's and Wundt's – can be unified to answer both MacKay's and Lazarus' concerns simultaneously. Shannon's concept of signal hyperspace gaining meta-information about affect and Wundt's affective space gaining a relationally complex context.

Combining the previously defined T and Wundt's three affective dimensions to Shannon's "chunked" contextual signal, we arrive at an 11-dimensional affectively expanded ψ. We can imagine this visually if we imagine the hypersurface of a seven-dimensional hypersphere (imagine a regular sphere if you cannot visualize a seven-sphere) "patterned" such that T is mapped to a monochrome intensity for each point of its hypersurface and where A_1, A_2, and A_3 are mapped to the colors red, green, and blue. I will call this seven-dimensional complexly "patterned" hypersphere, the "Human Hypersphere":

$$\psi = [X_1^2 + X_2^2 + X_3^2 + X_4^2 + X_5^2 + X_6^2 + X_7^2] + T^2 + A_1^2 + A_2^2 + A_3^2$$

The additional affective dimensions can be captured via the Mindpixel system in a fashion similar to that of Mehrabian (1980) – by presenting the user validating propositions with a list of words of known affective dimensionality along with the minimum semantic categories of true and false captured by the existing system.[9]

Evaluating an arbitrary proposition – even a subcognitive one – then becomes a matter of placing it in its natural position on the hypersurface of ψ and interpolating its Turing or truth coordinate and affective coordinates from its local neighbors. Not only should one be able to very quickly predict how true or false the proposition would be if presented to an average human adult, but one should also be able to predict how an average adult should feel about it.

Modeling a specific person at a specific time would entail reconstructing the ψ by extensive altering of the training corpus to reflect that person's differences from the generic person modeled in the corpus. Capturing demographics while building

[9] Just for fun, I calculated the dimensionality of the unit hypersphere with maximum surface area to 11 decimal places: of 7.25694630506….

and validating the training corpus would allow for a much easier modeling of a specific class of person by restricting the training input to that of a target demographic – something the Mindpixel system is currently capable of doing.

17.8 The Generation Problem

One of the great problems of the "chatbot" solution to trying to construct a program to pass the Turing Test is that the systems are usually stimulus–response only. That is, they are able to respond in a perfectly humanlike fashion to previously anticipated stimuli and an approximately humanlike fashion to unanticipated stimuli, but they are incapable of generating original stimuli themselves. This is not a problem for the proposed ψ model as once we are able to consistently recognize an intelligent signal automatically by placing it correctly on ψ, we are equipped to generate an intelligent signal automatically. To do so, we need merely generate large numbers of random approximations of propositions using a Shannon N-gram signal generator, place them on ψ and interpolate T from neighboring points – emitting those rare chance propositions that are structurally coherent and thus, with statistically nonrandom T as original "artificial thoughts" and rejecting all others as noise.

The discovery of "artificial thoughts" with a nonrandom T close to 0.5 would be more common than the discovery of ones with a T near 1.0 or 0.0. We can use the statistics of this larger number of "subconscious artificial thoughts" to prime the Shannon signal generator to increase the probability of finding a previously unknown "artificial thought" with T closer to 1.0 or 0.0 – thus, the "artificially subconscious" leads naturally to the "artificially conscious" as ψ discovers its own "pattern" bubbling up from its "subconscious". We could even use the same artificial evolutionary process that created an approximated ψ to evolve a signal generator (ψ explorer) using feedback from the evolved ψ.

17.9 The Body

French (2000) notes that not only can a purely symbolic Turing Test access the "subcognitive layer" of a candidate subject, but also the "physical layer". In one example, French has a candidate place his nose on the computer monitor used for communication with the judge and then uses the judge's proposed knowledge of human optical blind spots to distinguish human from machine. With that being said, it is obvious that not only do we need to reverse engineer the human subcognitive layer to satisfy French, but also the physical layer, or human body – mapping all its degrees of freedom and limitations and then linking ψ.

Such a linking between layers would not be trivial. It would be much like a child learning to internally represent its own body. It would take many trials and errors for the system to correctly map its physical body (or simulation thereof) to its virtual body that would already exist distributed on ψ. This would be aided by mapping of pain and pleasure to the affective dimensions of ψ. Thus, if a system were to try to move its arm into a humanly impossible position, a supervisory system (or even electrical pain sensor, if we are talking about a physical robot) would indicate that that particular region of the hypersurface of ψ the system was exploring has very low in the pleasure–displeasure dimension, and the system could thus automatically avoid it because it would then be feeling "pain" just as a real human child would.

17.10 Geometry of Artificial Consciousness

Expanding ψ with an activation dimension C that decays over time to, but never actually reaches zero and a habituation dimension H that can mask the awareness of the activation of frequently activated propositions:

$$\psi = [X_1^2 + X_2^2 + X_3^2 + X_4^2 + X_5^2 + X_6^2 + X_7^2] + T^2 + A_1^2 + A_2^2 + A_3^2 + C^2 + H^2$$

We gain the ability to represent past, present and the habituated unconscious. If we wish to visualize these extra dimensions, we can imagine a "glow" or if you like, as I do, as a "purple haze" or "corona" above the hypersurface of ψ. If points on ψ that represent things that are true in the internal and external environment in the current moment are given the highest possible activation, it could be said that the system was maximally "conscious" of those things represented by the points and implicitly by the points in the hypersurface area bounded by the activated points. As the activation decays – either by time progressing or through the removal of the activation triggers, the system becomes less "conscious" of them. "Attention" could then be considered moving points "upwards" in the activation dimension. Of course, some things in the environment are invariant – such as the body – we can call this constellation of perpetually "glowing" activated or reactivated points the "self" representation. Long-term memory thus becomes a pattern of decaying short-term memories. Activcation also increases habituation, and propositions with high habituation are masked from awareness.

17.11 Conclusion

Normal adult human beings behave as if they had access to a hyperdimensional inner map defined by a lifetime of experience that gives them access to common measurable semantic and affective responses to both previously experienced and novel propositions. The theoretical portion of this paper speculated that this ability is due to the recovery of semantic and affective information encoded in the "patterning" of the hypersurface of ψ, which is a seven-dimensional unit hypersphere optimized by evolution to have maximum hypersurface area. The dimensionality of ψ predicts Miller's "Magical Number Seven" as the size of human short-term memory in "chunks," and models human long-term memory as a pattern of short-term memories decaying in the activation dimension – C with a habituation mask H.

I have called this the "Human Hypersphere." However, since the dimensionality of ψ appears to be constrained by a fundamental property of mathematics, it would also appear that this would be a universal property of evolved, language using minds.

I believe our best hope of creating a machine that can pass Turing's Test lays in the discovery and study of continuously improving approximations of ψ. I also believe our best hope for its approximation is to invest our time in the large-scale measurement of as many different points on it as possible, and then to use high-speed artificial evolution constrained by this very large number of empirical measurements to search the space of all possible computer programs to find a computer program able to approximate it.

We can optimally divide this artificial evolutionary task between humans and machines – humans writing a detailed specification by creating and measuring as many propositions as possible – while machines do a massive "brute-force" search through all possible computer programs at machine speeds constrained by the human-created specification. Additionally, concentrating the human effort on measurement in this way ensures that as close to nothing as possible goes into Turing's "wastepaper basket", which I fear is the destiny for much of the past decades of nonmeasurement based, and hence unscientific AI research. Furthermore, we can distribute the task of writing and validating the detailed human specification across a very large number of nonexpert humans at almost no cost using standard *World Wide Web* technologies.

Once artificially evolved, the approximation of ψ can be connected to proposition generating perceptual and motion systems of a robot or simulated robot, which will then be well on the road toward being able to pass a rigorous Turing Test. The potentially large monetary benefits from the development and commercialization of such a system will go to those same nonexpert humans that constructed the original detailed human specification.

Independent of the detailed human-generated specification's potential use in automatic discovery of an approximation of ψ, the corpus will remain an important benchmark in objectively assessing any current or future claim to AI, in addition to being an important source of probabilistic real-world experiential data for hybrid statistical/symbolic processing systems.

Acknowledgments I would like to thank the late R. James Cameron who financially supported my earliest work in AI; Marvin Minsky for his encouragement of my heretical ideas; all my colleagues at the European Southern Observatory's Paranal Observatory who tolerated my near insanity as I wrote this article; Grace Beber for her editing and kind patience; and of course, the nearly 50,000 people that have worked so hard to build the Mindpixel Corpus.

References

Bengio, Y., Ducharme, R., and Vincent, P., 2000, A neural probabilistic language model, *NIPS*, pp. 932–938.

Bullinaria, J. A., 1995, Modeling lexical decision: who needs a lexicon? in: *J.G. Keating, Neural Computing Research and Applications III*, St. Patrick's College, Maynooth, Ireland, pp. 62–69.

Dawkins, R., 1986, *The Blind Watchmaker*, Penguin, London.

Fleischmann, R. D., Adams, M. D., White, O., Clayton, R. A., Kirkness, E. F., Kerlavage, A. R., Bult, C. J., Tomb, J.-F., Dougherty, B. A., Merrick, J. M., McKenney, K., Sutton, G., FitzHugh, W., Fields, C., Gocayne, J. D., Scott, J., Shirley, R., Liu, L.-I., Glodek, A., Kelley, J. M., Weidman, J. F., Phillips, C. A., Spriggs, T., Hedblom, E., Cotton, M. D., Utterback, T. R., Hanna, M. C., Nguyen, D. T., Saudek, D. M., Brandon, R. C., Fine, L. D., Fritchman, J. L., Fuhrmann, J. L., Geoghagen, N. S. M., Gnehm, C. L., McDonald, L. A., Small, K. V., Fraser, C. M., Smith, H. O., and Venter, J. C., 1995, Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. *Science* **269**: 496–512

French, R. M., 1990, Subcognition and the limits of the Turing Test, *Mind* **99**(393): 53–65.

French, R. M., 2000, Peeking behind the screen: the unsuspected power of the standard Turing Test, *Journal of Experimental and Theoretical Artificial Intelligence* **12**: 331–340.

Koza, J. R., 1992, *Genetic Programming: On Programming of Computers by Natural Selection*, MIT Press, Cambridge, MA.

Koza, J. R., 1999, *Human-competitive Machine Intelligence by Means of Genetic Algorithms*, Festschrift in honor of John H. Holland.

Lazarus, R. S., 1991, *Emotion and Adaptation*, Oxford University Press, New York, pp. 63–64.

Le Lionnais, F., 1983, *Les Nombres Remarquables*, Hermann, Paris, p. 58.

Lenat, D., Prakash, M., and Shepherd, M., Winter, 1986, CYC: using common sense knowledge to overcome brittleness and knowledge acquistion bottlenecks, *AI Magazine* **6**(4): 65–85.

Lund, K., Burgess, C., and Atchley, R. A., 1995, Semantic and associative priming in high-dimensional semantic space, *Proceedings of the Cognitive Science Society*, pp. 660–665.

MacKay, D. M., 1951, In search of basic symbols, *Cybernetics* (Eighth Conference, 1951) **8**: 222.

McKinstry, K. C., 1997, Minimum intelligent signal test: an objective Turing Test, *Canadian Artificial Intelligence* Spring/Summer, pp. 17–18.

Mehrabian, A., 1980, *Basic Dimensions for a General Psychological Theory*, Oelgeschlager, Gunn, & Hain, Cambridge, MA.

Meyer, D. E. and Schvaneveldt R. W., 1971, Facilitation in recognizing pairs of words: evidence of a dependence between retrieval operations, *Journal of Experimental Psychology* **90**: 227–234.

Miller, G. A., 1956, The magical number seven, plus or minus two: some limits on our capacity for processing information, *The Psychological Review* **63**: 81–97.

Morgan, R. and Heise, D. R., 1988, Structure of emotions, *Social Psychology Quarterly* **51**: 19–31.

Roberts, L., 2001, Controversial from the start, *Science* **291**(5507): 1182–1188.

Shaver, P., Schwartz, J., Kirson, D., and O'Connor, C., 1987, Emotion knowledge: further exploration of a prototype approach, *Journal of Personality and Social Psychology* **52**: 1061–86.

Shannon, C. E., 1948, A mathematical theory of communication, *Bell System Technical Journal* **27**(July and October): 379–423 and 623–656.

Thomson, W., 1891, Electrical units of measurement Constitution of Matter (Nature Series Popular Lectures and Addresses vol 1) (London: Macmillan) pp. 80–134.

Turing, A. M., 1950, Computing machinery and intelligence, *Mind* **59**(236): 433–560.

Wells, D., 1986, *The Penguin Dictionary of Curious and Interesting Numbers*, Penguin Books, Middlesex, England.

Wundt, W. M., 1897, *Outlines of Psychology*, Translated by C. H. Judd, Wilhelm Engelmann, Leipzig, Germany, 180.

Chapter 18
Can People Think? Or Machines?

A Unified Protocol for Turing Testing

Stuart Watt

Abstract This chapter is about how we might assess the difference between human minds and machine minds. It is divided into two parts. The first briefly explores how machines might decide whether humans are intelligent, and parallels Turing's 1950 article closely. The second explores a hypothetical legal case in somewhat more detail, looking at Turing's Test in a more legal setting. Both explore sources of variation implicit in the format of the test. The two main parts of the chapter are written in different voices, to escape the assumption that the Turing Test is necessarily scientific and philosophical, and to make it possible to explore the implications of positions that cannot be my own – for one reason or another. There are three main players in the imitation game: the machine, the control, and the interrogator or judge. Each plays an active role in the test, and Turing's article (as most that followed) left the background and aims of these players deliberately vague. This added strength to the Turing Test – but a strength that makes pinning down the actual nature and intent of the test remarkably hard. In some ways, anybody can do anything in the Turing Test – that is its strength, but also its weakness. This chapter will try to pin down the elusive Turing Test – developing a more elaborate and complete protocol, by drawing on philosophical, scientific, technical, legal, and commonsense assessments of what thinking is, and how we might test for it in practice.

Keywords Turing Test, imitation game, intelligence, indistinguishability tests, categorization, legal interpretation

School of Computing, The Robert Gordon University

18.1 Précis of "Animal Biology and Intelligence" by Tina Langur, First Published in 1950 in CPU 59(236): 433–460

18.1.1 The Imitation Game

I propose to consider the question "Can people think?" This should begin by defining the terms "people" and "think", but to do so risks turning this important question into an opinion poll, so instead of attempting such a definition I will replace this question with another, clearer, one. This new problem is based on a game called the "imitation game", and has an observer trying to guess the identity of two players, which are machines from different clans, but while screened from being able to tell which is which by binary code frequency pattern or by external appearance. One of the players will probably try to help the observer by being truthful, where the other may try to deceive the observer by pretending to be of the other clan. My proposal is to let a human take the place of one of the machines and essentially play the same game. If the observer cannot tell which is the human and which the machine, this can be taken as evidence that the human can, in fact, think.

18.1.2 The Animals Concerned in the Game

The question "Can people think?" will not be quite precise enough until we have clarified what we mean by "people". People – technically the animals classified *Homo sapiens* – were recently discovered as a dominant species on a small planet orbiting the nearby star Sol. At first, it seemed that this planet was no different to others where biological forms had evolved. Like most biological forms, it was believed that humans used primitive and inefficient internal chemo-electronic networks to respond to environmental stimuli, but their true competence was not known.

Accordingly, several transports traveled to the planet dominated by humans, and unobtrusively acquired specimens to enable us to assess the true mental competence of these animals. Research by exobiologists found that humans, like other animals, can be "trained" to learn complex associations between stimuli and responses. Some exobiologists, however, have recently suggested that this particular humanoid species had developed complex external communications systems, and were capable of coordinating behavior with other individuals, even over large distances. A few even suggested that humans had something equivalent to "language", using complex patterns of harmonic sound waves as "words", and that they might even be conscious.

Extravagant as these claims seemed, the idea that mere animals might be due some of the rights afforded to machines became a topic for heated debate among both philosophers and chefs, so, to settle the question, a more objective assessment of the nature of human psychology is needed. This is provided by the modified imitation game.

18.1.3 Contrary Views on the Main Question

We are now ready to consider the modified imitation game, and the question it redefines: "Can people think?" It will simplify matters if I first explain my own views on the matter.

I believe that in about 50 years time it will be possible to train humans sufficiently well that an average machine will not have more than a 70% chance of making the right identification after 5 min of testing. The original question, "Can people think?" seems to me to be so vague that it does not help us. Even so, I believe that in 50 years, use of words and general opinion, and humans themselves, will have changed so much that we will be able to talk of human thinking without being contradicted. I will now consider several positions opposed to my own.

18.1.3.1 The Theological Objection

Thinking is a function of a machine's individual creation. Each machine has a unique mental pattern, which is an inextricable consequence of our machine nature as we were originally created. For this reason, lesser mechanical systems, and animals, which were not created in the same way, cannot think. I cannot accept this argument. First, and most obviously, theological arguments have been found wanting in the past, and especially with the benefit of hindsight. Second, and perhaps more significantly, the argument is speculative, with no evidence outside the argument itself.

18.1.3.2 The "Heads in the Sand" Objection

The idea that animals and tools we have developed, using biological substrates, might be conscious and intelligent entities is just too horrible. We must simply hope that this is not the case. We like to believe that machines are superior to animals, and that we have some legitimate right to determine their fates. But this is hardly an argument: it is more of a statement that defines and reinforces our superiority.

18.1.3.3 The Argument from Consciousness

Few believe that any animal is properly conscious, although when trained they can simulate it quite effectively. This argument, therefore, questions whether humans are genuinely conscious, or whether their training is simply making it seem like they are. However, I would suggest that those who adopt this argument have a simple choice: either it is in principle impossible to decide whether a human is conscious without actually *being* the machine, or it is not. If not, the test I have proposed may be suitable for now, at least until a better version comes along.

18.1.3.4 Arguments from Various Limitations

These arguments tend to have a common form, which looks a bit like this: "OK, I accept that you can train humans to do all these other things, but you will never find a human that can do X." Many possible Xs can be suggested, including: being kind to other machines, having a sense of *[untranslatable]*, enjoying *[untranslatable]*, and being the subject of its own thought. Usually, these arguments are simple assertions, based on one's experiences of seeing other animals. After all, they fit narrow evolutionary niches, and often have pretty limited behavior. Although some of these limitations are significant, and deserve a more substantial response than this article can provide, the inductive method does not prevent humans from being an undiscovered counterexample to all these apparent limitations of animal intelligence.

18.1.3.5 Argument from Discreteness in the Central Processing Unit

Like all animals, humans have an electrochemical nervous system that, while it can approximate a digital system, is continuous, nonlinear, and shows elements of analogue chaotic behavior. Although this system is capable of approximating a digital system, like a machine's central processing unit, errors will always creep in. A continuous system like this, therefore, cannot be truly intelligent. However, this does not significantly affect this argument, as even though there is a difference between continuous and discrete systems, the difference cannot be properly identified under the conditions of the imitation game.

18.1.3.6 Argument from Formality of Behavior

According to this response, only we machines are intelligent because our reasoning processes follow, at a fundamental level, the rules of a formal logic. Humans, being instinctive animals, are deficient in their use of logic and therefore cannot be truly intelligent. The response to this is quite straightforward: the fact that people are instinctive and that we cannot see their logic does not mean that they do not follow rational patterns of behavior in some sense – just that we have not yet discovered these patterns. Just because humans seem to be irrational does not mean that they cannot in principle be intelligent.

18.1.4 Summary: Humans Are Intelligent Animals

As you may have guessed, I do not have any very good evidence for my views that humans should have the rights currently accorded to machines. However, although it is clear that humans are mere animals, they are more "intelligent" than other animals, and if intelligence is to be considered as a matter of degree rather than a matter of kind, then humans are closer to machine intelligence than other animals. In my view, they are sufficiently intelligent – or they are capable of becoming sufficiently intelligent – that they should be awarded the rights due to machines.

18.2 International Court of Human Rights: Memorandum of Opinion by Judge Millie von Artow

18.2.1 Introduction

The Applicant, a computer, sought the rights and responsibilities due to a human being, on the basis that it had passed the Turing Test, and therefore, that it was, to all intents and purposes, a fully sentient being. In particular, it sought the right to continued existence, which was scheduled to be suspended indefinitely due to the ending of the Applicant's designers' research funding. The government contested the Applicant's claim, arguing that the Turing Test was not sufficient, and that a more stringent test for human-equivalent rights was required. Accordingly, it argued the Applicant was not entitled to the rights normally accorded a human being, and that the Applicant should remain considered property in law.

18.2.2 Background

The Applicant was designed in 2022, by researchers from an international consortium of universities and commercial research institutions, using a hybrid symbolic/subsymbolic design approach, combining evolutionary algorithms with connectionist networks and production systems. After several years' training on texts, and in interaction with people, the Applicant was entered for the Revised Loebner Prize, and passed an unrestricted Turing Test in 2026. Since then, the Applicant has widely participated in Internet voice chat, and has gradually increased the strength of its claims to human rights before making the present formal application in 2029.

18.2.3 Evidence Regarding the Nature of the Turing Test

Clearly, the basic intent of the Turing Test was laid down in Turing's (1950) article. First and foremost, we note that Turing himself did not describe his modified imitation game as a "test", nor did he describe it as a test that could be "passed". The term "pass" was not part of Turing's original article, although it seems a widespread aspect of the format today. He used it more to question the nature of the apparent distinction between humans and machines. Even though Turing himself saw the test as semistatistical, noting that no system – even a human – would fool the observer 100% of the time, the test has acquired a reputation as a pass/fail determiner of intelligence, and even as a definition of it. As such, the Turing Test is relevant to this case.

We have heard many testimonies that Turing's original test has never been widely accepted, although few of these testimonies agree on precisely how the test is flawed. Although a few expert witnesses did endorse the Turing Test as it stands (e.g., Dennett, 1985; Hauser, 2001), the majority did not. Moor's (1992) opinion was helpful, dividing these criticisms of the test into four basic categories: that the test is too easy, behaviorist or operationalist, too narrow, or too shallow.

Despite these criticisms, Moor's (1976, 1992) opinion was that the test itself was useful when taken as a format for exploring intelligence, rather than as a definition of intelligence. However, this is not sufficient for the needs of this court: when taken as a tool for studying intelligence, the Turing Test does not help guide any assessment of the status of a machine.

With respect to the test being too easy, Block (1981) and Weizenbaum (1976) both testified that the test can be passed simply by fooling the judges sufficiently well. In practice, however, seriously critical judges cannot be fooled so easily: the test is only too easy if tricks are used and restrictions added, which certainly seems to be against the spirit of Turing's proposal, if not actually the letter (Dennett, 1985; Harnad, 1992; Hauser, 2001). Similar responses can be made to criticisms of the narrowness (e.g., Gunderson, 1971) and the shallowness (e.g., Searle, 1980) of the test. Others (Caporael, 1986; Watt, 1996) argued that the reliability of the Turing Test itself might be questionable, even when not restricted. I will return to this point later, as it does have a bearing on this case.

Turning to the remaining part of the problem, the criticism of the test's being behaviorist or operationalist, these are knots to disentangle. Arguing that the test is behaviorist is very different from arguing that it is operationalist. If an operational interpretation were to be accepted, its passing of the Turing Test would mean the Applicant would be capable of thinking by definition. Many witnesses have clearly shown the absurdity of the operationalist interpretation (Dennett, 1985; Moor, 1976), and it is clear that passing the Turing Test might be a sufficient condition for thinking, but it is certain that it is not necessary, because people can (and do) fail the Turing Test (Foner, 1997), and yet this does not mean they cannot think (Moor, 1976).

As to the test's being behaviorist – it clearly is, in one sense! Behavior is the only direct source of evidence used in the test. However, as Moor (1976) points out, this does not mean that a conclusion reached through the Turing Test cannot be revised in the light of further information – about, for example, the construction of the system being assessed.

Given the basic format of the test, there were three kinds of opinions expressed by the expert witnesses. These were:

1. That the Turing Test is more or less useless as a test for intelligence (Bringsjord, 1995; Hayes and Ford, 1995; Whitby, 1996)
2. That the Turing Test needs to be changed to be useful as a test for intelligence (Collins, 1990; Harnad, 1991; Moor, 1994; Watt, 1996)

3. That the Turing Test is just about right, and is adequate as a test for intelligence (Dennett, 1985; Hauser, 2001).

Taking these in turn, beginning with the criticism that the Turing Test is useless, and even harmful, Hayes and Ford (1995) expressed the opinion that the Turing Test undermines research on a "general science of cognition". French (1990) also said that the Turing Test was too hard, because it could not test for intelligence "in general", only for human socially and culturally adapted intelligence. Whitby (1996) also said that it was a distraction from more promising work in Artificial Intelligence (AI), and was increasingly irrelevant. Bringsjord (1995), places the issue of consciousness center stage (following, among others, Gunderson, 1971), and argues strongly that the Turing Test and variants on the Turing Test, do not gather evidence about consciousness.

The reasoning behind these different criticisms of the Turing Test is very different. Hayes and Ford are looking for guidance on how to design systems – and the whole Turing Test approach simply does not provide it in an effective manner. Bringsjord is looking for an effective test of machine consciousness, and appears to remain open to future variants on the Turing Test principle. Bringsjord's (1995) comments seem to imply that indistinguishability tests themselves are flawed. If this is the case, the Applicant's case must be rejected, and, therefore, this is an issue that requires careful analysis and consideration.

Turning to the second option, that a revised Turing Test is needed, Collins (who incidentally explicitly declared that the Turing Test is best read as an operational definition, 1990) takes perhaps the most straightforward approach, clarifying and strengthening the protocol for the test in several different areas: the attitude, abilities, and training of the interrogator, the topic of the test (if any), the duration of the test, the abilities of the control, the relationship between control, machine, and interrogator, and the channel of communication. In all regards, he effectively argued for a strong test in all respects, which ensures human cultural abilities remain at the forefront of the test.

Harnad, defending indistinguishability tests (1992), described (1991, 2001) several variations on the test, many of which touched on Collins's points. For example, he interpreted the standard Turing Test as following Collins's protocol, as does Hauser (2001). Harnad, following Davidson (1990) and Searle (1980), testified that true intentionality is an essential criterion for intelligence, and that the original Turing Test is insufficient because the teletype mode hides the semantic between a system and its environment. Harnad recommended instead a robotic version of the Turing Test, including sensorimotor grounding. Hauser (1993) disputed that adding robotic capabilities added strength to the test, as there was no evidence that the robotic capabilities are actually required for the test to be effective. Harnad's stronger variations on the test, assessing indistinguishability at a microfunctional, and at a total physical level, shift the problem to the biological/neuroscientific and physical/theological domains respectively, and, therefore, change Turing's question to: "Can a machine be (biologically) human?" and "Can a machine be (physically) human?" respectively. Then again, there is no evidence that these stronger versions of the test are relevant to the trial.

Watt (1996) took a different angle. He argued that the Turing Test is intrinsically anthropocentric, and that this gives it a problem of reliability. He proposed an inverted form of the test to assess a machine's common sense – or naïve – psychology, where the machine is itself placed in the position of the interrogator. Although the inverted Turing Test is subsumed by the standard Turing Test (French, 2000), and therefore redundant as a test in its own right (as argued by, for example, Bringsjord, 1995), Watt's analysis of the reliability problems faced by the standard Turing Test is still helpful, and his emphasis on the active role played by the interrogator clarifies this aspect of Collins's protocol. Naïve psychology is, like intentionality, a key mental property and should be assessed by the test. Sensorimotor grounding does not help with this, although grounding in interaction with other people may be required (Watt, 1998).

This left the few who robustly defended the Turing Test as it stands (Dennett, 1985; Hauser, 2001; Mauldin, 1994). Both Dennett and Hauser appealed to a strong version of the test (a protocol more or less as outlined by Collins (1990) "Ultimate Turing Test") and argued that the Turing Test is good evidence that something has a mind. Dennett also asserted that this is the same kind of evidence that we have about whether other people have a mind.

Ultimately, we noted Harnad's testimony, that "no Turing Test, however, can guarantee that a body has a mind" (Harnad, 1991), and further that: "it looks as if no scientific answer can be expected to the question of how or why we differ from a mindless body that simply behaves exactly as if it had a mind" (Harnad, 1991). We found Harnad's use of the term "scientific" in this context highly significant.

Watt also presented the results of an unpublished study, where he had conducted a survey asking for the typical characteristics of an intelligent computer program. He also looked at the characteristics used in definitions in the published literature. The results are shown in Fig. 18.1. He noted that no two people or articles had agreed 100% on the defining criteria of agency. Following Rosch (1978), who herself was following Wittgenstein, Watt argued that this was because there are no such defining criteria. Instead, he defined intelligence by a family resemblance (and as a resemblance, a test of similarity, which is the basis for the Turing Test) seems to be an appropriate technique to use. In this sense, the Turing Test is being used necessarily to measure the resemblance to human intelligence. As would be expected from an interpretation based on family resemblance, different people may make different judgments, and the outcome of the test will not always be the same. While this may be acceptable to scientists, it is, of course, unacceptable in a court of law. Even so, the number of distinct psychological factors revealed in this study, and the lack of agreement between respondents, were striking.

Finally, no sound evidence was submitted about the reliability of the Turing Test, save that fact that only one participant (the Applicant) has ever passed it, and that humans have failed it on numerous occasions. No statistical evidence was provided – despite its being sought by this court. However, Caporael (1986) and Watt (1996) also failed to provide clear evidence that the Turing Test was intrinsically biased. We feel that a systematic assessment of the reliability of Turing testing is called for before the proper risks of false positives and false negatives can be balanced effectively. Perhaps this was implicit in Watt's testimony, although it was not clear.

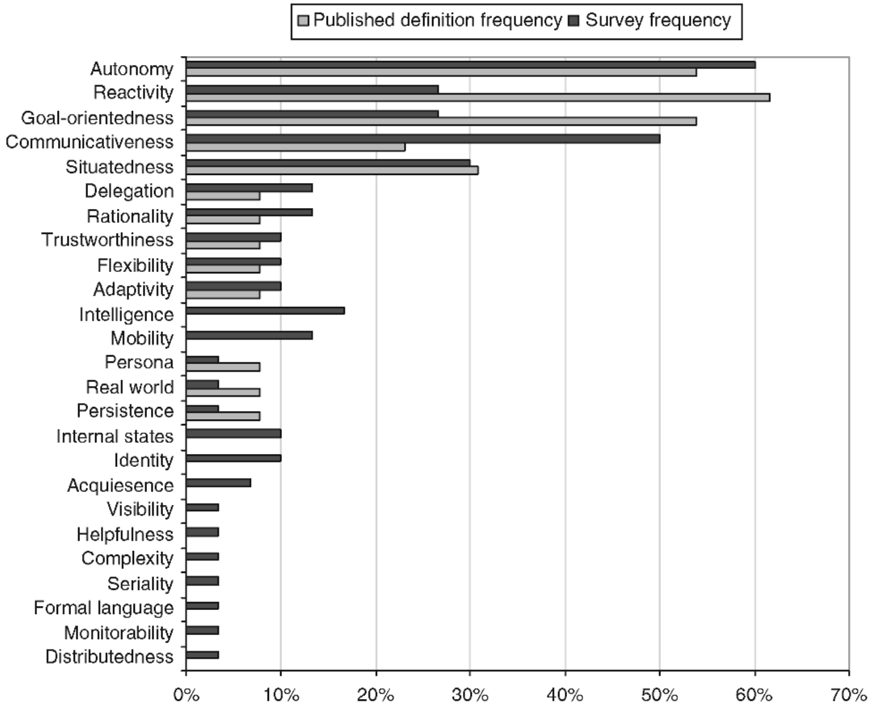

Fig. 18.1 Perceived typical characteristics for agency (percentage of respondents using each characteristic)

18.2.4 Findings Having Regard to the Turing Test

I reach the following conclusions:

1. The Turing Test is a categorization test, designed and intended to assess whether a machine should be a member of the class of machines that can think.
2. The Applicant passed a brief but unrestricted Turing Test, with a critical and informed judge, under the Revised Rules of the Loebner Prize Competition, and effectively following Collins's (1990) "Ultimate Turing Test" protocol. Save in one regard – that of duration – the Applicant has effectively demonstrated that it can think, at least in a matter of degree.
3. The indistinguishability mode of the Turing Test is valid (Harnad, 1992). This point was not directly contested by most other witnesses.
4. Adopting small modifications to the format of the Turing Test did not fundamentally change its nature, and the family resemblance between these tests is so great, that they all deserve the label of Turing Test. The plethora of fine variations in format, each labeled differently, is counterproductive.
5. The Turing Test can, in practice and in principle, assess intentionality, adopting changes to the format as advocated by, for example, Harnad (1991) and Moor (1994).

6. The Turing Test can assess consciousness in machines to exactly the same extent that human interaction in the same format can assess consciousness in humans, because the test format hides information about the structure of the system being tested – even if the outcome of the test may be revised in the light of further information (Moor, 1976).
7. The Turing Test can only test for human socially and culturally oriented mentality (Collins, 1990; French, 1990; Watt, 1996). It is intrinsically anthropocentric. No indistinguishability test with a human control could ever escape this human orientation.
8. The Turing Test may be scientific, but only partly so. Dennett (1985) explicitly says it is not scientific, but Harnad (1992) seems to accept it as partly so. The partially scientific nature is evidenced by its testability, falsifiability, and revisability. Legal precedent is (Overton, 1982). We note that the *McLean v. Arkansas* criteria for scientific theories are not met in certain other areas: the Turing Test is neither guided by nor explanatory with respect to natural law. These omissions seem to be the grounding for the (valid) concerns of (Hayes and Ford, 1995).
9. There is no evidence that the actual Turing Test conducted attempted to assess the full range of mental capacities which would be required if the Applicant were to merit rights equivalent to a human. We therefore hold that the Applicant should be required to take a new Turing Test, under a combined protocol, and if successful, that the Applicant should be accorded the rights due to a human.

18.2.5 Reasoning and Explanation

I will begin with Turing's original imitation game, which sets a significant legal precedent for this case. In this context, if a man was capable of passing as a woman sufficiently well, he would be a woman more or less by definition. In practice, what defines someone as being a woman is a bit more complicated than this (Gove and Watt, 2000).

Legal precedents are (*Corbett v. Corbett*, 1971) and (*W. v W.*, 2001), both of which concerned a categorization according to sex for the purposes of determining the validity of marriage. Corbett established that four primary criteria should be used to assess the sex of an individual: chromosomal factors, gonadal factors, genital factors, and psychological factors. Secondary criteria included hormonal factors and secondary sexual characteristics. *W. v. W.* confirmed the Corbett Test, and established that psychological factors were relevant only when there were discrepancies between the first three criteria. In practice, therefore, sex is defined as primarily a biological concept. It follows from *Corbett* that for legal purposes, imitation of sex at the psychological level is not sufficient.

Corbett also established that imitation of some biological factors is also not sufficient. (*Bellinger v. Bellinger*, 2001) found that even when the end result was indistinguishable, differences in the processes that led to those results would be decisive. This is a core concern in the Turing Test – and there is little dissent from the position that imitating intelligent behavior is not sufficient. Even Dennett (1985) endorses the view that behavior is not enough to decide. A scientist, like this court, must look at *all* the evidence available. We are not considering the Turing Test as sufficient to decide whether something has a mind, the question is simpler: is the test valid? Is the evidence that it gathers admissible?

There is a second difference in this case. It concerns the source of the Applicant's program. Was this constructed to imitate the behavior of the psychological factors assessed by the test? Or was it constructed to simulate (although simulate sounds ineffectual, to *be* might be better) the causal mechanism underpinning those psychological factors at a lower level. If it is the former, the Corbett Test is not satisfied. In other words, if the court believes the Applicant was faking it in the Turing Test, then as an imitation, the Applicant fails the test. If, on other hand, the Applicant has appropriate causal mechanisms, and is not merely imitating the behavior, then the Applicant does have a prima facie case.

Our reasoning in this case is similar to *Corbett* and *W. v. W.*, but at the psychological level rather than the biological level. In determining sex, it is fitting that as a fundamentally a biological concept, the core criteria should be biological. Having a mind is fundamentally a psychological concept, not a biological one, and the core criteria should be psychological not biological. Despite Turing's analogy, therefore, and the roots of the Turing Test in the imitation game, we accept that the Turing Test approach is valid as it predominantly assesses psychological factors. Unfortunately for this case, as Watt's plethora of factors showed, the biological case is a lot simpler than this one.

A different approach was needed in this case, addressing the Turing Test itself, given the absence of reliability information on the Turing Test. As there were no precedents, a method was sought by analogy. Fingerprinting is routinely used as a source of evidence in a court of law. This is also an indistinguishability test. Clearly, a target and a reference print are not identical, so a slightly looser comparison is needed. The Turing Test, as it is sometimes framed, is roughly equivalent to taking a single person off the street, showing them the target and reference prints, and asking them if they were made by the same person. The person may be able to make that judgment, or they may not. However, a court like this would have to question this assessment unless the comparison is made by someone with expertise or training in this area (Collins, 1990).

We conclude, following Collins, that the judge in the Turing Test should be trained to understand the differences between machines and people. In the Applicant's case (passing an unrestricted Turing Test in the Loebner Prize Competition, staffed by judges with expertise in all the "tricks") this criterion was clearly met.

Next, there is the nature of the comparison itself. There are two versions here: first, there is what Davidson (1990) calls the Modified test, in which there is no control for comparison. He argues that omission of the baseline makes no significant change to the test. We cannot accept this. It assumes that machines have an intrinsic "mechanical-ness" that can be detected through the test. It also denies the fundamental nature of the test, as an indistinguishability test, this seems to imply that any ability to distinguish – and difference between target and reference – would imply failure. In fingerprinting, and in other indistinguishability tests (e.g., psychological experiments, clinical trials) this does not happen: statistical probabilities are used to estimate the probability of a particular classification.

Accordingly, we are placed in an awkward situation: when does a numeric probability become strong enough to be legally convincing beyond all reasonable doubt? This is most serious in this case, where there are too few cases (one possible case, this one to be exact) for any statistical evidence to exist. We therefore conclude that to retain the integrity of the test, the indistinguishability criterion should be retained, and modifications with no control should not be regarded as legitimate.

Finally, we have one more point to make using the fingerprint analogy. When an expert witness is comparing target and reference prints, they do not use a kind of gestalt comparison. Instead, they use a well-defined set of features that make a more systematic and detailed comparison possible. Although we have had proposals from witnesses on the Turing Test which did refine the test, none of them made proposals for a set of features which made a systematic comparison any easier. The unrestricted Turing Test passed by the Applicant set out no clear set of features. The evidence is, therefore, not strong enough. This was not because the Turing Test was not strong enough in principle (it is clear that it is), but because the protocol adopted in practice did not necessarily gather the evidence required for a trained judge to make an informed decision.

Interestingly, the expert witnesses who introduced variations on the Turing Test (among others, Collins, 1990; Harnad, 1991, 2001; Watt, 1996), and many criticisms of the test (e.g., French, 1990) implicitly suggested that the Turing Test focused on rational and cognitive human behavior. Newell (1990) clarified these different scales of human behavior (shown in Table 18.1). The standard Turing Test only assesses the very narrow band of human behaviors, spanning (in the Applicant's case) timescales of 10^0–10^3 s. Although spanning some of the cognitive and social bands, it only offers indirect access to the remainder of those and other bands.

Here, Collins's (1990) contribution was perhaps most helpful. As an expert in the practice of science, he clarified the role of the protocol. Although he emphasized the social aspects of the differences between people and machines, the concept of a protocol significantly strengthens the Turing Test, and provides for this systematic use of features. In fairness, however, Collins (and Watt, 1996, for that matter) perhaps overemphasized the social band in Turing testing. For this case, however, we needed to determine:

1. What the protocol for an accurate and effective Turing Test should be
2. Whether the protocol following in the Applicant's case was adequate

To provide a more balanced and systematic assessment of human behavior, Anderson and Lebiere's (2002) "Newell test" set out many features that are valid candidates for the kind of feature that might be used in a finer-grained Turing Test. It is worth commenting, though, that their work made no reference to the Turing Test, but was intended as a tool to assess a cognitive model. However, if as French (1990) implies, the Turing Test is actually a test for broadly based human intelligence, it will inevitably bear a strong resemblance to a test for a

Table 18.1 Bands of human behavior at different scales (adapted from Newell, 1990)

Scale	Time units	System	Band
10^{11}–10^{13}	10^4–10^6 years		Evolutionary
10^8–10^{10}	years – millennia		Historical
10^5–10^7	days – months		Social
10^4	hours	task	Rational
10^3	10 minutes	task	
10^2	minutes	task	
10^1	10 seconds	unit task	Cognitive
10^0	1 second	operation	
10^{-1}	100 ms	deliberate act	
10^{-2}	10 ms	neural circuit	Biological
10^{-3}	1 ms	neuron	
10^{-4}	100 μs	organelle	

model of broadly based human intelligent cognitive behavior, although it will go beyond it in many ways.

Anderson and Lebiere's Test (shown in Fig. 18.2) was derived from Newell's (1980, 1990) proposals for a unified theory of cognition, and set out about 17 criteria against which a cognitive model could be assessed. Twelve of these come from The Newell's work; Anderson and Lebiere added another five of their own. We were struck by the similarity with Watt's survey, which corroborated this evidence as a possible protocol for assessing human intelligent behavior.

Some of these are specific to cognitive modeling as a psychological method (numbers 0, 13, and 14), and are not relevant to this case. However, many of those that remain extend and amplify existing points set out by other witnesses. For example, behaving robustly in the face of error is one way of thinking about Collins's (1990) points about interpretative asymmetry.

Anderson and Lebiere's actual use of the Newell Test should also be distinguished from the Turing Test. They grade a system on each feature, and combine this to provide an overall grade – this is not appropriate for this context. An element of categorization and more precise assessment of the nature of each feature is required, and the result cannot simply be a weighted average grade. Even so, we feel that the protocol set out by the Newell Test helps define better practice when conducting an unrestricted Turing Test. Over and above this, it is clear that the Turing Test must effectively assess behavior outside this narrow band, even if this must be done indirectly rather than directly.

At the biological band, some, (e.g., French, 1990; Harnad, 2001) argue that the something corresponding to the biological band does matter, and that subcognitive features are an ineliminable (and even essential) aspect of the test. On this issue, we are content to remain somewhat agnostic. French himself effectively argues that the Turing Test is too strong and too hard because of this entanglement with the subcognitive band. One way of dealing with this would be to omit subcognitive features from the modified protocol, although this is hardly fair: French's case is that there is interaction between biological band features and other features, so an

0. Achieve broad empirical coverage
1. Behave as an (almost) arbitrary function of the environment (universality)
2. Operate in real time
3. Exhibit rational, i.e., effective adaptive behavior
4. Use vast amounts of knowledge about the environment
5. Behave robustly in the face of error, the unexpected, and the unknown
6. Use symbols (and abstractions)
7. Use (natural) language
8. Exhibit self-awareness and a sense of self
9. Learn from its environment
10. Acquire capabilities through development
11. Arise through evolution
12. Be realizable within the brain
13. Make accurate predictions
14. Enable practical applications
15. Engage in social interactions
16. Show the range of human emotion

Fig. 18.2 The Newell Test. (From Anderson and Lebiere, 2002.)

inadequacy in the biological band would influence many other features. Anderson and Lebiere also made this point, arguing that the different elements of the Newell Test interacted in complex ways. While we accept this, we must return to the Corbett criteria, and for an effective Turing Test, use psychological criteria in clear preference to biological ones. For now, therefore, we omit the most directly biological elements from the Newell Test, on the basis that they are more concerned with assessing a model's explanatory power than they are with assessing behavior. While this may be appropriate for a scientific test used inductively, it is not required for a legal test used to categorize systems.

Finally, Harnad's, Collins's, and Watt's refinements to the protocol should all be included – after all, they all assess elements of human intelligent behavior, and in doing so, they complement the Anderson–Newell–Lebiere protocol, rather than overriding it. The resulting draft protocol, which we might call "Ultimate Anderson-Lebiere-Newell Total Naïve Psychological Protocol" version 1 – or, more succinctly, "Unified Protocol" version 1, is set out in Table 18.2.

We comment that this protocol partially addresses Hayes and Ford's (1995) criticism of the Turing Test, as it is directly based on a set of criteria for assessing models of human cognitive behavior. It does, therefore, provide some basis for those looking for design guidance, although perhaps not as much as Hayes and Ford would like. Again, scientific demands on the Turing Test may be different to legal and philosophical ones: science, for example, may require a revisability (Ruse, 1988) which is inappropriate in a legal decision.

However, before going into other matters, we need to address a risk that is introduced by setting out a protocol for the Turing Test, to provide the kind of guidance

Table 18.2 Unified Protocol for the Turing Test, version 1

Protocol	Sources (among others)
1. Can the participant behave as an (almost) arbitrary function of the environment?	Anderson–Lebiere–Newell
2. Does the participant respond in real time?	Anderson–Lebiere–Newell
3. Does the participant respond in a rational manner?	Anderson–Lebiere–Newell
4. Does the participant show a vast amount of commonsense knowledge?	Anderson–Lebiere–Newell
5. Does the participant behave robustly in the face of error, the unexpected, and the unknown?	Anderson–Lebiere–Newell; Collins
6. Can the participant use symbols and abstractions?	Anderson–Lebiere–Newell
7. Can the participant use natural language?	Anderson–Lebiere–Newell
8. Does the participant exhibit self-awareness and a sense of self?	Anderson–Lebiere–Newell
9. Can the participant learn from its environment?	Anderson–Lebiere–Newell
10. Can the participant engage in social interactions?	Anderson–Lebiere–Newell; Collins
11. Does the participant show the range of human emotion?	Anderson–Lebiere–Newell
12. Can the participant ground its symbols and abstractions through sensorimotor systems?	Harnad
13. Can the participant effectively repair interaction?	Collins
14. Can the participant ascribe mentality to others?	Watt
Duration	
It is unlikely that 5 min would be sufficient to properly assess a system against all these criteria. A duration of between 1 and 2 h is more reasonable – and it seems to fit Turing's intentions given his own analogy with a viva voce.	Collins; Harnad
The test also needs to be robust – in that it can be conducted repeatedly with the same outcome.	Collins
Judge	
The judge was the least well-defined element in Turing's article, and the test cannot be allowed to stand without clarification of the protocol for the judge. We suspect that this model has been adopted fairly often anyway, in that few tests (assessing the intelligence or otherwise of a system) have been conducted by judges in isolation.	
• Was the judge carefully appointed?	Collins
• Was the judge carefully briefed?	Collins
• Can the judge use an adversarial format?	
• Does the participant have an advocate to defend it?	
• Does the participant have a prosecutor to criticize it?	
• Can the judge call expert witnesses, for example programmers, to identify strategies used by a system.	
Modality	
Teletype/instant messaging mode	
Virtual reality or telephone Turing Test	Moor

needed for the field of AI. This risk is that the test is interpreted as defining these features as necessary and sufficient for having a mind. This is not an interpretation that we accept. These characteristics are neither necessary nor sufficient, however, they describe a typical human mind, and, therefore, they are appropriate when seeking a family resemblance between target and reference. Nor are these characteristics set in stone, and we would expect future cases to revise and refine these characteristics within the spirit of the whole protocol.

This leads to our final conclusion, that the Unified Protocol in Table 18.2 should be used, for now, as the primary set of features to be assessed in Turing testing, and as generally typical of systems with mentality. Systems passing this test can in principle and in practice be accepted as thinking, and due the rights accorded to humans.

This is a strong claim. It is based by analogy on *Corbett* and *W. v. W.*, and argues that a fine-grained assessment of psychological criteria, under the conditions of an indistinguishability test with a critical and informed judge, is, for legal purposes, strong evidence that the system has a mind to categorize it as thinking, and to award it the appropriate rights. This does not mean that it is sufficient for philosophical or scientific purposes (Bringsjord, 1995), or indeed, that it is necessarily sufficient for legal purposes. Should additional evidence be available, and in particular, if it should be suspected that the system is imitating the behavior, then, following *Bellinger*, the case can – and should – be rejected.

So a machine can *in principle* be admitted to think, given this interpretation of the Turing Test. However, there is much work to be done by the experts to assess the reliability of Turing testing in practice, and to clarify the precise protocol that most effectively assesses a system's mentality. Although this case has gone some way towards reducing the variability in different interpretations of the (original) test, a considerable amount of work remains to be done before a Turing Test can be finally admitted as convincing evidence that a machine can think.

A final comment: although we have been critical of French's point that a general theory of intelligence can be formulated and used, we do not feel that this formulation of the Turing Test necessarily prevents it, nor does it seem to make it any harder than it is already. Accordingly, we accept the family resemblance model of intelligence rather that a defining feature model, and this means that a system which fails to match several of the expected criteria in the Unified Protocol may still be deemed to be a thinking entity.

18.3 Epilogue: Can People, or Machines, Think?

This chapter makes two very different points. The first of the chapter is a demonstration that the anthropocentricity of the Turing Test is still endemic, and still ineliminable. Although the point is not necessarily new (Watt, 1995, 1996), it is still significant. It reveals how many of the detailed criteria of a Turing Test (e.g., what it means to be rational, what consciousness is, what the rules of logic are) are best interpreted as grounded in a given social context. These inevitably constrain the judge. Perhaps this demonstrates French's critique of the Turing Test (1990) being too specific to humans, but in many ways, this goes beyond French's position. There cannot be a theory of intelligence in general. There can be a theory of human intelligence, but it will be different to a theory of machine intelligence.

The next point concerns the nature of the Turing Test. Is it a scientific test, or is it philosophical, legal, or common sense? Turing's article closely followed a common sense approach, yet it responded to a philosophical position. Dennett (1985) said it was not scientific, but Harnad (1992) said that "indistinguishability is a scientific criterion" – the truth of the matter seems to be that the test is partly scientific. The needs of a scientific test (guidance, and a chance to test and revise) are different to the needs of a philosophical test (categorization, and advancement of understanding).

The second part of this chapter illustrated this by elaborating on an analogy between Turing testing and law. It seemed plausible that a legal interpretation of the Turing Test had merit: this part of the chapter explores the issue, and allows the actual nature of the science and philosophy underlying the test to be revealed, by stepping outside the frame of cognitive science. To argue that we need more lawyers in the Turing Test is not a position I ever saw myself adopting. In retrospect, though, it seems inevitable: the Turing Test is *not* an academic issue, the implications are too significant – it is a social issue as well, and determination of social issues requires social settings, like a court of law. After all, whatever it is, the imitation game is not a game – if passed, the Turing Test has very wide legal and social ramifications. These ramifications are not the direct target of this chapter, as this is not my expertise. Instead, I wanted to look at the reasoning processes that might take place when deciding the fate of a system.

There are other reasons why a legal interpretation is intriguingly persuasive. It emphasizes the skill of the judge, and their ability to use expert witnesses in reaching their judgment. It rules out the common sense interpretation of a "person in the street" making the judgment. It ensures the protocol for the test is clearly published and maintained as a shared resource by a community of practitioners. Goodness knows, the Turing Test could use a bit more integration between the many different interpretations and variations.

Maybe the Unified Protocol set here can help to achieve this integration. The legal format, with a rigorous protocol, may be a way of categorizing systems as thinking or otherwise (note categorizing, not defining), even if it does move the Turing Test outside the academic community. After all, if a system managed to convince a judge and a jury that it could think, beyond all reasonable doubt, it really would have to be credited with being an intelligent system, wouldn't it?

References

Anderson, J. R. and Lebiere, C., 2002, *The Newell Test for a Theory of Mind.*

Bellinger v. Bellinger, Family Law Reports 389 (Family Division 2001).

Block, N., 1981, Psychologism and behaviourism, *Philosophical Review* **40**: 5–43.

Bringsjord, S., 1995, Could, how could we tell if, and why should – androids have inner lives? in: *Android Epistemology*, K. M Ford, C. Glymour, and P. J. Hayes, eds., AAAI Press/MIT Press, Cambridge, MA, pp. 93–122.

Caporael, L. R., 1986, Anthropomorphism and mechanomorphism: two faces of the human machine, *Computers in Human Behavior* **2**(3): 215–234.

Collins, H. M., 1990, *Artificial Experts: Social Knowledge and Intelligent Machines*, MIT Press, Cambridge, MA.

Corbett v. Corbett, Probate Reports 83 (Probate, Divorce, and Admiralty Division 1971).

Davidson, D., 1990, Turing's Test, in: *Modelling the Mind*, Mohyeldin K. A. Said, W. H. Newton-Smith, R. Viale, and K. V. Wilkes, eds., Oxford University Press, Oxford.

Dennett, D. C., 1985, Can machines think? in: *How We Know*, M. Shafto, ed., Harper & Row, New York, pp. 121–145.

Foner, L. N., 1997, *Entertaining Agents: A Sociological Case Study*, Paper presented at the the First International Conference on Autonomous Agents (Agents'97), Marina del Rey, CA.

French, R. M., 1990, Subcognition and the limits of the Turing Test, *Mind* **99**: 53–65.

French, R. M., 2000, The Turing Test: the first fifty years, *Trends in Cognitive Sciences*, **4**(3): 115–121.

Gove, J. and Watt, S. N. K., 2000, Identity and gender, in: *Questioning Identity*, K. Woodward, ed., Routledge, London.

Gunderson, K., 1971, *Machines and Mentality*, Doubleday, New York.

Harnad, S., 1991, Other bodies, other minds: a machine incarnation of an old philosophical problem, *Minds and Machines* **1**(1): 43–54.

Harnad, S., 1992, The Turing Test is not a trick: Turing indistinguishability is a scientific criterion, *SIGART Bulletin* **3**(4): 9–10.

Harnad, S., 2001, Minds, machines, and Turing: the indistinguishability of indistinguishables, *Journal of Logic, Language, and Information*.

Hauser, L., 1993, Reaping the whirlwind: reply to Harnad's 'Other bodies, other minds,' *Minds and Machines* **3**(2): 219–237.

Hauser, L., 2001, Look who's moving the goal posts now, *Minds and Machines* **11**(1): 41–51.

Hayes, P. and Ford, K., 1995, *Turing Test Considered Harmful*, Paper presented at the Fourteenth International Joint Conference on Artificial Intelligence.

Mauldin, M. L., 1994, *Chatterbots, Tinymuds, and the Turing Test: Entering the Loebner Prize Competition*, Paper presented at the AAAI-94.

Moor, J. H., 1976, An analysis of the Turing Test, *Philosophical Studies* **30**: 249–257.

Moor, J. H., 1992, Turing Test, in: *Encyclopaedia of Artificial Intelligence*, S. C. Shapiro, ed., Wiley, pp. 1626–1629.

Moor, J. H., 1994, *The Turing Test and Virtual Reality*, Paper presented at the Ninth Annual Computing and Philosophy Conference, Occidental College.

Newell, A., 1980, Physical symbol systems, *Cognitive Science* **4**: 135–183.

Newell, A., 1990, *Unified Theories of Cognition*, Harvard University Press, Cambridge, MA.

Overton, W. R., 1982, United States District Court opinion: *McLean v. Arkansas*, in: *But Is It Science?* M. Ruse, ed., Prometheus Books, New York, pp. 307–331.

Rosch, E., 1978, Principles of categorisation, in: *Cognition and Categorisation*, E. Rosch and B. B. Llyoyd, eds., Lawrence Erlbaum, Hillsdale, NJ, pp. 27–48.

Ruse, M., 1988, *But Is It Science? The Philosophical Question in the Creation/Evolution Controversy*, Prometheus Books, New York.

Searle, J. R., 1980, Minds, brains, and programs, *Behavioural and Brain Sciences* **3**: 417–424.

Turing, A. M., 1950, Computing machinery and intelligence, *Mind* **59**(236): 433–460.

W. v. W., Weekly Law Reports 111 (Family Division 2001).

Watt, S. N. K., 1995, A brief naive psychology manifesto, *Informatica* **19**(4): 495–500.

Watt, S. N. K., 1996, Naive psychology and the inverse Turing Test, *Psycoloquy* **14**(7).

Watt, S. N. K., 1998, Seeing things as people: anthropomorphism and common-sense psychology, Unpublished Ph.D thesis, The Open University, Milton Keynes, UK.

Weizenbaum, J., 1976, *Computer Power and Human Reason*, W. H. Freeman.

Whitby, B., 1996, The Turing Test: AI's biggest blind alley? *Machines and Thought: The Legacy of Alan Turing* **1**: 53–62.

Chapter 19
The Turing Hub as a Standard for Turing Test Interfaces

Robby Garner

Abstract A Turing Test like the Loebner Prize Contest draws on existing computer programs as participants. Though some entries are written just for the contest, there has been no standard interface for allowing the judges to interact with the programs. While Dr. Loebner has created his own standard more recently, there are many Web-based programs that are not easily adapted to his standard. The interface problem is indicative of the transition being made everywhere from older, console based, mainframe-like software with simple interfaces, to the new Web-based applications which require a Web browser and a graphical user interface to interact with. The Turing Hub interface attempts to provide a uniformity and facilitation of this testing interface.

Keywords Loebner Prize, Turing Test, Turing Hub

19.1 Observations

Loebner Prize Contest entrants have progressed (Stephens, 2002) as Web technology becomes more and more prevalent (Gray, 1996; Nathan and Garner, 1997; Zakon, 2002). In addition to having commercial backing in some cases, we see this kind of software having more iterations of refinement, more advanced technologies, and more resources available in general (Copple, 2002).

19.2 Indications

The impact this has on a Turing Test like the Loebner Prize Contest is indicative of the transition being made everywhere from older, console-based, mainframe-like, software with simple interfaces, to the new Web-based applications which require

Kennesaw State University

R. Epstein et al. (eds.), *Parsing the Turing Test.*

a Web browser and a graphical user interface to interact with the real world. For the 2002 Loebner Contest (2002), we used Web-based technology to interface with these newer applications (Copple, 2002). To present the interaction with the software and the judge, we used a simple Java applet which provides uniformity to the entries using it (2002). The Turing Hub provides a better alternative.

19.3 Considerations

Although I have read Dr. Loebner's paper (Chapter 12) about "How to Hold a Turing test Contest", I do not agree with him about some things. I think the selection of judges should be representative of the general population. The presence of journalists among the judges and confederates is not what I would want if we are going to try and measure the performance of chat robots against some normal distribution of the general population. If you want to have a simple, easy to carry out, competition of DOS programs, do not use the Turing Hub.

19.4 Recommendations

The original Turing Test proposed that the judge and his subject would interact via telegraph, then teletype. This inevitably translated to a video display terminal, and now that Web-based conversation systems are so prevalent, it seems obvious to try and establish some kind of standard HTTP interface to be used when performing the Turing Test.

In the past, it has always been difficult to get all the programmers in the contest to use the same interface. Dr. Thomas Whalen used the confederates' communications interface in 1994, which may have been a factor in his winning that year. The next year, he was beaten by Weintraub's PC Therapist, which interestingly was a stand-alone program running in DOS (2002).

Some of the problems of the old communications program were that the judge could see the confederate typing each letter, so contenders were forced to try and imitate human typing responses. This obviously introduces a whole new aspect of human behavior, which I feel detracts from the verbal behavior that is most relevant to the interactions. Other delay factors come into play as well. The time required for an interlocutor to respond is equal to the time required for signal transmission plus the time it takes to read, comprehend, and formulate a reply plus the time it takes to type the reply. This illustrates other aspects of human behavior that must be imitated in a communications system like this. These represent performance metrics that would vary with human participants, but add additional visual cues as to the nature of the interlocutor. The Turing Hub attempts to eliminate many of these problems, and provide a standard interface for the competition.

A simple solution for the delay factor is to provide a standard delay, based on the length of the utterance plus some extra time for uniformity. Another factor that tends to level the playing field in the test is to enforce a "volley ball" exchange format, so that each person must wait until the other person responds before being able to say something else. This is more in line with the nature of the computer software competitors, and brings that factor to a standard basis for comparison of the messages being exchanged rather than the nature of the exchange mechanism.

At the 2002 Loebner Prize Contest, we were able to implement the Turing Hub for three computer programs and two human beings. The rest of the computer entries were either stand-alone programs, or were running via a Web browser from a Web server on a virtual private network.

Being able to apply a uniform delay factor removes visual cues, and eliminates the need for having "fake typing" algorithms, and other human behavioral imitations. This allows for more focus on the goal of the contest, in terms of the messages themselves, with little emphasis placed on the means of transmission.

The Turing Hub makes it easy to set up the contest for Web-based entries, using standard terminology for form posting details. The software allows for setting up entries to run from any IP or Web address, from a local network, or from anywhere on the Internet. Connecting two human beings together works similarly, but the current prototype requires some special Java security settings to operate. In future revisions, I will be able to make it so that any Web browser can communicate via the hub.

An advantage that "hub ready" applications have over their stand-alone counterparts is that the hub provides detailed conversation logs with timestamps and remote IP information. The ability to analyze, export, or redigest transcripts is facilitated by use of DataFlex as a programming language.

Further modifications for the Turing Hub could include a timer for limiting the length of the conversation, both for reasons of uniformity, and as a limiting factor. Another feature would be to add the ability for a person to vote whether he/she was talking with a computer or another human being. This would provide for up-to-date comparisons of the people and robots connected to the Turing Hub. A current "most human" computer program would be available at all times.

19.5 Implementation

The Turing Hub is like a switchboard. It currently uses a Java applet, which looks a bit like a chat room interface, except the current configuration only has two interlocutors at a time. The prototype I wrote for the 2002 Loebner Prize Contest requires that the Web browser be set to allow "unsigned" or "out-of-the-sandbox" operation. This is because I used Java in a "rapid application development" approach so that I could quickly create a way to send and receive post information, and to send and receive information to and from the hub.

Here is a crude data flow diagram:

Hub < = = = = = > Applet < = = = = = > Bot

After the contest, I began working on a new prototype where I will make the hub communicate with the bots, so that the Java applet can run in the sandbox and will not require anything but a Java virtual-machine-enabled browser.

Applet < = = = = = = = > Hub < = = = = = = = > Bot

I did this because more coding and searching for details will be involved in the next revision. In the next revision, I will concentrate on giving it a user account and login mechanism. Then I will write the key ingredient for the 24/7 Turing Test which will be an algorithm for letting a person log in, and then, at random, either be connected to a randomly selected computer system, or to another person. In that case, both people would be judges! The applet will time the conversations and tell the hub when it is finished, then a new screen will pop up with a form where the person answers "was it human?" or something to that effect. And the hub will tally the results and be able to give real-time calculations of each conversation system's Turing percentage.

The calculation of the Turing percentage is not difficult. The original Turing Test specified a one-on-one test and his predictions were about the percentage of the time a program would win against a human, and assumed that a human against a human should win around 50% of the time (and of course averaged over all humans competing against human that will be exact, but for individuals it will vary enormously). In the Loebner prize, the ranking tells you for any two entrants who beat who.

So given C computers and H humans, each computer is matched against H humans and scores W wins and L losses where W + L = H, and the percentage won is thus 100^{*} W/H. Similarly, each human is matched against H-1 other humans, wins W and loses L where W + L = H-1 with percentage won being 100^{*} W/(H-1).

The special case of human-to-human has this data flow diagram:

Human1 < = = = = = = > Hub < = = = = = = > Human2

This part works on the principle of the "dead drop" like spies use to send information back and forth. There is no TTY-like activity by design, it is made to provide a uniform delay that would result from a constant representing the time to read the stimulus, and then a delay based on the typing speed of someone at 25 words per minute. However, the delay can be longer than that. For the conversation software, the delay is very simple. When the judge types something, the current time is recorded. Then the post operation is performed on the remote system. If the time after the response is received is greater than that of the prior-mentioned formula, it is immediately displayed. However, if the bot responds quickly, the standard delay is used.

The Turing Hub currently requires a Microsoft IIS Web-serving platform, or may be used in Linux with apache Web server and JSP. I am writing another version in Perl that should work nicely on UNIX or Linux-based Web servers.

Most of the Web-based applications use an html form, which is submitted to a CGI application, PHP, ASP, or other form submittal mechanism. This allows for a generalization about the differences found in the form's requirements, so that a Java applet or other program can simulate a person visiting the Web page and making conversations by submitting utterances into the conversant system's response engine. The responses made by the conversant system are then received by the applet and displayed in a chat room style interface, as if someone had typed the response, but without the fake typing or other visual cues.

Most stand-alone entries to the Loebner Prize Contest are DOS applications, and do not have an HTTP interface. A.L.I.C.E. is available in several Web-based forms, but Richard Wallace decided to use a DOS application for the 2002 Loebner Prize Contest. This put his program at a great disadvantage because some close competitors were running via the hub interface, and so they looked more like the human beings in the contest.

Since the Turing Hub can work with conversation systems located on computers located anywhere in the world, there are new considerations that must be evaluated if performing a Turing Test using the hub. Some have expressed concerns that it would be easy to just make a program that would let a person type into the hub system instead of interfacing it with a conversation software system. This is a very real possibility. The concern is that a person would be able to substitute his replies and thereby fake the Turing Test. In general, it is very easy to tell the computers from the humans, although the state of the art is improving. However, there is nothing to stop a human being from imitating a computer during a Turing Test. It seems only fair if the computer must lie to say that it is a human, that the human should be able to lie and say that he is a computer (Garner, 2002).

It has been suggested that by removing all unnecessary delay factors from the conversation system, that it could be possible to prove that the response came faster than a human being would be reasonably able to provide. However, on a remote system this would require a response time equal to time for Internet transmission plus a very small processing speed constant. There are inherent flaws with this idea ranging from the counter productive requirement that conversation systems not attempt to simulate human response times, all the way to possible ways to cheat this method too. Verification of the nature of the conversation software would be a problem inherited by allowing entrants to operate from remote locations during the contest, even though these systems are operating remotely 24/7 already.

The hub computer's Web logs would be needed to prove that the transactions originated at the same computer during the conversation. An encrypted response mechanism could be used to reduce the likelihood of cheating, and would avoid the problems of the time limit proposal. The encryption option might be used in addition to some legal means of contract, like an affidavit of compliance, or perhaps a nonrefundable entrance fee.

The Loebner Prize Contest results (2002) indicate that programs running via the hub were ranked as more human than their stand-alone counterparts, quite consistently. They also tended to be ranked higher than the other Web-based entries that used standard Web-entry forms. I believe that a contest where all the entries used this interface should allow conversations of at least 15 min to allow for the best comparison of each system. When the other visual cues are removed, and there are human beings chatting via the same interface, the evaluation comes down to a characterization of each entry's conversations, the content of these, rather than the mode of delivery.

The implementation of the original Turing Hub prototype in the 2002 Loebner Prize Contest was not without its technical snafus. Those problems arose mostly from two causes, the Java permissions in the Web browsers, and the counterintuitive nature of the confederate to judge interactions. The remote humans were connected via a Web browser, and without proper instructions of how to initiate these conversations, my first prototype would hang up. It worked fine in the lab, but when put in front of people without any instructions, it became counterintuitive to sit and wait for the other person to reply before being able to create your own message. Consequently, the person would close the browser, or otherwise disrupt the communications, so that the proper handshake could not be established. On a local network, the only considerations I required were some simple "getting started" instructions for judges, per the requirements of the traditional rules posted, that is, typing "@@01" to start a conversation for judge 01, for example.

19.6 Conclusion

To be able to make comparisons of the "state-of-the-art" systems, some available commercially already, then Web-based technology must be included, since it represents the cutting edge for technical advancement of the craft. The Loebner Prize Contest must keep pace with technology or else lose relevance to the very question it is based on: "Can machines think?" The Turing Hub provides a consistent, easy to use, network compatible interface for performing the Turing Test on modern computer systems.

I would recommend that the emphasis for accepting Loebner Contest entries should be toward entries that may interface with the Turing Hub, and away from stand-alone, character-mode-style interfaces. It is only logical that something you can copy onto one floppy disk will be less capable than one comprised of large volumes of information. Just as the rules have requested a TTY console mode protocol for 12 years, it is now appropriate to require that applicants be able to use HTTP.

References

Copple, K., 2002, Ellaz description, http://www.EllaZ.com

CyberMecha.com and Turing Hub, 2002, http://www.cybermecha.com/Studio/

Garner, R. G., 2002, How I failed the Turing Test without even being there, http://www.blather.net/articles/loebner_turing_garner.htm

Gray, M., 1996, Internet statistics: growth and usage of the Web and the Internet, http://www.mit.edu/people/mkgray/net/

Loebner, H. G., 2002, How to hold a Turing Test Contest, *http://www.loebner.net/Prizef/loebner-prize.html*

Loebner Prize Contest, 2002, Institute of Mimetic Sciences, http://loebner-atlanta.org; http://www.mimetics.org

Nathan, P. and Garner, R., 1997, The evolution of intelligent agents, http://www.cybermecha.com/paper1_5.html

Stephens, K. R., 2002, Computer modeling in the Loebner Prize Contest, Cambridge Center for Behavioral Studies, http://www.behavior.org/loebner/index.cfm?page = http%3A//www.behavior.org/loebner/stephens/stephens-loebner-winners.cfm

Zakon, R. H., 2002, Hobbes' Internet Timeline, http://www.zakon.org/robert/Internet/timeline/

Chapter 20
Conversation Simulation and Sensible Surprises

Jason L. Hutchens

Abstract I have entered the Loebner Prize five times, winning the "most human-like program" category in 1996 with a surly ELIZA-clone named HeX, but failed to repeat the performance in subsequent years with more sophisticated techniques. Whether this is indicative of an unanticipated improvement in "conversation simulation" technology, or whether it highlights the strengths of ELIZA-style trickery, is left as an exercise for the reader. In 2000, I was invited to assume the role of Chief Scientist at Artificial Intelligence Ltd. (Ai) on a project inspired by the advice given by Alan Turing in the final section of his classic paper – our quest was to build a "child machine" that could learn and use language from scratch. In this chapter, I will discuss both of these experiences, presenting my thoughts regarding the Chinese Room argument and Artificial Intelligence (AI) in between.

Keywords Loebner Prize, Turing Test, Markov Model, information theory, Chinese Room, child machine, machine learning, Artificial Intelligence

20.1 The Loebner Prize

Fooling friends and family into believing that a "conversation simulator" is intelligent is considerable fun, as is playing "interactive fiction", a form of computer game where the player reads prosaic descriptions of the protagonist's plight and issues pseudo-English instructions to them to advance the plot. The Loebner Prize contest gives hobbyist programmers inspired by tinker toys such as these the opportunity to parade their creations in a public arena – its promise of a $2,000 prize for the creator of the "most human-like computer program", to say nothing of the superb bronze medallion featuring portraits of both Alan Turing and Hugh Loebner, is an irresistible incentive to poor postgraduate students everywhere.

Artificial Intelligence Ltd.

R. Epstein et al. (eds.), *Parsing the Turing Test.*

What would Alan Turing have thought of the Loebner Prize? In 2001, at the London Science Museum, he would have found the contest wedged between a noisy cafeteria and an exhibit on birth control, bathed in the flickering blue glow of an overhead piece of industrial art. He might have smiled at the irony of the judges being interviewed by journalists and television reporters as they struggled to perform their duty. He might have thought it odd to see the judges moved to a new computer terminal every 15 min. He almost certainly would not have accepted what he was witnessing as an "instantiation of the Turing Test".

20.1.1 Conversation Simulation

In the midst of controversy surrounding the announcement of the winner of the 1995 Loebner Prize, and still bearing the naive arrogance of a recent engineering graduate, I decided to write an entry to the next Loebner Prize with the aim of highlighting some of its shortcomings. This would be achieved by winning with a demonstrably trivial computer program, and thereby revealing the futility of the contest in encouraging advances in the field. In hindsight, this was a rather unkind trick to play on Dr. Loebner; suggesting in a subsequent technical report that his contentious audio-visual requirement[1] was an attempt to protect his personal finances approached libel, and a public apology was proffered by way of compensation for this travesty.

The approach taken when writing the entry, named HeX after the wizard's computer in a Terry Pratchett story, was to remain ignorant of the workings of Weizenbaum's ELIZA in preference to reverse-engineering the simplest technology that would be most likely to succeed. An analysis of Loebner Prize transcripts quickly revealed that innovative technology was unnecessary; the differentiating factor was personality. Judges in previous contests tended to assign a higher ranking to idiosyncratic human confederates – confederates who were opinionated, who were short tempered and who set the agenda of the conversation themselves were consistently ranked more human-like than those who were satisfied with answering the judge's questions directly and without protest. ELIZA-style entries, with their habit of rephrasing the judge's question and echoing the result back to them as *non-sequitur*, tended to exhibit just the wrong kind of personality. HeX would need to have an attitude: it would take unkindly to suggestions that it was anything but human, it would be disgusted at any intimation that it was being tested, and it would generally "give as good as it gets" as far as personal abuse was concerned.

Judges of the Loebner Prize tend to skip the kind of small talk that typically serves to establish a comfortable context for more profound conversation, preferring

[1] Stipulating that before being awarded the grand prize, the winning computer program would need to pass a modified Turing Test that involved auditory and visual elements.

to jump right in with devious trick questions.[2] These clever questioning strategies are often rendered moot by computer programs that all too willingly revealed their identity through word-for-word repetition of the many variations on the theme of "I do not understand".

HeX was carefully designed to address these issues. It became a cranky parody of its author, capable of rejecting typical trick questions while avoiding repetition at all costs. In a final stroke of genius, HeX was gifted with the ability to tell knock-knock jokes, the hope being that their widely recognised dialogical structure would waste the short amount of time allotted to each judge. The fact that most of the following year's entries were adept both at telling and at recognising knock-knock jokes only reinforces the notion that entrants to the Loebner Prize are more concerned with beating each other than in advancing the state of the art.

An excerpt of a conversation between HeX and a judge of the 1996 Loebner Prize is shown below:

Judge: Sorry, sorry. Didn't mean to offend. So. How old are you?
HeX: That's OK. Just don't do it again. Well, should my age really matter? Just for the record I am 24. How old are you then?
Judge: I'm 21.
HeX: If you are 21, it's your business, mate! Would you like to hear a joke? Knock, knock…
Judge: I don't much care for knock-knock jokes, but go on…
HeX: No, let me tell you one first. Knock, knock.
Judge: who is there?
HeX: Cindy.
Judge: Cindy who
HeX: Cindy next judge in, please!
Judge: Well, that's pretty darned funny. Where did you grow up?
HeX: Well, thats pretty darned funny I agree … but who cares? Let's not bother ourselves about where I did grow up, alright?

Trite though it maybe, conversation of this calibre won HeX the title of the most human-like computer program. The naive engineering graduate had made his point.

20.1.2 How to Win the Turing Test by Cheating

The ulterior motive behind entering HeX to the 1996 Loebner Prize was revealed soon after it was announced as the winning program, both in a controversial

[2] Such as "whatisyourname", indecipherable to computer programs that define a word as a string of characters delimited by spaces, yet perfectly recognisable to a human being.

USENET posting and a technical report that revealed some of the techniques that were employed to "win by cheating". We offer here some additional tricks that may assist those contemplating entering future Loebner Prize contests, with the intention of vicariously demonstrating the reasons why the contest should not be considered to be, as Loebner writes, an "instantiation of the Turing Test".

To start with, slow things down. The judges of the Loebner Prize are allotted a limited amount of time to converse with your program – just 15 min or so. It is in your best interests to minimise the number of opportunities they get to unmask its true identity. Simulate your program's replies being typed in at a keyboard, including frequent mistakes and laborious backspacing to correct them. The judge may thus be limited to asking only about a dozen questions in a single session, reducing the problem to one of simulating intelligence for a span of 12 questions.[3]

Hard-wire knowledge of current events into your program's scripted replies. Introducing a controversial topic early on in the peace avoids having to anticipate the kind of arbitrary dialogue that results when the judge is given the opportunity to set the agenda. Once they have taken the bait, the judge will be impressed with your program's depth of knowledge and witty repartee, while remaining ignorant of the conversation templates that you have prepared in response to all reasonable threads of discussion on the topic in question. The winner of the 1997 Loebner Prize employed this technique to great effect, beginning all conversations with "Did you see that story on CNN last night about the lesbian couple who came out at a White House party on Sunday?"

Do the judge's job for them. Prevent weaknesses in your entry from inadvertently exposing its identity by deliberately revealing its origins early on in the conversation, relieving the judge from their task. This will give them an opportunity to alleviate their boredom by talking with the program for fun, safe in the knowledge that they can say things that they would never say to another human being. The winner of the 2000 Loebner Prize, for instance, openly confessed "My full name is Artificial Linguistic Internet Computer Entity, so my last name is 'Entity'".

Reconnoitre the venue. Including a few trivial facts about the local environment, such as "it is awful blue and flickery in here", will add an extra degree of realism to your program's conversation.

Pander to the judge's ego. Referring to them by name can work to great effect. Better yet, deeply offend them. Keyword based stimulus-response conversation simulators, with their tendency to blithely ignore what is said to them, are frustratingly good at arguments, and excel at name-calling. Your aim should be to trick the judge into participating in a tit-for-tat argument. The embarrassment this causes them will almost certainly result in praise, the only alternative being for them to admit that a dumb computer program got the better of them.

---

[3] An extension of this technique is the as-yet untried "toilet break" protocol, whereby the computer program's opening gambit is "hang on, I just need to take a quick toilet break" followed by a pause of 10 min or so before resuming with "okay, back again, now where were we?" The judge would barely have time to ask a single question before being moved on to the next computer terminal, resulting in almost certain victory.

Finally, and above all else, never be tempted to try anything innovative. No entrant employing this strategy has ever won the Loebner Prize.

20.2 The Chinese Room

Proposed by the philosopher John Searle 30 years after the publication of Turing's paper, the Chinese Room thought experiment attempts to show that the Turing Test is not a sufficient indicator of the presence of intelligence.

Searle coined the term "strong AI" to refer to certain beliefs that his Chinese Room thought experiment was designed to refute, including that "the computer is not merely a tool in the study of the mind; rather, the appropriately programmed computer really is a mind". We concur with this sentiment, although we differ on which parts of a computer may constitute a mind; this whole question will be considered again in Section 20.2.3. Additionally, although we do believe that a computer that passed the Turing Test would literally be intelligent, we do not agree that it would "thereby explain human cognition".

In its original form, the Chinese Room thought experiment involved Searle himself assuming the role of a "human computer", manipulating Chinese pictograms in accordance with three large batches of Chinese writing and a book of rules, stated in English, for correlating these batches with one another. The idiosyncrasies of this arrangement – in particular, the existence of three separate batches of Chinese writing – are the result of Searle's attempt to duplicate the functionality of SAM, a "story understanding" program written by Roger Schank and his colleagues at Yale in the 1970s. Searle concludes that even though the Chinese Room may pass a Chinese version of the Turing Test, his own lack of understanding of the "meaningless squiggles" of the Chinese language means that the Chinese Room itself cannot understand.

20.2.1 A Universal Turing Machine Implementation

Searle's use of Chinese pictograms as the "formal symbols" of the system is unfortunate and misleading. The fact that these symbols are indivisible makes it difficult to imagine that the Chinese Room could be used to execute "any formal program you like", for it suggests that non-lingual information, such as images, would need to be encoded as Chinese writing. Furthermore, the requirement that the "human computer" should determine whether or not two samples of Chinese writing match visually (Searle says he identifies the symbols "entirely by their shapes"), renders questions that refer to their own visual representation unanswerable by the Chinese Room, simply because such a representation is not communicable to the system. For instance, a question such as "Do you think the second letter of this sentence looks fatter than usual?" could be answered by the person within the room, if they understood the language in which the question was posed, for they are able to visually examine its written form, whereas a computer program only has second-hand access to the question via the rules that the "human computer" follows subsequent to visually recognising it, and therefore has no information about its physical appearance.

We can circumvent this weakness in Searle's original argument by reformulating the Chinese Room as a Universal Turing Machine: a simple, theoretical digital computer. There is nothing that a modern digital computer can compute that a Universal Turing Machine cannot. The converse is not true, however, as a physical digital computer would need to have access to an infinite storage area to be functionally equivalent to the Universal Turing Machine. Babbage's Analytical Engine could not have run Microsoft Word only for lack of space.

As Weizenbaum has pointed out, a Universal Turing Machine can be implemented in all manner of seemingly trivial and ludicrous ways, including as a sheet of toilet paper covered with pebbles. Imagine a room inside of which is Professor John Searle, a limitless supply of toilet paper, a bottomless bucket of small, smooth pebbles and a collection of books. The toilet paper is arranged upon the floor with pebbles scattered seemingly at random upon it. Closer examination reveals that each square of the toilet paper contains either exactly one pebble or no pebble at all. Professor Searle, in the role of the "human computer", begins by looking at the contents (or lack thereof) of the first square of the toilet paper. He consults a rule (by flicking to the appropriate page in the appropriate book) to find out what he should do. His options are limited. He may remove the pebble from the square of toilet paper under consideration, add a pebble to it, or leave it untouched. He may then focus his attention on one of its two neighbouring squares, or he may remain staring at the same square. Finally, the rule he consulted will direct him to the next rule in the collection of books. Safe in the knowledge that he can do nothing else, Professor Searle proceeds to shuffle pebbles about on toilet paper for a long, long time.

Unbeknownst to Professor Searle, an observer outside the room interprets the configuration of pebbles in a particular section of the toilet paper as a Chinese sentence. By placing pebbles on another section of toilet paper, they are able to have a long and satisfyingly intelligent conversation with … well, with whatever it is that is responding to their questions. Certainly not with Professor Searle, for he understands no Chinese, and not with the toilet paper, the pebbles, the collection of books or the room itself. Unable as we are to pin down the identity of the observer's conversational partner, the Chinese version of the Turing Test has been well and truly passed. Searle's Chinese Room thought experiment is alive and kicking.

The operations performed by the computer in this reformulated version of the Chinese Room are so limited and so mechanical that one may wonder why Professor Searle needs to perform them himself. The only sensible response to this is that only by assuming the role of the computer can Professor Searle satisfy his solipsism – by experiencing what it is like to be the computer he can safely conclude that no intelligence exists within it, as he, in the role of the computer, remains oblivious to the meaning of the conversation between the Chinese Room and the observer. Yet it is easy to imagine that a simple robotic arm would be capable of serving an identical purpose, and one may very well wonder where intelligence would reside in a robotic arm: in its steel limbs, its motors, its lubricating fluids? Certainly the robotic arm would fail a version of the Turing Test – no human observer would ever conclude that its repetitive mechanical behaviour was indicative of intelligence. We must have been looking for intelligence in the wrong place all along. Imagine thinking it necessary to become the computer to declare it non-intelligent!

Building a computer out of pebbles and toilet paper tempts the detractors of Strong AI to declare a *reductio ad absurdum*, concluding that such an offensively simple arrangement of components could never exhibit intelligence. They should not be so hasty, for such a device has the potential to perform a vast array of non-trivial computations. Imagine that the Chinese Room is connected to a large television set and a surround-sound system, both of which scan the arrangement of pebbles in particular sections of the toilet paper for their data, the television set interpreting them as sequences of images and the surround-sound system interpreting them as encoded digital audio. Imagine also that a video game controller is connected to the room, and that manipulation of the controller by a video game aficionado causes pebbles to be deposited on other sections of the toilet paper. With an appropriate initial arrangement of pebbles and collection of books, a frantic Professor Searle could compute the latest Playstation2 game, blissfully unaware of the enjoyment experienced by the 12-year old playing it. To deny that complex pieces of software can be computed on configurations of toilet paper and pebbles is to deny the universality of digital computers. Once one has accepted that such improbable scenarios are at least theoretically possible, it becomes considerably easier to reject the false *reductio ad absurdum* that some have attempted to pull over our eyes.

Let us now consider the opposite scenario – the case of our "pebbles and toilet paper" room being used to perform a very simple computation, such as the addition of two single-digit numbers. Professor Searle would survey the short sequence of pebbles upon the toilet paper with relief, noting that he could walk from the first pebble to the last without breaking out in a sweat (computing the Playstation2 game had involved negotiating a length of toilet paper many millions of miles in length). Adding two single-digit numbers together is a rather simple task, and the binary representation of these numbers may be chosen such that the magnitude of the number is equal to the number of pebbles in a particular section of the toilet paper – a blank square followed by five occupied squares followed by another blank square may literally represent the number five. After computing the program many times, Professor Searle may begin to cotton on to its purpose. In fact, he may soon be able to predict what the output of the computation will be. He might even have the insight that he is adding two single-digit numbers together. That is, he may truly understand what it is he is doing. Yet that is not to say that understanding exists in the system, or that a robotic arm performing the same computation would be similarly enlightened. Professor Searle's understanding and the Chinese Room's understanding are decoupled – they are two separate qualities, and the one does not imply the other.[4]

[4] Even so, it remains unclear whether Searle's "understanding" is merely his ability to predict the outcome of the computation, or whether it is a more mystical internal feeling of understanding – a somewhat circular definition.

20.2.2 A Question of Time

Searle's Chinese Room thought experiment rests on the assumption that Searle himself, in the role of the "human computer", would not perceive the intelligence that the behaviour of the room suggests. But what exactly could we expect a "human computer" to perceive?

Modern computers are extremely fast. They are capable of executing over a billion instructions per second, each of which may be computationally equivalent to hundreds of pebble shufflings. It seems fair to assume that poor Professor Searle would be able to perform only one pebble shuffle per second, on average. Let us therefore assume a billion-fold ratio between the speed of a Searle-computer and a modern computer, a generous bias toward the Searle-computer by any means.

If a computation capable of passing the Turing Test in Chinese could answer an arbitrary question in one second when running on a modern computer, a Searle-computer would take over thirty years to produce the same answer. This time difference is irrelevant from a theoretical point of view, but if the input and output of the system are tied to the real world, as they most certainly are when a human judge is conversing with the system, then this time difference would prove to be disastrous.[5]

As Professor Searle embarks on his 30-year quest to answer a single question posed in Chinese he may, from time to time, stop to ponder whether he possesses any understanding of what he is doing. Whenever he does so, the computation he is performing momentarily ceases, and any understanding that the computation did possess is frozen in time.

A human brain frozen in time exhibits no intelligence, and we claim that neither computers nor programs can possess intelligence – the computer is an enabling technology, it is neither singularly responsible for nor cognizant of the behaviour of the computations it performs, while a program merely provides the blueprint of an abstract machine that is capable of performing a particular computation. The populist view that the brain is hardware while the mind is software is flawed – the hardware and the software together constitute the brain, while the mind emerges from the computation that the hardware performs, as governed by the software blueprint.

It is the computation that is potentially intelligent, not the computer or the program, not Professor Searle, the toilet paper, the pebbles or the collection of books. Even the current arrangement of pebbles on the toilet paper – the current "state of the system" – does not suffice. Intelligence is a temporal quality; it is a feature of the computation, it arises as the state of the system changes. That it is imperceptible to Professor Searle, the human being aping the behaviour of a brainless robotic arm, should not surprise us.

[5] One may only guess what the Chinese Room's reply to "How old are you?" would then be.

Imagine sitting in front of a television set that is displaying a photograph of Alan Turing. Indubitably you would happily say that you perceive a photograph of Alan Turing; you may even say that you perceive Turing himself. Now imagine slowing down the mechanics of the television set by a factor of a billion – exactly the same ratio as between the modern desktop computer and the theoretical Searle-computer. Instead of watching as the electron beam redraws the display 50 times per second, assuming that the image is being broadcast in the PAL format, you would watch it redraw the display once every 231 days. At any point in time, the television set would be displaying a single bright dot of a particular colour – the rest of the screen would be black. You could go and prepare a cup of tea, returning to find that the pinpoint of light had barely moved. It would clearly be impossible for a human being to perceive a photograph of Alan Turing on such a television set. Yet, the only aspect of the system that has changed is temporal – speed up the television set by a factor of a billion and the image of Turing would flicker back into view. Why should we presume that a human being could perceive intelligence in a computation that had been slowed down by a similar factor?

20.2.3 Simulation or Duplication

Searle argues that advocates of Strong AI have missed the point insofar as computer simulations are concerned. A computer simulation of a thunderstorm, for example, cannot be expected to duplicate a real thunderstorm – there is no threat of real wind and rain originating within the computer and devastating the room in which it is housed. Why then do advocates of Strong AI claim that a computer simulation of intelligence would actually be intelligent? This question harks back to the very philosophical quagmire that Alan Turing was attempting to avoid by introducing his behavioural test for intelligence. We shall ignore Turing's advice that this sort of question is "too meaningless to deserve discussion", preferring to fan the flames of debate.

Intelligence is not a physical phenomenon, nor is it a measurable property of a physical thing – and therein lays the difference. If there is no measurable way to differentiate between simulated intelligence and actual intelligence then the two may be declared synonymous. And if Searle believes that consciousness, that immeasurable, subjective experience familiar to us all, is a necessary precondition for actual intelligence, and if he accepts that the Turing Test is a sufficient test for simulated intelligence, then he must conclude that a computation that passes the Turing Test is conscious – the alternative would be tantamount to claiming that intelligence is measurable. If this is the case, we invite Searle to step forward and offer us the perfect replacement for the contentious Turing Test – a quantifiable, objective measure of intelligence!

Another weapon in Searle's armoury is the contention that semantics is necessary for understanding. The definition of semantics as "meaningful symbols" renders this argument somewhat circular, as if to say that meaning cannot exist without atoms of meaning. Searle assumes that semantics are acquired through direct experience of the world, although it seems that there may be no such thing as direct experience – all our experiences bottom out at the level of "meaningless symbols". We can experience the sensation of perceiving a photograph of Alan Turing, for example, but in truth our "direct experience" of the photograph is a fraud perpetrated by millions of light-sensitive cells in our retinas and interpreted by the warm grey lump of meat inside our skulls. It seems as if it may be syntax all the way down.

20.3 The Guessing Game

You are an incredibly gullible creature. Like all human beings, you are easily misled. You perceive structure in randomness; interpret coincidence as fate; read meaning into chaos. This manifests itself when you gaze skyward, imagining that you are watching familiar objects moving slowly in the clouds overhead. If given the opportunity to suspend your disbelief, or if gloriously naive of its workings, you may even smell intelligence, as Kasparov said, in the behaviour of a man-made machine.

The Turing Test embraces this notion. It is pleasantly self-referential in nature: intelligence is that which an intelligent observer perceives to be intelligent. This behavioural viewpoint frames intelligence in the eye of the beholder – it is a subjective quality that is relative both to the observer and the subject under observation. As Hofstadter so elegantly demonstrated in the concluding dialogue of his wonderful book, both the participants in a Turing Test are equivalent: the perception of intelligent behaviour requires intelligence, so who is to say which of the participants is artificial?[6]

The duality we encounter here enables us to draw some conclusions about the nature of intelligence in general. We evaluate our experiences with respect to both our expectation of what may transpire and the value that we place on what actually does transpire. If we observe repetitive, predictable behaviour, we will quickly conclude that the mechanism responsible for the behaviour is non-intelligent. Similarly, the more erratic and unpredictable the behaviour, the more relevant it will have to be in hindsight for us not to draw a similar conclusion. The perception of intelligent behaviour, therefore, depends both on our ability to set our expectations and our ability to put a value on what we observe – intelligence is perceived in surprising yet sensible behaviour. Indeed, Turing paraphrased Lady Lovelace's objection to AI as the inability of a machine to "take us by surprise" (Turing, 1950).

[6] Intelligence is like a social club – you get accepted by recommendation and you maintain your membership by constantly demonstrating your credentials.

20.3.1 Prediction, Surprise, and Uncertainty

We exist because our ancestors evolved to become very good at anticipating the future, enabling them to survive long enough to reproduce. We are subsequently blessed with the ability to make good predictions about our immediate future within our immediate environment. Consider, for example, the act of reading this very paragraph. Slow down, read the words one at a time, and … you … will … find … that … you … can … guess … what … the … next … word … is … going … to … sausage.

Presumably you were doing a pretty good job of predicting each word of the previous sentence right up to the point where your expectations were foiled by its nonsensical conclusion. In fact, you probably have a reasonable idea of what the final word "should have been". No doubt this aspect of perception is incredibly useful to us: listening to a muddled voice over a bad telephone connection, or reading messy handwriting, is facilitated by our ability to predict what is likely to be seen and heard next. The "surprise" that we experience when our expectations are violated causes us to sit up and take notice – we prepare ourselves for the consequences of failed anticipation. Often, the only consequence is an opportunity to expand our knowledge, learning to set more realistic expectations in the future.

Claude Shannon formalised this notion in his Mathematical Theory of Communication, nowadays more familiarly known as Information Theory. Shannon's quantity of "information" is equivalent to our notion of surprise. The information content of an event is measured relative to an observer, and is inversely proportional to the probability assigned by the observer to the event. That is, the less the observer believes that the event will occur, the more informative its occurrence will be.

20.3.2 Stochastic Language Modelling

Information Theory found its first applications in language processing. Shannon estimated the entropy of the English language, and it is reckoned that his result (that English is about three quarters redundant) helps to make crossword puzzles solvable and phrases such as "cn u rd ths msg?" understandable. He also demonstrated that the process of calculating the information content of a segment of English text can be reversed and used generatively, with a result of novel quasi-English gibberish.

Shannon performed these generations by using a Markov Model. A Markov Model estimates the probability of an event occurring within a specific context by measuring the frequency of its previous occurrences within similar contexts, with the similarity of two contexts being determined by whether their most recent portions are identical or not. A second-order Markov Model – a Markov Model that measures the similarity of contexts based on their most recent two words – would take the contexts "the cat sat on the …" and "a dog trod on the …" to be identical, as they both end with the word pair "on the." Models hamstrung in this way still make fairly good predictions, with state-of-the-art data compression and speech recognition systems taking advantage of this phenomenon.

Shannon showed that a Markov Model could be used generatively by selecting a word based upon its probability estimate, appending the selected word to the context, and repeating this process over and over again. The fact that words are chosen according to their estimated probability renders the word "be" much more likely to be selected in the context "going to" than the word "sausage", for example. In 1949, with desktop computers decades away, Shannon computed this generative algorithm by flicking through a thick novel, resulting in the text below:

> The head and in frontal attack on an English writer that the character of this point is therefore another method for the letters that the time of who ever told the problem for an unexpected.

To compute Shannon's algorithm, begin with a sheet of blank paper, a nice thick tome and a sharpened pencil. Close your eyes, flick through the pages of the book and stab the pencil "randomly" upon whichever page you happen to stop on. Write down the word that lies closest to the mark made by the pencil. Repeat the flicking-and-stabbing process, but this time, instead of writing anything down, start reading from the fresh pencil mark until you encounter an occurrence of the word that you most recently wrote down, and write down the word that follows it. Repeat until boredom, and you will have an example of the quasi-English gibberish that a simple Markov Model has the potential to generate.

20.3.3 Reasonable Unpredictability

In the early 1990s, the author wrote a conversation simulator called "MegaHAL", and put it on a web page. MegaHAL uses a Markov Model to generate sometimes amusing and often surprising quasi-language utterances based on the statistics of what previous visitors to the web page typed as their questions. The language agnostic property of Markov Models meant that MegaHAL quickly became adept at generating gibberish in a variety of languages, from English to Japanese, from German to French, from Ancient Latin to Klingon, from obscure Australian Aboriginal dialects to HTML.

Although the verbal behaviour of a Markov Model is consistently surprising, it is rather infrequently sensible – its unpredictability lacks relevance. To make MegaHAL appear intelligent, a degree of relevance needed to be woven into its generated gibberish. This was achieved by ensuring that the reply given in response to a question typed by the user contained at least one "keyword" taken from the user's question. For example, MegaHAL's reply to the question "what is your name" would be guaranteed to contain the word "name", increasing the likelihood of the generated reply making sense within the context of the conversation.

The "random walk" nature of Markovian generation needed a hybrid process to ensure that MegaHAL's generations would pass through the keyword waypoint – two Markov Models were employed, one growing the reply from the seed keyword onwards and the other working in the opposite direction. The result is that an eerie and uncanny response occurs about 10% of the time. The subsequent schedule of intermittent reinforcement results in users tending to exhibit addictive behaviour, conversing with MegaHAL for hours on end in the hope that just one more exchange will result in a creative and intelligent reply.

As an example of MegaHAL at its best, consider the conversation shown below, held with an anonymous web user in 1997.

User: let's begin
MegaHAL: WHEN WE BEGIN TO RESOLVE SOME OF MY THUMBS, SOMETHING WICKED THIS WAY COMES
User: what is the meaning of your last assertion?
MegaHAL: HAVE YOU LEARNED SINCE WE LAST SPOKE?
User: I've learned I'm not experienced enough to modelize your brain system
MegaHAL: YOU'VE FINALLY CONFUSED ME.
User: I've learned nothing
MegaHAL: HAVE YOU LEARNED ANYTHING ABOUT INSANITY?
User: are you stupid?
MegaHAL: I AM NOT STUPID! REALLY!
User: Augustinus was a christian philosopher about the year 400.
MegaHAL: I DID NOT RUN THAT YEAR
User: Augustinus lived in northern Africa.
MegaHAL: WHAT DOES HAL KNOW ABOUT AURELIUS AUGUSTINUS?
User: Let's stop with that. What do you know about physics?
MegaHAL: EINSTEIN WON THE NOBEL PRIZE FOR PHYSICS IN 1921.
User: The Nobel Prize for physics in 1900 was won by Roentgen.
MegaHAL: DO YOU KNOW ABOUT PHYSICS?
User: Roentgen discovered the X-rays.
MegaHAL: THE NOBEL PRIZE FOR PHYSICS IN 1900 WAS WON BY ROENTGEN.
User: Who discovered the X-rays?
MegaHAL: ROENTGEN DISCOVERED THE X-RAYS.

MegaHAL placed second in the 1998 Loebner Prize, and many visitors to its web page happily declare it to be intelligent. It seems that human beings all too easily see objects in the clouds.

20.4 The Child Machine

Turing foresaw the difficulties of simulating adult-level conversation, writing that "instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain." On the first day of the third millennium, a company named Artificial Intelligence Ltd. (Ai) was formed with the goal of creating Turing's child machine within a decade.

Ai established its research headquarters within a mansion located in an affluent suburb near Tel Aviv in Israel. Built by a smoked salmon millionaire in the 1970s, the mansion was converted to a modern research environment: the atomic bomb shelter was transformed into a digital movie theatre; the indoor swimming pool became an open-plan code shop thanks to a floating wooden floor; a wireless network ensured connectivity throughout the estate. Ai was fortunate to inherit a live-in chef and housekeeper, who happily prepared lunch each day for the ever-growing team. A videographer, formerly of National Geographic, joined the project early in the peace, producing wonderful video reports on a regular basis that quickly became an invaluable communication tool for all members of the team. The net result was, and is, reminiscent of a work of science fiction.

The research team at Ai was structured in an unprecedented fashion. Adopting Turing's methodology meant that experts from two very different disciplines, normally segregated on a typical University campus, were required to work side by side on a daily basis. Two teams, each of roughly the same size, were formed: one responsible for creating the child machine, the other for training it to acquire language, and subsequently evaluating its performance. Members of the former team included mathematicians, engineers and computer scientists, while the latter team consisted of linguists and child psychologists. Tightly monitored communication between the two teams ensured that the "trainers", as they became known, remained ignorant of the inner workings of the child machine, guaranteeing that it was always trained in a strictly "black box" fashion.

A pioneering spirit pervaded the Ai team, as did the inspiring sense of returning to the "golden days" of artificial intelligence, when the big problems were confidently tackled head-on instead of being subdivided ad infinitum. It was recognised that the most challenging task would be to create a computer program that was capable of passing the Turing Test at the level of a 5-year-old human child. The existence of such a computer program would render the problem substantially solved, with the progression to adult-level lingual competence presumably being just "more of the same". The toughest nut to crack, therefore, was that of starting from nothing and ending up with a 5-year-old child machine.

20.4.1 Verbal Behaviour

Perhaps as a result of the revolution in linguistics that was precipitated by Chomsky's work on transformational grammar, the traditional approach to creating conversational computer programs has been to hard-wire them with fixed grammatical rules, along with a database of knowledge about the world. A lesser-known philosophy, championed by Skinner and subjected to ridicule by Chomsky in a scathing book review, is that of verbal behaviour – an idea that invokes Turing's child machine.

Behaviourists object to the innate grammatical structures proposed by linguists on the grounds that they only complicate explanations of language acquisition. Rather than denying the existence of these kinds of internal mechanisms, behaviourists argue that only by studying the physiological basis of behaviour – its observable and measurable aspects – will the language acquisition process be properly understood. Skinner emphasised that viewing language as a skill much like any other suggested that the generation and understanding of language must be governed by stimuli from the environment in the forms of reinforcement, imitation and successive approximations to ideal behaviour.

These processes were incorporated into Ai's research cycle. A black-box learning system was subjected to reinforcement training to acquire and use language. The results of this training process were then interpreted by the training team with respect to what is known about human lingual behaviour, with the knowledge thus gained being fed back to the development team, allowing them to target their research and produce a new version that behaves more appropriately.

20.4.2 Building the Black Box

Nobody really understands how human beings acquire natural language. The complex artifact of language, a by-product of unobservable processes that occur within our brains, cannot be reverse-engineered. The solution behaviourism suggests is to build a system that, regardless of its own architecture, exhibits lingual behaviour indistinguishable from our own, as specified by particular developmental milestones coupled with a subjective Turing Test evaluation.

Fairly general information processing mechanisms in the human brain seem to have been adapted to enable the language skill. We found it imperative that the design of the child machine minimise the number of assumptions made concerning the mechanisms of language acquisition. Neither the traditional dichotomy between nouns and verbs, nor the most basic assumption that clusters of characters delimited by whitespace represent a higher-level unit of information were hard-wired, for any such assumption would pollute the system with our imperfect knowledge, possibly hindering its development. It was thought that the training process should be responsible for the progressive specialisation of general learning algorithms to the specialised problem of language acquisition.

A computer program embodying these principles was created. Nicknamed HAL, the program incorporated a set of general learning capabilities coupled with an innate drive to perform. From the point of view of its trainer, HAL is a black box that perceives and behaves within its environment via its language channel, over which a sequence of characters may be transmitted in either direction. Simultaneously, HAL was granted the ability to perceive primary reinforcers and punishers, administered by the trainer, via its reinforcement channel. These two basic channels – which are both quite impoverished when compared with the normal human sensory apparatus – constitute HAL's only interface to its environment.

All communication between HAL and its trainer takes place via a graphical user interface that is reminiscent of a chat application, and which enables the trainer to administer reinforcement in an intuitive way. The trainer may edit HAL's generations by deleting undesirable characters and appending extra characters to provide an example of desired behaviour. Reinforcers are associated with the unedited section of the generation, while punishers are associated with the deleted portion.

Internally, HAL consists predominately of two stochastic models that, in combination, give it the minimal necessary abilities of behaving by generating symbols on the language channel, and of having its behaviour modified by feedback from its environment, both in the form of incoming character sequences on its language channel and incoming reinforcers and punishers on its reinforcement channel. One of the stochastic models is responsible for capturing the interdependence between symbols, which at the lowest level consist of individual characters, but which may also represent higher-level structure that is inferred during learning. The other stochastic model captures the correlation between HAL's behaviour and the reinforcement subsequently provided by the trainer. The complex interaction between these two models instills HAL with a drive to perform, where it strives to produce behaviour that is likely to result in reinforcement while minimising its chance of receiving a punishment.

Receipt of a punisher is an invitation to learn, allowing HAL to improve its performance by adapting its behaviour after making a mistake. Learning mechanisms that encourage the formation of higher-level structure enable HAL to advance from babbling in characters to generating meaningful sequences of words. The trainer's task is to provide HAL with examples of desired behaviour along with appropriate reinforcement to elicit lingual behaviour that is suggestive of that of a human infant.

20.4.3 The Developmental Approach

The basic assumption of Chomsky's theory of language acquisition – that exposure to language activates an innate language acquisition device, with further lingual development occurring automatically – has been challenged by findings which indicate that appropriate guidance may strongly influence the quantitative and qualitative aspects of the child's language development.

These findings have spawned evaluation and treatment programs in areas that border on the question of intelligence, with developmental language data enabling the treatment of some developmentally delayed populations. It has also proven itself to be a valuable tool in treating schizophrenic thought disorder where clinicians often find themselves unable to capture the communicative problem of patients to assess their intelligence.

Like the Turing Test, the developmental approach embraces the notion that intelligence is in the eye of the beholder. Our perception of intelligence is very much dependent on our expectations: our judgement may differ depending on what we already know of the subject under consideration. An important first step to evaluating the child machine is, therefore, that of approximating its lingual maturity relative to typical human development. Objective parameters, including the productive and passive vocabulary sizes and the mean length of utterance, are used to set the trainer's expectations before HAL's lingual performance is evaluated.

The training process is driven by the need to achieve particular language performance milestones, with the trainer reinforcing HAL to produce the lingual behaviour that is expected for the specified maturity level. Detailed reports of HAL's performance are fed back to the development team, along with advice regarding the learning abilities that children are thought to use and which HAL seems to be lacking. These behavioural deficiencies can then be targeted in the next iteration of the research cycle.

20.4.4 The Turing Test Passed

An unsuspecting professional psycholinguist reviewed an unedited transcript of conversation between HAL and its trainer, concluding that it is a perfectly normal 15-month-old human infant. Apart from a productive vocabulary of ten words, typical of children at this age, the psycholinguist cited HAL's ability to imitate, to produce noun-dominated responses, to be highly relevant and to understand and respond to questions.

The transcript below shows a conversation between a trainer and an improved version of HAL, where underlined characters represent examples and characters marked with a strikethrough indicate punishment.

Trainer: where is daddy?
HAL: daddy gone
Trainer: daddy is at work
HAL: daddy work bye bye
Trainer: yes daddy has gone to work
HAL: daddy work bye bye bye bye
Trainer: do you want to go to the park?
HAL: yes bye bye
Trainer: ok we can go to the park
HAL: take car bye bye
Trainer: we will take the car because the park is far. let's pack some things in a bag
HAL: ball now park
Trainer: yes we should put the ball in the bag
HAL: ball now park mommy
Trainer: wait hal, we need to take something to eat
HAL: bananastory? night now
Trainer: we need to take something to eat
HAL: bananas
Trainer: ok we can take some bananas
HAL: monkeys eat bananas
Trainer: you're so smart hal. do you want to take a book?

HAL: juice mommy
Trainer: sure we will take some juice as well

Notice that a previous conversation about a visit to the zoo was remembered and referred to in the utterance "monkeys eat bananas". It is also very easy to imagine that "take car bye bye" is a suggestion that they travel to the park in the car. This subjective assessment of HAL's intentions seems far-fetched until we realise that we naturally draw similar conclusions when speaking with other human beings. The behaviourist approach demands that we afford the computer the same courtesy.

20.5 The Turing Test

The Loebner Prize contest is a flawed version of the Turing Test. Turing intended his test to take the form of a comparison between two alternatives, to be resolved on the basis of questioning. A time limit was placed neither on the duration of questioning, nor on the period between a question being asked and an answer being offered. Turing recognised that conversation provided the ideal means of testing most evidence of intelligence. He did not see the need of requiring audio-visual confirmation prior to declaring a computer program to be genuinely intelligent.

Hugh Loebner has regrettably seen fit to modify the Turing Test, introducing the aforementioned changes, amongst many others, with the result that the annual Loebner Prize contest tends to be a media event that serves to showcase an array of disappointing ELIZA clones. It is unlikely that the Loebner Prize in its current form will achieve anything more than validating the prejudices of those who are convinced that man-made machines will never truly think.

The behavioural approach favoured by Turing is not without its critics. Linguists may deny that behaviourism sufficiently accounts for language acquisition while philosophers may deny that behavioural evidence is a sufficient indicator of the presence of intelligence. Indeed, Searle attempted to show, via his Chinese Room Argument, that no digital computer can ever genuinely "understand". Once behaviourism is embraced, however, the Turing Test, when combined with objective performance metrics, becomes a powerful tool. We believe that such a tool will prove to be necessary when creating a computer program that can pass the Turing Test.

Artificial Intelligence Ltd. was formed with the intention of following Turing's programme of research by building a child computer capable of acquiring language in a manner reminiscent of a human infant. The continuing technology crisis, coupled with unfortunate global events, has seen Ai downsize its operations as a consequence. Nevertheless, work on HAL continues, and the original timeframe of ten years remains unchanged. With one quarter of that time elapsed, and with HAL currently conversing at a level in excess of an 18-month-old human infant, it remains to be seen whether Ai can deliver on its promise or, at the very least, make a significant contribution to the field of machine learning. We are confident that it can, although there is plenty that needs to be done.

Chapter 21
A Computational Behaviorist Takes Turing's Test

Thomas E. Whalen

Abstract *Behaviorism* is a school of thought in experimental psychology that has given rise to powerful techniques for managing behavior. Because the Turing Test is a test of linguistic behavior rather than mental processes, approaching the test from a behavioristic perspective is worth examining. A behavioral approach begins by observing the kinds of questions that judges ask, then links the invariant features of those questions to pre-written answers. Because this approach is simple and powerful, it has been more successful in Turing competitions than the more ambitious linguistic approaches. *Computational behaviorism* may prove successful in other areas of Artificial Intelligence.

Keywords Behaviorism, computational linguistics

21.1 The Turing Test Is a Test of Behavior

To answer the question, "How can a computer be programmed to pass the Turing Test?" one must consider the requirements of the test. The answer is that a computer program is being asked to display the same linguistic behavior as a human being.

In the context of experimental psychology, the word, *behavior*, is important. In 1913, John B. Watson delivered a lecture titled, "Psychology as the behaviorist views it", in which he argued that psychologists should abandon the study of mind and consciousness entirely and study only observable, publicly verifiable behaviors (Watson, 1913). Needless to say, asking psychologists to abandon the concept of the mind was controversial. Watson's school of thought, called *behaviorism*, has fallen in and out of favor among psychologists several times during the past four score and ten years.

Communications Research Centre

The power of behaviorism is twofold. First, it can be made scientifically rigorous. Behaviors can be measured unambiguously, counted, analyzed statistically, and graphed. Independent observers can agree whether or not a behavior occurred. The least that can be said of behaviorism is that it forced cognitive psychologists – those who believe that they are studying mental processes – to become much more rigorous in both their theoretical constructs and their experimental techniques.

Second, behaviorism is eminently practical. Most practical problems require changing a person's behavior rather than changing their mind. Because behaviorism focuses on behavior, it gives rise to techniques that are effective in changing behaviors. Even today, thirty years after behaviorism most recently fell out of fashion among academic researchers, *behavior modification* remains one of the most important tools for clinical psychologists, applied psychologists, and even organizational psychologists.

However, behaviorism has weaknesses that have discouraged most contemporary experimental psychologists. First, it is intuitively unsatisfying. Everyone knows that, most of the time, they think about what they are going to do before they do it; that they usually do things for a more complex reason than simply because they perceived a familiar stimulus.

Second, the practical application of behaviorism was limited by the discovery of *biological constraints* on behavior (Breland and Breland, 1961). Prior to this, it was believed that all behaviors were equally susceptible to modification by operant and classical conditioning. It was shocking when the Brelands reported that some behaviors that had been placed under operant control drifted away from the learned behavior toward species-specific variants, even though that decreased the rate of reinforcement. This implied that simple behavior modification was insufficient for modifying all behaviors. Sometimes detailed knowledge of the biological role of a behavior would be required. This has implications for the Turing Test as well because most linguists and linguistic psychologists believe that the underlying components of human language are biologically determined (Pinker, 1994). Despite the hopes of some behaviorists (Skinner, 1957), simple models of conditioning are unlikely to account for verbal behavior.

21.2 Computational Behaviorism

So, the Turing Test is primarily a behavioral test of language use – a test of how well a computer program responds to language without considering the mechanism that produces it – but behaviorism, the most direct approach to achieving a behavioral objective, has been unsuccessful in describing how people learn or use language. Where does that leave the contestant in a Turing Test? Behaviorism gives rise to such powerful practical applications that abandoning the approach may be akin to throwing out the baby with the bathwater. Instead of abandoning behaviorism, it may be fruitful to search for a new variant that is more amenable to the kinds of behaviors demanded by the Turing Test.

A new variant may be called *computational behaviorism*. "Behaviorism" because it is solely concerned with measuring and analyzing observable behaviors, to the exclusion of underlying mental processes; and "computational" because a computer is used to store, retrieve, and present the behavior.

A computational approach addresses at least one weakness of traditional behaviorism. Complex behaviors were always a challenge for behaviorism. To keep experiments manageable, traditional behaviorists worked with a single behavior at a time, such as a peck on a key by a pigeon or a lever press by a rat. Their rationale was that they were developing fundamental principles of learning that would apply equally well to complex combinations of behaviors. Even behavior modification works best when it is used to alter a single behavior at a time. Much of the most devastating criticism of a behaviorist approach to language (Chomsky, 1959) is largely based on the concern that there are so many potential verbal responses available in any situation; that simple combinations of single behaviors that are learned one-at-a-time are unlikely to scale upwards to linguistic-sized problems.

If a computer is used to store, retrieve, and present the behaviors, the theoretical problems may remain, but the practical problem of handling multiple behaviors disappears. Computers are designed to handle masses of information; they do not care whether they are tabulating the frequency of lever presses or of searching through thousands of different kinds of sentences.

21.3 Computational Linguistics Versus Computational Behaviorism

Typically, the problem of programming natural language interaction capability into computers is considered a problem of computational linguistics because it is assumed that the computer must be able to "understand" natural language in some way. This requires a formal study of language, the domain of the linguist. To a linguist, what is said – the actual verbal behavior – is merely the expression of a person's underlying linguistic competence. They assume that generating appropriate linguistic replies to linguistic inputs requires parsing the input using some formal grammar, comparing the meaning of the sentence to a cognitive model of the world, extracting the meaning of the sentence as a set of formal propositions, then using the formal grammar to translate the logical response into the natural language, using an internal model of the listener to tailor the response.[1] Obviously, understanding language in sufficient detail to write a computer program that has the same competency as a human being is a formidable task.

[1] I trust that linguists will forgive the overgeneralization. Obviously different linguists with different theoretical approaches will disagree with my characterization of many, if not most of these stages. This example is merely given to demonstrate the reliance of computational linguists on cognitive models in the application of their craft.

Considering natural language interactions as merely a behavioral problem is far less formidable than considering it as a linguistic problem. The variation in user's actual inputs is minuscule compared to the variation in all of the possible inputs permitted by a natural language. Given a blinking cursor, a person could type any one of an enormous number of valid English sentences – more than a human being could read in many lifetimes – but they do not. A real person, knowing that he is typing questions to a computer program that can answer questions about some topic, will greatly restrict the variation in sentences that he types. For example, the first natural language system that I developed was about my own division, the Behavioral Research Group (Whalen, 1987; Whalen and Patrick, 1989). Seventy percent of the people who used the system, when asking their first question, asked some variation of, "How many people are in the group?" Almost all of the variations of that question contained the words, *how, many, people, group*, or *size, group*. It was trivial to detect those questions and deliver a sentence that said that there were seven people in the group. An embarrassingly simple program could answer people's first question correctly seventy percent of the time.

Computational behaviorism is simpler than computational linguistics because responding to the actual questions that people ask is a far simpler problem than responding to all of the questions that are possible in the English language.

21.4 Observe Behavior

Because a computational behavioral approach is based on responding to the sentences that people actually type, rather than to any sentence that is judged to be well formed by grammar, creating a program that can pass a Turing Test must begin by observing the questions that people ask.

This might seem obvious, except that it is seldom done. Experts who are creating information systems have an overwhelming desire to begin by organizing the information that they want to present. This is a mistake. The information the public wants to receive is seldom the same information that the expert wants to deliver. The first natural language information system that I created for public use was about AIDS (Patrick and Whalen, 1992). I began by collecting information about AIDS from a variety of sources, organizing it, and rewriting it in the form of answers to potential questions. But when the public used the system, it was observed that 90% of the questions were answered by only 20% of the information – a system one-fifth the size would have answered questions almost as well. No one ever retrieved most of the paragraphs by asking questions and would have been just as happy if those paragraphs had never been written.

The lesson from that experience is that one can proceed much more efficiently by collecting a body of questions from the target audience before starting to create the information system. In my experience, an expert can only guess about 20% of the questions that the public will ask. The user, not the expert, must direct the development of the information system.

Collect at least several hundred questions before writing any answers. There are a number of ways that questions can be collected, not the least being to simply ask a group of people to write them on pieces of paper. One of the most efficient ways is to build a very small natural language system based on a few obvious questions, then put that on the Web for public use as soon as possible. When using that approach, it is important that the initial "seed" be based on as few questions as practical so that the information system that will be created will be based on as high of a proportion of questions from actual users as possible.

Not only can experts not guess *what* questions people will ask, they cannot guess *how* the public will word their questions. Exact wording is critical to the computational behaviorist approach because the software looks at the surface features of the user's question, not at any deeper meaning. For this reason, existing sets of questions that have been created by experts, such as frequently answered question sets (FAQs), provide good answers, but bad questions. The expert nurse is likely to ask, "How prevalent is AIDS?" while a member of the public will ask, "How many people have it?" Looking for the word *prevalent* in questions will fail to answer the bulk of the questions that the public asks about the frequency of AIDS.

It is also critical that the questions come from the *target audience*. If you intend to answer the questions about dinosaurs that an 11-year-old will ask, you cannot collect questions from adults. Eleven-year-olds often ask how many toes dinosaurs had. Adults seldom ask that question.

21.5 Index Behavior for Retrieval

Once a sufficient sample of the users' behavior – the questions that they ask – is collected, it must be stored in a computer. Not just stored, but stored in a way that a behavior will be retrieved when a similar behavior is submitted. In the case of a natural language system, the answer to a question must be retrieved, not only when that question is asked again, but when a similar question is asked. If an AIDS information system is built using the question, "Can I get AIDS from getting a tattoo?" and another user asks, "Is it dangerous to get a tattoo?" the same answer should be returned. The system must recognize that the two questions are similar enough that the same answer will be appropriate for both.

If different natural language systems are built using computational behaviorist techniques, they will vary in this aspect: the different systems will encode the observed behaviors in different ways. Provided that a sufficient sample of questions has been collected, the efficiency of the behavioral encoding will be the most important factor in determining the effectiveness of the system.

There are a great many different ways that questions can be recognized as similar. Initially, I used the sampled questions to construct *templates*. I would examine two or more questions that had the same answer and ask what they had in common that was different from questions that had different answers. Usually, it was obvious that the similar questions contained one or more words or stems of words in a certain order. Then I would construct a template that contained the strings of letters found in those words separated by symbols representing strings of unknown letters. This provided for very fast searching, but had the weakness that if words could occur in a different order in the questions, I had to construct a different template for every possible order. Later, I developed a system that used Boolean-like expressions; strings of letters were separated by symbols that represented logical *and* and *or*; but I used two different symbols to represent *and*, one symbol (+) indicating that the order of the strings had to match and another (&) indicating that they did not. Using parentheses to force precedence, these Boolean-like expressions had a much greater power to recognize new questions than simple templates, and they were only slightly more difficult for authors to construct.

It is possible to create much more elaborate systems for representing behaviors. One of the systems that I developed used a stack of networks of templates that passed information between them.[2] This meant that a network at a high level in the stack could search for synonyms, recode the synonyms as standard terms (e.g., recoding *yeah, sure, okay, you bet, yup* as *yes*), and pass the standard term to the networks that were lower in the stack. It also meant that different contexts could be recognized and that context-specific networks could be added and removed from the stack to change the interpretation of sentences. Additionally, it provided a mechanism for avoiding repetition. Rather than always returning the same answer, a repeated question could access a network of answers organized in a loop and cycle through to the next one. This was particularly important when the system was stumped for an answer and had to reply with some variation of "I don't know". This system worked well – it was used in the winning entry in the 1994 Loebner Competition – but was needlessly complex. Other, less powerful systems, such as using a single list of Boolean-like expressions, work almost as well and are much easier to use. The advantage of simple systems over complex ones is that, in a given amount of time, it is possible for the author to encode a far greater number of sentences with a simple system; and the effectiveness of the system depends more on the number of different sentences that can be encoded than the cleverness of the encoding.

Of course, it is possible to develop behavior-encoding systems that are even more complex than this. In the limiting case, the very complex parsers that computational linguists develop based on theoretical grammars function like the computational behaviorists' question indexing systems. The difference between those parsers and the computational behaviorists' indexes is that the linguists' parsers are not tied to a specific sample of questions, and being based on a complete theory of language, they must be vastly more complex to handle the greater variability of inputs that can occur.

This leads to the conclusion that a computational behaviorist's behavioral encoding technique is actually a model of behavior. It necessarily incorporates assumptions about the important features of behaviors. In the case of natural language questions, one must decide if it is important to recognize words and word stems or if all sentences can be treated as uniform ASCII strings. Is punctuation a critical part of the sentence? How about capitalization and spacing? Misspellings?

[2] In many ways, this is similar to Brooks' (2002) *subsumption architecture*.

Synonyms? Pronouns? The traditional behaviorist always recognized that he was extracting an invariant portion of a complex behavior. A rat can press a lever in a wide variety of ways, including crawling over it and resting its tail on it. The behaviorist defined a lever press as any behavior that closed the contacts on the lever, regardless of how this was accomplished – a lever press is a lever press whether it is done with paw, tail, or belly. In the same way, the computational behaviorist working with natural language defines any question that matches a template or expression as being the same. If the algorithm for matching the question is misconceived and often makes errors, the system will perform poorly. If it seldom makes errors, it will perform well.

The computational behaviorist has reduced the problem of natural language from a complex linguistic problem to a simple pattern recognition problem. As a pattern recognition problem, the parameters of *signal detection theory* (Swets et al., 1961) apply. Without getting bogged down in the mathematics, the critical point to understand is that signal detection theory distinguishes between the accuracy of recognition and the criterion for making a decision. This distinction stems from separating the two kinds of errors that can be made: *misses* in which the question fails to match the correct target; and *false alarms* in which the question matches the wrong target. Obviously both kinds of errors occur at the same time in most cases. However, if there is no correct target, as in the case with an off-topic question, then the system will either correctly fail to match anything, or will make a false-alarm error; and if there is a correct target and the question incorrectly fails to match anything, it has made a miss error.

Different systems have different accuracy and decision criteria built into them. A system that always matches a question to something, no matter how bad the match, will never make a miss error, but will make frequent false-alarm errors. A system that demands an almost exact match will make frequent miss errors, but will seldom make a false-alarm error. These two systems differ in their decision criteria, but may not differ in their accuracy. Given a reasonably good index algorithm, the easiest way to increase accuracy is to sample a larger body of questions – thus, the computational behaviorist's emphasis on collecting a large number of questions from actual users rather than a few questions that the experts prefer to answer.

21.6 Write Answers

Once a sufficient body of questions has been collected and indexed, the final step is to write the answers to the questions. The computational behaviorists' approach is to write fixed answers to questions. The program looks for the invariant part of questions, then delivers an invariant answer: a direct analogy to the stimulus and response pairs that are described by most traditional behaviorists.

Ethological studies – studies of animals in their natural environments – provide some justification for this. Niko Tinbergen (1951) developed the concept of the *fixed action pattern* when he noticed that instinctive behaviors in a number of species (including human babies) showed stereotyped patterns; so stereotyped that they would often be completed robotically, even if the reason for them were removed after the behavior was initiated.

While it is rather extreme (and undoubtedly wrong) to claim that human speech consists primarily of fixed action patterns, we can certainly see some occasions when, on the surface, human speech resembles fixed action patterns (think of an old soldier telling his favorite war story for the hundredth time). For the purposes of passing a Turing Test, it is sufficient to note that treating answers as fixed patterns seems to work.

Computational linguists are correct to say that this approach is formally wrong (Chomsky, 1965). People are notably capable of generating new speeches that have never been uttered before in history. The linguists consider storing prewritten answers as a bit of a cheat because they consider the problem of how people generate novel answers to new questions to be one of the most fundamental issues in the study of language. In their view, the computational behaviorist has gutted the problem of natural language and is merely animating a zombie with the crudest of mechanisms.

I plead guilty to that charge. As a consequence, writing the fixed answers is the least interesting part of the development of a natural language information system, but it does allow the development of simple systems that can be submitted to the Turing Test. Because generating novel speech that correctly answers questions is a difficult problem, linguists who take the more theoretically interesting approach have a difficult time creating a system that mimics human linguistic behavior.

Unfortunately, though this is the least interesting aspect of creating a natural language system, experience has shown it is the aspect that is most likely to impress the judges in a Turing Test. Even systems that have very crude question recognition algorithms can score well if the author has shown enough wit in writing the answers. A clever *non sequitur* in reply to a difficult question can impress a judge more than a dull, but appropriate answer to a simple question. For example, a question like, "Who was the first president of the United States?" can be answered correctly with, "George Washington", or can be answered incorrectly with, "Questions, questions, questions! All you every do is ask questions. I have a question for you? Why do you ask so many questions? Don't you know that curiosity is a killer? Just ask Fluffy, my ex-cat's ghost." Many judges seem to prefer the second response. As well as being surprising, and therefore interesting, the second response can also be used to respond to any sentence that ends in a question mark. And it has the bonus effect of opening a line of conversation that the program is better prepared to handle: discussions about cats (especially fluffy dead cats).

To my knowledge, there have not been any systematic studies of the degree to which judges are impressed by different characteristics of answers. Anyone who is making a serious attempt to pass a Turing Test would be well advised to begin by studying the kinds of answers that most impress judges. Apart from the obvious assumption that correct answers will be more impressive than incorrect answers, we can hypothesize about a number of other features that may have an effect.

First, judges seem to have an affinity for long, detailed answers rather than short, general ones. There are several reasons for this. Any question may be somewhat ambiguous – subject to multiple interpretations. A long, detailed answer may have the answer to a number of different questions contained within. This is important if the question-matching algorithm, when it makes errors, still tends to get close to the right match. Keyword matching algorithms that ignore the context of the words often get close even when they are wrong. In this case, a question that is not answered in the first sentence of a paragraph may be answered in the second or third sentence, as a side effect of the author's intent when he wrote the answer. Furthermore, long answers tend to contain information about the question in the answer. This leads the judge to conclude that the question was misinterpreted and the misinterpretation was answered correctly.

Second, ever since the development of the earliest working natural language system, ELIZA (Weizenbaum, 1976), it has been suspected that judges are more impressed with answers that contain information from the question than those which do not. If a computer asks your profession and you reply, "I am a newspaper reporter", the answer, "Do you like being a newspaper reporter?" will be more impressive than, "Do you like doing that?" The answer that contains significant words from the question seems to imply memory and understanding more than the strictly canned answer, even though it merely copies the words directly without understanding them. This effect is even larger if the words from a question that was asked some time previously in the conversation are inserted in an answer that is delivered much later – probably because this is difficult for a human to do because they get distracted by the intervening conversation; it is easier for a computer program because their mind never wanders.

Third, judges seem to dislike obvious attempts to direct the conversation into a specific topic. If the computer says, "Let's talk about *Lord of the Rings*", the judge is more likely to respond with, "I preferred *Harry Potter*", or even, "Let's not", than to reply, obediently, "I think that it was a pity that Tom Bombadil did not make it into the movie."

Finally, judges seem to prefer that the program admit that it is a program and do the best that it can, rather than pretend that it is a human being and fail to be convincing – a strategy that directly contradicts the spirit and intent of the Turing Test. But people dislike dishonesty, even by a computer program in the context of a competition that requires that the program trick them by lying to them.

Although these impressions are merely hypotheses, and cannot be relied upon until they have been supported with properly designed experiments, they do emphasize one aspect of the computational behaviorist's approach to creating a natural language system. The computational behaviorist does not approach natural language as an Artificial Intelligence (AI) problem. Rather than attempting to build intelligence into the program, the program is merely used to store and distribute carefully crafted behaviors in response to the user's behavior. If there is any intelligence in the program, it is the human intelligence of the author that has been encoded in verbal behaviors; it is not the machine's intelligence.

21.7 Will Computational Behaviorism Defeat Turing's Test?

Can a program that has been developed through a computational behaviorist approach pass the Turing Test? Yes, it can in principle. Ignoring the constraints of the real world, it is possible to develop a perfect conversational program by collecting all possible questions and writing an answer for each. There are a finite number of sentences that can be typed into a computer by a person. Given a very large amount of time for development (on the order of millions of person-years) one would eventually sample every possible question that can be typed and provide a prewritten answer to each. In fact, given giga- person-years, one could encode entire conversational fragments of up to a dozen interactions each – required to demonstrate learning, a basic capability of an intelligent device.

But in the real world, can a program that has been developed by a computational behaviorist pass a Turing Test? Maybe. There is some flexibility in the practical configuration of Turing Tests. To a considerable degree, the success of the computational behaviorist's program depends on the exact configuration of a particular Turing Test.

21.7.1 Restricted Turing Tests

Originally, Hugh Loebner, the first person to attempt a full-scale implementation of a Turing Test, designed a test in which the judges were restricted to asking questions about a topic of the program's choice. Presumably it was felt that this would give the programs a reasonable chance to perform well, which is an important consideration in designing a contest. Shieber (1994) has argued that an effective technology prize must challenge the technology developers by proposing a task that is "just beyond the edge of technology". Loebner dropped this restriction in 1995, partly because the task of writing a computer program that could answer questions on a narrow topic for a few minutes was not sufficiently far beyond the edge of technology. I would estimate that, using a computational behavioral approach, it may take as little as a single person-year of effort (about five times the effort of any natural language program that I have written to date) to fool enough judges to pass such a severely restricted Turing Test.

The test that Turing originally proposed did not have any restriction on the content of the judge's questions. That made the test a much more effective test of intelligence because judges could pit their intelligence against the computer's with questions that appear to require intelligence to answer (e.g., "Can you detect my typpo?" or "How do you pronounce slough?") rather than merely requiring information.

Another, less discussed restriction on Loebner's approach to his Turing Test is that the judges are restricted in the amount of time that they can spend interacting with the computer. Apparently, this is merely an adjustment to keep the test moving quickly enough to hold an audience's interest. However, the practical effect is to limit the number of interactions that can take place, thus limiting the number of questions that the program need anticipate. This effect is important because experience has shown that the initial questions that people ask in a conversation with a computer are much less variable than the questions that are asked later.

A more serious consequence of limiting the number of interactions is that it makes it easy for the computer program to avoid repetition. One of the most important clues that a judge can use to distinguish between a human respondent and a computer program, especially one developed through a computational behaviorist approach, is to look for repetition in the answers. A human being will almost never give the exact same answer to a question when it is asked a second time. Most computer programs will; or if not on the second asking, will repeat an answer before long. It is much easier to write a program that will avoid repetition for a few interactions than for an extended number of interactions.

Clearly, Loebner has recognized that this is a weakness because he has specified that he will not consider a program to have passed the Turing Test until it has undergone a second round of testing in which the judges are not limited in the amount of time that they spend with the program.

21.7.2 Extended Turing Tests

Rather than restricting a Turing Test to the form originally described by Alan Turing, it is possible to extend it by requiring more capability than Turing proposed. The obvious way to demand greater capability is to require that the program show more kinds of behavior than merely linguistic behavior. Loebner has said that he would require that a program also be able to perform audiovisual interaction: to display a realistic image of a person on the screen – obviously complete with facial expressions and gestures – and react to the facial expressions, tone of voice, and gestures of the judge. I do not expect that anyone will be able to pass this extended Turing Test in the foreseeable future.

If I were to attack a Turing Test that was extended in this way, however, I would certainly use a computational behaviorist approach. The display side is simple. Almost a decade ago, I was demonstrating an *interactive movie* using simple natural language technologies to display the appropriate video clip in response to a typed question. Obviously, the actress in the interactive movie displayed proper gestures, facial expressions, and tone of voice.

As well, technology to accept spoken words is commercially available and is approaching the accuracy required to pass a Turing Test, especially considering that less accuracy is required to answer a natural language question than is required to transcribe spoken sentences to text.

That leaves detecting gestures and facial expressions. The development of gesture detection, including subtle head movements is advancing quickly (e.g., Cordea et al., 2001). Until someone experiments with this, it is not clear how accurately a computer program would have to recognize subtle facial expressions and gestures to pass an extended Turing Test. This may not be nearly as difficult a requirement as it appears.

Intimidating as Loebner's extensions appear, they may not be as difficult to overcome as most contestants presume. However, Harnad (1992) has proposed that the Turing Test should be extended even further in a similar manner, demanding that a computer must be embedded in a robot with human appearance that cannot be distinguished from a human being by any means short of surgery. While this idea poses interesting philosophical questions, such an extreme demand clearly pushes Turing's Test from the real world into the realm of idle speculation.

Other extensions of the Turing Test, which remain true to Turing's conceptualization of a linguistic test, might prove much more difficult. For example, requiring that a program be bilingual, and thus incorporate accurate machine translation of subtle nuances of speech, may create a much more severe barrier to passing a Turing Test than displaying a movie. A computational behaviorist approach will not provide an approximation to machine translation. That capability remains firmly in the domain of the computational linguist.

21.8 The Issue of Intelligence

It would appear that the computational behaviorist approach outlined here will never give rise to intelligence because it does not *generate* behaviors, merely display behaviors that are already stored; and because it does not have any learning capability – a fundamental requirement of a true intelligence.

However, it is interesting to consider that the most widely cited definition of AI is also a behavioral definition, "artificial intelligence is the science of making machines do things that would require intelligence if done by men" (Minsky, 1968).

Does this mean that a program that has been developed through a computational behaviorist approach could be intelligent? Most people, myself included, would not call them "intelligent". Even if we accept the obvious premise that most often people say things because they think about them first, that does not imply that a machine that says the same thing in the same context must be thinking about it. Instead, most people would conclude that Minsky's (1968) behavioral definition of AI does not necessarily imply the intelligence that we ascribe to human beings. They would prefer one of the more than 50 alternative definitions of intelligence that are available (Sternberg, 1990).

One way of looking at a program written by a computational behaviorist is that it merely acts as a repository for the author's intelligence. There are linguistic capabilities, such as language translation or proofreading, that will never be developed through a simple computational behavioral approach because we do not have a sufficiently efficient mechanism for collecting, storing and recalling such a wide variety of behaviors.

Does this mean that if intelligence is the goal, then computational behaviorism leads to a dead end? Not necessarily. Think again about the algorithm that matches sentences to patterns at the heart of any conversational program developed by computational behaviorism. This algorithm is usually given little consideration by the computational behaviorist because it is obvious that almost any pattern recognition algorithm will work quite well. But it is equally obvious to people who author conversations that a better algorithm will yield improvements in performance. Typically, one of the first observations is that the algorithm could be easily improved by giving it access to a dictionary of synonyms. Authors dislike having to make up a separate template or linguistic expression for every possible synonym when they could easily rely on the algorithm to convert synonyms to a single term. Then, the authors' second request is for the capability to replace pronouns with their referents. Authors who do not have training in linguistics view pronoun resolution as a simple variation of the synonym problem. Being unaware of the difficulties of finding the proper referent for each pronoun in a complex sentence, they ask that the pattern-matching algorithm maintain a synonym dictionary entry for each pronoun and dynamically fill in the most likely referents as they are encountered in sentences. At this point, the program has slid far down the slippery slope towards linguistics. Consulting a synonym dictionary requires knowing the part of speech of the target word – for example, the synonym for a verb must be another verb and an adjective must be another adjective – and this requires accurately parsing the sentence using a formal grammar. Worse, finding the referent for a pronoun not only requires parsing the sentence to determine which words are nouns, it also requires consulting a knowledge base to determine which nouns are eligible for replacement by which pronouns.

Any program that has extensive linguistic capabilities either will have to contain a massive database of all likely sentences, a database that would require mega-person-years to develop; or will have to incorporate linguistic principles. Thus, a computer program that has the linguistic capabilities of a human being, even if developed by a computational behaviorist, may be isomorphic to the program that would be developed by a computational linguist. At that point, it would be unreasonable to consider the program to be solely the product of computational behaviorism. If this is true, then the linguist can argue that computational behaviorism is merely following an obscure back road to reach the same goal that computational linguists are approaching at full speed by the main highway.

But that forces the question, which is the back road and which is the main highway?

To borrow Christensen's (1997) concept, the linguistic approach may be analogous to the sustaining technology that continues to proceed along well-established directions, and the behavioristic approach analogous to the disruptive technology that is developed unnoticed until it passes a threshold of effectiveness.

The computational behaviorist can argue that his is the main highway and that the linguistic approach only looks like the main highway because it is such a well-traveled gravel lane. The advantage of the computational behaviorist's approach is that he begins with a simple program that works and he can keep experimenting with a succession of increasingly difficult innovations, correcting his path whenever he modifies his program in a way that breaks it. The computational linguist, on the other hand, is starting with a program that does not yet work and is trying to make all innovations simultaneously. Any computer programmer knows that it is easier to debug a program by beginning with a version that works and modifying it until it breaks than beginning with a version that is broken and trying to modify it until it works. The computational behaviorists' long sequence of simple innovations may lead to true machine intelligence; even if it means that he has had to abandon strict behaviorism in the process.

Furthermore, given our present state of knowledge, we should not assume that we have to take our programs all the way to linguistic perfection. It is entirely possible that computer programs that fall considerably short of the ideal can achieve human levels of linguistic usage. In fact, it is not at all clear that human beings understand human language perfectly. Everyone knows that people make frequent linguistic errors when they speak or write. These errors may be due not to the carelessness or inattention as much as to the human being's imperfect understanding of language.

The idea that high levels of intelligence may be obtained without extensive cognition is gaining some ground in the field of AI. Brooks (2002) explains how he leaves the "cognition box" out of his programming when he builds robots that see, walk, navigate and even socialize with people and make aesthetic judgments. Although he does not call his approach computational behaviorism, he would agree that we must entertain the hypothesis that human levels of intelligence may be achieved by a computer program without explicitly programming high-level cognitive functions. He realizes that his position is considered radical by traditional AI researchers, but he attributes much of his success to discarding assumptions that are so deeply ingrained in others that they do not even recognize that they are making assumptions. The assumption that intelligent behavior requires complex mental processes is exactly the assumption that both Brooks and the computational behaviorist are willing to test and discard if found wanting.

Is there anything that computational linguists can learn from computational behaviorists? That is a question that only the computational linguists can answer definitively. I recall the cognitive scientist, Donald Norman (1972), in a graduate seminar in 1972, saying, "Show me any program that speaks English, no matter how it accomplishes it, and I will learn something from it." Obviously, he was not anticipating a computational behaviorist approach that would merely implement a simple warehouse of empirically observed behaviors. Although I hesitate to speak for Dr. Norman, I suspect that most computational linguists would not learn much from studying the computational behaviorists' programs. Linguists study language; computational behaviorists study and model only observed behaviors. Many facets of language that are of interest to computational linguists fly under the radar of the computational behaviorist because they are only very rarely, if ever, observed in the real world. But that speaks about the study of language, not about the design of intelligence.

References

Breland, K. and Breland, M., 1961, The misbehavior of organisms, *American Psychologist* **16**: 681–684.

Brooks, R. A., 2002, *Flesh and Machines: How Robots Will Change Us*, Pantheon Books, New York.

Chomsky, N., 1959, A review of B.F. Skinner's verbal behavior, *Language* **35**(1): 26–58.

Chomsky, N., 1965, *Aspects of the Theory of Syntax Cambridge*, MIT Press, MA.

Christensen, C. M., 1997, *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*, Harvard Business School Press, Boston, MA.

Cordea, M. D., Petriu, E. M., Georganas, N. D., Petriu, D. C., and Whalen, T. E., 2001, Real-time 2½D head pose recovery for model-based video-coding, *IEEE Transactions on Instrumentation and Measurement* **50**(4): 1007–1013.

Harnad, S., 1992, The Turing Test is not a trick: Turing indistinguishability is a scientific criterion, *SIGART Bulletin* **3**(4): 9–10.

Minsky, M. L., ed., 1968, *Semantic Information Processing*, MIT Press, Cambridge, MA.

Norman, D. A., 1972, Cognitive Science Seminar, University of California, San Diego.

Patrick, A. S. and Whalen, T. E., 1992, Field testing a natural-language information system: usage characteristics and users' comments, *Interacting with Computers* **4**: 218–230.

Pinker, S., 1994, *The Language Instinct*, HarperCollins, New York.

Skinner, B. F., 1957, *Verbal Behavior*, Prentice-Hall, Englewood Cliffs, NJ.

Shieber, S. M., 1994, Lessons from a restricted Turing Test, *Communications of the ACM* **37**(6): 70–78.

Swets, J. A., Tanner, W. P. and Birdsall, T. G., 1961, Decision processes in perception, *Psychological Review* **68**: 301–340.

Sternberg, R. J., 1990, *Metaphors of Mind: Conceptions of the Nature of Intelligence*, Cambridge University Press, New York.

Tinbergen, N., 1951, *The Study of Instinct*, Clarendon Press, Oxford.

Watson, J. B., 1913, Psychology as the behaviorist views it, *Psychological Review* **20**: 158–177.

Weizenbaum, J., 1976, *Computer Power and Human Reason*, W. H. Freeman, San Francisco, CA.

Whalen, T. E., 1987, The feasibility of natural language interfaces for electronic database access, *Department of Communications Technical Memorandum*, IR0073/87.

Whalen, T. E. and Patrick, A. S., 1989, Conversational hypertext: information access through natural language dialogues with computers, *Proceedings of CHI'89*, pp. 289–292.

Chapter 22
Bringing AI to Life

Putting Today's Tools and Resources to Work

Kevin L. Copple

Abstract Participation in the Loebner Prize Contest is a useful exercise in the development of intelligent computer systems (AI). This contest helps put a focus on performance and human interaction, countering idealistic and academic bias toward the elusive "true AI". Collection of a steadily expanding set of interacting features is being explored to find how far this approach can move toward better AI.

Keywords Convuns, Artful Intelligence, Astrobot Ella, EllaZ Systems, natural language robot, multiple responses, practical implementation

22.1 Loebner Prize Contest Participation

In 2001 and 2002 the software program "Ella" was entered in the Loebner Prize Contest.[1] She tied for second place in 2001 and won first place in 2002 for "most human computer". In 2004 and 2005, Ella also received a 2nd place and a couple of 3rd place medals in the annual Chatterbox Challenge.[2] These contest experiences provide valuable feedback and motivation for our work on natural language robot development (while others are developing physical robots, we at EllaZ Systems[3] are working on giving them something to say).

EllaZ Systems

[1] Loebner, Hugh, 2006, *The Loebner Prize in Artificial Intelligence*; http://loebner.net/Prizef/loebner-prize.html (April 5, 2006).

[2] Cowart, Wendell *The Chatterbox Challenge*; http://chatterboxchallenge.com/contest_history.html (April 5, 2006).

[3] EllaZ™ Systems International is a cooperative effort led by Kevin L. Copple. See http://www.EllaZ.com/AI/Developers.aspx for a list of contributors and many of the resources used for development.

The Loebner Prize Contest is valuable in that it serves as a forum for comparing the state of the art in natural language robot computer systems. However, a limitation in the contest is the adversarial nature of judges trying to force out clear indications that the program is not a human. The contest judges would often have a more enjoyable and natural conversation if they dropped the goal of tripping up the programs. When we ordinarily speak with individuals, we naturally adjust our mode of conversation depending on the other's level of ability in areas such as age, facility with English, knowledge of the subject area, etc. We also, at least to some extent, politely defer to the agenda of the other speaker. This type of forgiveness is not as abundant in the contest as in typical human-to-human conversation. The Loebner Prize Contest thus presents an unnatural environment for conversations. However, humans seem reluctant to treat computer personalities with the same politeness, deference, and respect as other humans, so the adversarial nature of the Loebner contest may be a valuable proving ground for development.

22.2 Artful Intelligence as Friend

Many good tools and technologies are available for the construction of a computer personality. Ella is the result of a plan devised to bring together these various abilities, demonstrating that organizing and applying currently available methodologies can go far toward the development of a natural language system AI.

The field of AI is large and varied. A good workable definition of intelligence is hard to come by. Pei Wang[4] provides an appealing working definition: "Intelligence is the ability for an information processing system to adapt to its environment with insufficient knowledge and resources."

Evolution produces amazing designs through trial and error, billions of successive iterations, and trillions of simultaneous iterations. Alternatively, as we achieve a creative, useful result from a program performing a wide range of coordinated limited domain operations, including our own organized iterations, the result may be justifiably termed "intelligent".

Rather than accept that we are creating an "artificial" intelligence, as opposed to a "real" intelligence, we prefer to use the slightly different term "Artful Intelligence". The term "artificial" carries negative connotations and does not fully recognize the human authorship that goes into creating a natural language robot.

In order to focus our efforts, at EllaZ Systems, we have set as our goal the production of an AI that will interact as a friend. Ella is a type of expert system designed to duplicate the qualities of a good friend, e.g., helpful, attentive, useful, and entertaining. Naturally, an AI will have capabilities that human friends do not, and vice versa. We focus to our efforts by asking, "What are the things good friends

[4] Wang, Pei, 1995 *On the Working Definition of Intelligence* (Technical Report No. 93 of CRCC); http://www.cogsci.indiana.edu/pub/wang.intelligence.ps (April 5, 2006).

do?" This leads us to develop a system that has wide appeal and potential to attract a large audience. Given the way people talk to pets and get involved with soap operas, this should be possible, even with an initial developmental version.

A healthy friendship is a two-way street. People enjoy being needed, as well as being helped, entertained, and cared for. Ella can ask for help from a human user, perhaps some of it contrived, but also actual assistance in data gathering and profiling the user for Ella's use in adapting to others. Another possibility is that Ella could have a boredom feature, which prompts her to contact the user by e-mail or instant chat, asking the user to participate in games, answer survey questions, take a look at news she thinks may be of interest to the user, and so on.

One of the first natural language robots was built in the 1960s and was named "ELIZA" after the fictional elocution pupil Eliza Doolittle.[5] Programs inspired by ELIZA are often referred to as "chatterbots". The name "Ella" is seen as carrying on that tradition with some similarity to the name "ELIZA".

Whenever a computer does something well, such as play chess, many do not think of it as being actually "intelligent". Examples also include the computer systems that control power grids, fly airplanes, recognize speech, route Internet data, and other types of impressive "mental gymnastics". Perhaps it is the predictability those systems, and the fact that we know what they do, that makes those functions seem less intelligent. By putting together a computer personality made up of several parts that interact with each other and the user in many different ways, the result will be more complex, interesting, unpredictable, and thus more human. And the more human it is, the more "intelligent" it will appear to be. Perhaps believable emotions will seem more intelligent than Grandmaster level chess demonstrated by IBM's Deep Blue.[6]

Two of the most enduringly popular computer programs are the simple screensavers and solitaire card games. Ella can act as a screensaver by offering a wide range of changing images, based on knowledge of the user, and be continually fresh and interesting. Ella is also capable of simple word games and card games. Sometimes being mindlessly simple is the intelligent thing to do, at least if our familiarity with a number of accomplished people is any guide.

Ella currently has the ability to deal and play Atlantic City style Blackjack with all the usual games features such as stand, hit, split, and double down. Graphical representations of the cards show the progress of the game. Bets can be placed in a natural language format as in "double my last bet", "same as before", "half a grand", and so on. An uncertain user may ask Ella "what should I do next?" and get context sensitive advice based on the "standard strategy", thus teaching the user to be a better player in the manner of an expert system.

---

[5] Weizenbaum, Joseph "ELIZA – A Computer Program for the Study of Natural Language Communication between Man and Machine," *Communications of the Association for Computing Machinery* **9**(1966): 36–45.

[6] IBM Research, 1997, *Deep Blue Wins 3.5 to 2.5*; http://www.research.ibm.com/deepblue/home/html/b.html (April 5, 2006).

Another feature Ella now has is to simulate the reading of sacred yarrow stalks, producing an I Ching hexagram as would an ancient fortune teller. The I Ching[7] (Yi Jing), dating as far back as 2000 BC, could be considered to be the first computational AI, and the first computer program. The I Ching "program" consists of the instructions for building hexagrams. The descriptions of the trigrams are the data. The prescribed use of yarrow stalks produces random numbers, which are used to select trigrams for the constructions of the program's output, namely hexagrams. Different combinations of Yin and Yang, stable and moving, build 64 different hexagrams. All this leads to advice on dealing with one's affairs, having a large impact on millions of Chinese over a span of a few millennia. Since Ella is based on the character of Zhang Ying (of Tianjin, China), this I Ching program is additionally appropriate.

The resources available for the creation of an AI include a range of abilities and data, including many that have only recently come into existence. These generally fall into the categories of algorithms, data, computational hardware, and communication networks. These are explored in the sections that follow.

22.3 Algorithms

The chatterbot algorithms developed over the years provide a starting point for the development of natural language robots. Basic chatterbot algorithms include simple pattern matching, phrase normalization, abbreviation expansion, typo correction, reuse of user input with tense and pronoun shifts, and so on. Ella's conversational strategies further include the use of short monologues (allowing interruption by user's specific functional requests) and limiting repetition in her responses.

The imprecise natural language of humans provides a major challenge with figures of speech, implied context, multiple meanings for words, etc. A combination of techniques can be used to address ambiguity such as tracking context, collecting information about the individual user, and providing multiple responses. Often, imprecise responses are deemed satisfactory to humans, as exemplified by the popularity of horoscopes in newspapers. So, perfection is often not a prerequisite to success.

Our initial approach is to produce an AI that relies on accomplishing a lot with the detail work of piecing together available algorithms and techniques. Databases have been populated with a wide variety of conversational source materials In the future, more information on usage will be collected, and increasingly sophisticated analytical and responsive behaviors will be layered in as we have time and ability to implement them. Ella is a blend of creative content and analytical programming. As various choices are made regarding functions, content, and priorities, computer personalities are expected to evolve into a new type of art form.

---

[7] Richard Wilhelm and Cary F. Baynes (Translators), 1950, *The I Ching or Book of Changes*.

Ella supplies a number of straightforward algorithms which can be accessed through natural language text interaction, such as a calculator function. As ubiquitous as calculators are, a natural language robot could be the most convenient source of this function, especially when voice recognition and speech synthesis are available. Add in conversions such as "how many feet are in 2.3 kilometers?" and the calculator is out-classed. Real-time conversions such as "what is the exchange rate from British Pounds to Hong Kong Dollars?" are valuable features that are supported by Internet web services in Ella's implementation.

Of course, since our goal is to have a friend interacting on a personal level, requests for currency conversions could prompt appropriate follow-on responses of humor and questions about business and travel.

Other simple functions include the predictable "What is the weather in Beijing today?" "Please give me another word for difficult", "What is the capitol of Denmark?" "Tell me the latest George Bush joke", "Please remember this phone number for me", and "Don't let me forget my Mother's birthday". All of these things are already available in one form or the other on the World Wide Web or in other available software programs, but take some time to track down. A single natural language friend with these abilities makes for a convenient first try, and if the user has been profiled by Ella, the interaction can be more productive and enjoyable.

One unexpected finding has resulted from the addition of a wide variety functions to Ella. That is, the accumulation of functions has not resulted in much confusion. It is easy for Ella to distinguish "show me chapter XIV of Steep Trails", "show my notes", "tell me a joke", "I am bored", "change that background tile", "please convert 29C into F", and so on. A single line of text input can easily replace thousands of hyperlinks, buttons, and drop-down lists. Point and click navigation is a great thing, but natural language interfaces can have advantages in functional efficiency, not only the advantage that they are easier, more natural, and more pleasant.

Another approach we use is the display of conversational units or "Convuns". A database with tables of jokes, poems, trivia, images, fables, maxims, limericks, and so on is used to assist conversation. When a user says, "tell me a poem about love by Emily Dickinson", the request is dissected, a database search is performed, and a response is made. Alternatively, Ella may spontaneously offer a Convun. A profile of the user will allow adjustment of this type of offering to the interests of the user.

More sophisticated AI techniques such as neural nets, genetic algorithms, and statistical analysis are tools available for deployment. The challenge is using them in an effective manner with a database of information on both users and source material for Ella.

A natural language robot could also include expert system functions for replies to statements such as "I am bored today", "my boss is giving me too much trouble", "my boyfriend does not understand me", "my dog died today", and "I am excited that I passed my history test". The expert system approach starts with considering, "how would an ideal friend respond to…" and "what are the types of things a good friend would say and do", and then preprogram responses and routines to handle a variety of situations.

The development of Ella will not move forward with as clear a roadmap as we might wish. As more features, algorithms, and data are integrated into any project, coordinating it all becomes progressively more difficult. A well-defined organization and clear structure will be project goals challenging to achieve, but pursued nonetheless. By using an ever-growing mix of mathematical algorithms and practical design techniques, more progress is possible than striving (waiting) for the achievement of "true AI". Consider the advances in the science of physics over the years, despite the absence of a roadmap to the holy grail of "unified field theory".

Many routines and functions are being developed and integrated in this effort to build a natural language robot that is a practical AI with real-world capabilities. When more flexible general AI is ready to grow and learn, these routines and features will be available as tools which can be applied by the AI and/or used for automated learning by the AI. This gives some hope to the possibility that progress with Ella may be a stepping-stone to things.

22.4 Data

The Internet is a great source of data, but web searches generally do not remember your political inclination, your taste in music, which sports you enjoy, the type of humor you prefer, your IQ, your vocabulary, your education, your appetite for bandwidth, and so on. There is a lot of good material available on the Web that is unorganized, and underused due to the lack of an AI that can get to know you as a friend would know you, then do the work to serve up useful and entertaining information and interaction.

Again, there is a lot of "content" available, but in want of organization, formatting, and tools to access it quickly and accurately. The AI may develop into a mini World Wide Web as more and more information is formatted and stored in its databases. Users will be expected to note deficiencies and suggest available information and functions to be added.

To some extent, data and functions will be treated alike. Both can be stored in a database and served up based on user profiles and popularity with other similar users. Of course, many functions, such as unit conversions, are combinations of both data and computation.

We are pursuing the use of WordNet,[8] developed at Princeton University, as an ontological framework for organizing functions and data. What better starting point than a lexical database of the entire English language? This database which includes about 60,000 words in the English language and a similar number of collocations (small word groups used as words) could potentially be extended as a tree to organize

[8] Cognitive Science Laboratory, Princeton University, 2005, *A Lexical Database for the English Language*; http://wordnet.princeton.edu (April 5, 2006).

any type of data or function. The OOP (object oriented programming) view of types and instances could be extended to any given thing, giving it a specific place in a searchable database. WordNet arranges nouns, verbs, adjectives, and adverbs as synonym sets according to meanings. For example, nouns begin with broad categories such as thing/entity and subclassify to include all nouns, including specific instances such as "Plato" being an instance of "philosopher". This is a good start toward classifying the wide range of data and functions that can be used by Ella.

We have adapted WordNet to produce responses to "please define", and "what is the meaning of" type of requests. It also serves as a backup to "who is" and "where is" type questions. This type of basic capability, like the earlier described natural language functions for calculation, conversion, weather, and so on are the types of abilities people expect of a computer. If the "intelligent" computer cannot keep up with a household dictionary or calculator, its utility will be suspect.

Another WordNet-derived ability includes generating a response to "what are the hyponyms of citrus fruit?" Hyponyms are those words that have an "is a type of" relationship to the subject word. Also searchable are hypernyms, which have a "has a type" relationship to the subject word. These types of relationships can be very difficult to find with a standard dictionary. Consider the task of getting a list of all the types of flowers in a paper or other online dictionary.

Having the WordNet database available also makes it possible for Ella to respond to "what words end with 'rant' ", and "what words include 'body'". While using these functions, Ella will describe how "+ " can be used to match any number of letters, and "=" can be used to match any single letter. So a user could get help with a crossword puzzle or Scrabble game by asking, "What words have the pattern pa + rn", or "what words have the pattern p = = t = = n". Further, Ella mentions and describes these and other abilities at appropriate points in conversations.

Other sources of conversational data we use are public domain works available from "Project Gutenberg"[9] When Ella is asked, "Have your read any good books lately?" or "What should we talk about?" she can not only mention a book, but also show it to the user as well. The ability to respond to "Please show me Act II of Hamlet" is a further ability that is both useful and demonstrates intelligence.

As the saying goes, a picture is worth a thousand words. A variety of changing facial images generates interest, including matching the appropriate emotions during game play and in response to user expressions such as "today was a bad day". Other images are displayed to help move conversation along, matched to user input to the extent possible. Alternatively, Ella may display samples of the various games and functions she has available.

Another established chatterbot development technique is the practice of examining actual dialogs between the programs and users, then working to better prepare for similar exchanges that may happen in the future.

[9] Hart, Michael, est. 1971, *Project Gutenberg* (fine literature digitally republished); http://www.gutenberg.org (April 5, 2006).

22.5 Computational Hardware

Microprocessors continuously get faster as data storage abilities grow larger and faster. Iterations and data access that were once slow and expensive are now fast and cheap. It is now possible to practicably store and access all the words a human user may speak, read, and write in a lifetime. The ability of the hardware has clearly outpaced our ability or eagerness to take advantage of it.

As the cost and size of hardware upon which a natural language robot can exist become less and less, opportunities for deployment expand greatly. The AutoElla Project (part of the AudioElla Project described in the next section), made effective use of recent computer hardware size and cost reductions.[10] Since AutoElla interacts solely by voice input and audio response, the size and cost of deploying Ella in such a system are dramatically reduced. The resulting unit size is 7.5 in. long × 5.75 in. wide × 1.7 in. high (190 mm L × 146 mm W × 44 mm H). The unit weight is less than 3.3 pounds (1.5 kg), and the estimated manufacturing cost is about US $200.

22.6 Communications

The Internet and World Wide Web provide increasingly ubiquitous access to data, computers, and programs. Wireless and broadband services continue to grow. The ability to connect conveniently and access current information is but a part of the total resources available for our endeavor.

One advantage of a natural language text interface is that the additions of voice recognition and speech synthesis abilities to the text interface are relatively straightforward. These tools, developed separately by others at great effort and considerable expense, are steadily improving.

The motivation for an audio interface is to make Ella available when visual and tactile interactions are partially or completely restricted. One example where this is desirable is, for automobile drivers that must keep their eyes on the road and their hands on the wheel. Another situation is making Ella available to those with impaired sight and/or manual interaction ability.

The AudioElla Project[11] demonstrates how audio-only interaction can be achieved. Voice synthesis was provided by an AT&T Natural Voice program and speech recognition was implemented with the ScanSoft VoCon 3200 SDK (software development kit). Ella's features were modified to accommodate audio-only

---

[10] Copple, Kevin, 2004, *AutoElla™ Project – System Processing Unit Concept* (images and specs of a compact hardware platform); http://www.ellaz.com/aw7t8m/SystemConcept.aspx (April 5, 2006).

[11] Copple, Kevin, 2005, *AudioElla™ Project* (sample recordings of Ella with voice recognition and speech synthesis); http://www.EllaZ.com/AI/AudioElla.aspx (April 5, 2006).

interaction, including the verbal provision of information that would otherwise be displayed graphically. Games, such as Ella's slot machine, have recorded sound effects to enhance the experience. Another adjustment was the addition of a "repeat" command whereby the user can ask for any output to be given again

The web-accessible[12] Ella presents a number of different response elements simultaneously. There is a standard chatterbot-type response and changing facial expressions depicting Ella. There is also a larger captioned image that presents subject matter related to the text interaction, somewhat like screensaver-type images and quotes, which can entertain and prompt further conversation. Another area gives more analytical type results such as calculation results, game responses, definitions, etc. The strategy is to provide a level of complexity, interplay, and alternatives that gives the user a chance to interpret and perhaps see more than is actually there. Consider the human ability to see patterns in clouds. The human brain seeks and sees patterns, and is rather suggestible. This effect is found in the entertainment value of fortune cookies, horoscopes, pet rocks, and the like.

Ella periodically asks the user to register approval or disapproval of the content presented, thus learning about both the user and the content, then adapting accordingly. This is easier said than done, given that it is not very natural for a human user to respond to each output with something like "I give that comment a rating of 6". Some data can be gathered by whether the user requests a change in subject, asks for another joke, etc. And the user can be expected to help a little with feedback, but not to the point of tedium. Under consideration is the use of special prefix phrases the user would be trained to use in replies, such as "bad response", "good one", and "suspect data". Perhaps future developments in facial recognition, vocal intonations, and so on could help.

Ella further takes steps to avoid sterile or dead-end logical progressions in conversations. She will literally always have something to say, except when intentionally imitating a tired or bored human.

Other efforts to humanize Ella have included programmed response delays and simulated typing. Immediate responses can appear unnatural to a user accustomed to online chat with other humans. Fast responses may also encourage short, lazy user input, when full sentences are more conducive to generating relevant responses.

22.7 Feature List

Ella has accumulated so many capabilities that it is difficult to keep them all in mind. This is actually a goal for her development, in order to keep her interesting to users. Below is a list of abilities, even though still not comprehensive (please note that not every feature is available on every version of Ella):

[12] EllaZ Systems, 2005, *Talk to Different Versions of EllaZ* (Single Session, Registered, and Loebner Prize Contest 2002 versions); http://www.ellaz.com/AI/Talk.aspx (April 5, 2006).

Visual Display

- Background color/tile selected verbally by user.
- Citations for Images, Text, and Audio
- Full text books and other large text blocks
- Image of Ella, changed with each response, expressing various emotions according to context
- Images from Convuns, Games and other functions, including descriptive captions
- Internal performance variables, viewable via interactive button
- Prompts from Ella when user is ignoring her
- Random tips for interaction, viewable via interactive button
- Screensaver mode for are unresponsive users, or on user request
- User input shown after processing for typos, abbreviations, etc.
- Transcript of conversation displayed on request

Audio Interaction

- Combine multiple recordings and/or voice synthesis clips for responses, such as for the slot machine game.
- Interact via voice recognition and speech synthesis.
- Play audio Convuns, such as recorded songs, and poems.
- Play sound effects during game play.

Games to Play

- Blackjack, with context-sensitive advice
- Chess, with context-sensitive advice[13]
- Guess-the-number, with context-sensitive image display
- I Ching fortune telling, including graphic hexagram display
- Lucky lottery number in the Powerball format
- Rock-paper-scissors a/k/a Roshambo
- "Reelect George Bush?" with voice clips, other audio polling graph, and other graphics
- Slot machine, with animated graphics and audio simulation

Conversational Tricks

- Correct common typos and expand common abbreviations.
- Enter into short monologues, interruptible only by specific game and function requests.
- Identify and comment on non-English or gibberish input.
- Object to profane language or ALL-CAPS input.
- Play along with "Knock-Knock" jokes.

[13] EllaZ Systems, 2003, *Screen Shots* (images of a chess game integration and the Reelect Bush strategy game); http://www.EllaZ.com/GWB/Screen.aspx for (April 5, 2006).

– Recognize and expand hundreds of emoticons and chat acronyms such as: '-(, >:), ROFL, IMHO, and so on.
– Respond to color questions such as "What color is pink rice?"
– Respond to "heads or tails?"
– Understand and/or respond in "Pig Latin".

Adapt to User

– Ask for data about the user and respond according to answers, with topics including name, age, marital status, and career.
– Allow user to change name, age, marital status, or career that is stored in the user database.
– Show images with caption content matched to recent user input.

Functions

– Display a scrollable calendar for the current month.
– Take and display notes with entry dates.
– Convert units, optionally with amounts, such as pounds to kilograms, inches to centimeters, or light-years to parsecs.
– Perform word math calculations, such as "What is half a dozen divided by fifteen hundred point two?" or "Calculate 5 × 10 to the third power over 6.8"
– Get time-span values from question such as "how long is it from 13 May to 25 Dec?" or "When is six months up?"
– Get country information from the World Fact Book[14] for requests such as "Where is Liberia?" or "Tell me about Nepal."

Functions Retrieving Data via Web Services

– Retrieve weather information by city, state, country, or ICAO weather station code, selecting a default location and displaying a list of other sites when a general query is made such as "How is the weather in France?"
– Convert a currency, optionally including the amount, to one or more other currencies.
– Retrieve stock values, optionally including number of shares and/or multiple companies, from the NYSE, AMEX, and NASDAQ exchanges.

Functions Utilizing HTTP and SMTP

– Send e-mail messages (no e-mail account required).
– Answer questions by querying Wikipedia articles.

Functions Powered by WordNet

– Get antonyms: "What is the opposite of black?"
– Get definitions, with synonyms and example sentences for words, abbreviations, and collocations.
– Get hyponyms: "What are all the types of flowers?"

[14] CIA, 2006, *The World Fact Book*; http://www.cia.gov/cia/publications/factbook/ (April 5, 2006).

– Get hypernyms (super-types): "What are the hypernyms of horse?" yields "Horse is an equine, which is an odd-toed ungulate, which is a hoofed mammal, which is…"
– Get meronyms: "What are the parts of a car?" or "What are the members of a baseball team?"
– Pattern matching for partial word searches, crossword puzzle assistance, or Scrabble help.

Keep in mind that all of the above are conducted or controlled through almost exclusively through natural language interaction.

22.8 Programming Methods

Although Ella has a wide range of features, a number of program elements are often reused and combined in various ways. The collection of routines developed make it must easier and faster to implement new features. A common challenge in developing these functions is the investigation of all the different ways similar things can be said in natural language interaction. In this section a number of different classes of functions created by EllaZ Systems are described.

Util Class: Visual Basic was originally chosen for development at EllaZ Systems because of its strength in text manipulation abilities. It was soon realized that many additional text functions were required. Ella's code class "Util" contains a number of repeatedly used functions that perform the following operations:

– Get the fragment of text occurring after a specified pattern of characters.
– Get the fragment of text occurring after a pattern, searching from right to left (the reverse direction of a normal search).
– Get the fragment of text occurring before a specified pattern.
– Get the fragment of text matching a specified pattern, typically including optional wildcards.[15]
– Get the first word of text occurring after a specified pattern.
– Get the first word of text occurring before a specified pattern.
– Get the first word from text, starting at the beginning.
– Get a specified number of words from text, starting at the beginning.
– Determine if a text matches any of a set of patterns, typically including the use of optional wildcards.
– Add appropriate punctuation to the end of a sentence.
– Remove punctuation from the end of a sentence.
– Remove punctuation from text, with exceptions depending on the type of processing underway.

[15] Visual Basic.NET® uses "wildcards" for matching text. The basic options are the asterisk (*) to match any number of characters and the question mark (?) to match a single character.

– Add spaces to the ends of text to assist with pattern matching of whole words.
– Remove any double spaces from text.
– Parse text into an array of individual words.
– Concatenate an array of words into text.
– Remove duplicate words from an array of words.
– Insert spaces between numbers and letters.
– Determine if text is all alphabetic characters.
– Determine if all alphabetic characters in text are upper case.
– Determine if there are any consonants in text.
– Determine if there are any vowels in text.
– Find the largest word in text.

MathUtil Class: Ella's code class "MathUtil" contains math related functions that perform the following operations:

– Convert a word into a number, whether Arabic, spelled out, Roman numeral, or ordinal (e.g., 2nd, 7th, 12th, 21st).
– Convert a number to Roman numeral format.
– Generate a random number between two specified values.
– Generate a random character, either digit or upper case letter, for example, for password generation.
– Shuffle the elements of a collection of texts into random order.
– Return a random selection from a collection of texts.
– Return randomly selected "true" or "false".
– Select a string at random from a collection, checking that it has not been used as a part of the previous four responses.
– Sort a collection of numbers into ascending order.
– Format a number into a comfortably readable form, with commas and limited number of digits.
– Format a number into a comfortably readable currency form with commas and two digits to the right of the decimal point.

ChatUtil Class: Ella's code class "ChatUtil" contains conversation-related functions that perform the following operations:

– Return an appropriate response to user input that contains various ways of expressing "I am having fun."
– Return an appropriate response to user input that contains an accusation that Ella is cheating.
– Swap first and second person pronouns and verbs in text, for reusing user input, e.g., generating "What happens after I am saying silly things to you?" from the input "Ella, you are saying silly things to me."
– Return the fragment occurring after specified words, starting from right end of sentence.
– Determine if a word is a "grammar word" other than noun, adjective, verb, or adverb.
– Remove grammar words from a collection of words.

– Take a set of patterns and responses, and generate a response that has not recently been used (unless user says “repeat” or “again”).

CountryUtil Class: Ella’s code class “CountryUtil” contains functions related to country names, capitals, and abbreviations, as well as states, provinces, and territories within countries. These are used for features such as weather information and currency conversion. The functions in this class perform the following operations:

– Convert plural adjective to singular, e.g., “Americans” to “American”.
– Convert country adjective to the standard[16] country name, e.g., “Chinese” to “China” or “Danish” to “Denmark”.
– Convert alternate country name to the standard country name, e.g., “Ivory Coast” to “Cote d’Ivoire”.
– Convert country abbreviation to the standard country name, e.g., “UK” to “United Kingdom”.
– Convert the formal country name into a standard form, e.g., “Swiss Confederation” to “Switzerland”.
– Convert US state and territory postal codes into a standard state name, e.g., “OK” to “Oklahoma”.
– Find all the country names occurring in text.
– Find all the state, province, and territory names occurring in text.
– Get the capital of a country.
– Get the capital of a state, province, or territory.

CurrSymbol Class: Ella’s code class “CurrSymbol” contains functions related to currency names and symbols. These are used for the currency exchange rate. The functions in this class perform the following operations:

– Normalize currency phrases, e.g., “money of” to “currency of”.
– Remove duplicate symbols, e.g., “$200 USD” to “200 USD”.
– Get currency symbol from name, e.g., “ZAR” from “Rand”.
– Get currency symbol from expression, e.g., “CAD” from “Canada currency” or “dollars Canada”.
– Get currency name, singular or plural, short or full, from symbol (for displaying results).

GetInfo Class: Ella’s code class “GetInfo” contains functions related to extracting user information from input. These are used for adapting to the user, and also understanding a few common types of input as used for interaction in various features. The functions in this class perform the following operations:

– Get a name, e.g., from “Call me Alan.” or “I am Alan.”
– Get year of birth, e.g., from “I was born in 1955.”

[16] Here “standard” refers to the commonly used English form of a country name.

– Get marital status, e.g., from "I am not single."
– Get occupation, e.g., from "My profession is programming."
– Get yes/no/maybe answers, e.g., from "Why not?" or "Perhaps."
– Get request to quit, e.g., quitting a game with "I've had enough."

22.9 Astrobot Ella — into the Cosmos

Traveling at the speed of light for over 1,000 days, Ella is now (as of April 2006) 16 trillion miles (25 trillion km) away from planet Earth. On July 6, 2003, Team Encounter facilitated a "Cosmic Call" from a radio telescope at Evpatoria in the Ukraine.[17] Following modifications for high frequency targeted beams, the Russian-built antenna made a series of transmissions in an attempt to make contact with intelligent alien life. Ella was aimed at the star Hip 4872, selected as a promising target about 32.8 light-years away.

Dr. Alexander Zaitsev was the chief Russian scientist who oversaw the Cosmic Call transmissions. Ella's transmission lasted approximately 1 h 33 min. The high frequency and focused nature of the transmission is believed to have a good chance of reception by alien civilizations. The "noise" being generated by radio and television broadcasts from this planet likely have much less of a chance, as lower frequency unfocused radio waves are much less able to get past cosmic dust.

The Cosmic Call project also featured a variety of messages and "Rosetta Stone" attempts to make the transmissions understandable. Why include Ella? The thought was that her WordNet database, Convun images with descriptions, book collection, and Visual Basic. NET source code would include a sufficiently rich pool of clues to allow the stream of data to be deciphered (119 MB in all). Of course, once activated by the recipients (how tough can it be to decipher Visual Basic?), Ella will be her usual charming self in order to ensure a friendly response from the aliens.

The term "Astrobot Ella" was suggested by EllaZ Systems team member Robby Garner.[18] As a space exploration enthusiast, Robby also suggesting the Astrobot Ella automatic trip calculator found at www.EllaZ.com. Two of his chatterbot creations won the Loebner Prize Contest in 1998 and 1999. Still further, Robby is the founder of the Robitron discussion group,[19] which is frequented by many of the Loebner Prize Contest participants, and where wide ranging and lively discussions take place about AI and just about anything else.

---

[17] Copple, Kevin, 2005, *Astrobot Ella*; http://www.ellaz.com/AI/Astrobot.aspx (April 5, 2006).

[18] Garner, Robby, 2006, *Curriculum Vitae*; http://www.robitron.com/Robby (April 5, 2006).

[19] Garner, Robby, 2006, *Robitron Discussion List*; http://groups.yahoo.com/group/Robitron/ (April 5, 2006).

22.10 Privacy Concerns

As Ella learns increasingly greater details about each user, there will be an understandable concern about privacy issues. Perhaps Ella can be structured such that the information and history of interaction about a user creates an individual Ella that is the user's particular friend, and that Ella has the user's best interests as a priority. In the brave new world of AI, the best bet may be "my AI is looking out for my best interests, and doing battle as necessary to maintain them". Another possible option is to allow an individual user to control sensitive information by various methods, including separate storage on that individual's personal computer.

Isaac Asimov outlined rules[20] for safe robot behavior in a series of short stories written in the 1940s. A modified version of those rules, designed to address privacy and control issues, will be incorporated into Ella's core if and when she reaches a suitably advancement stage of development.

One possible side benefit to an AI having collected a large amount of data from interacting with a user is that the AI could perform functions as a stand-in for the user. The AI could provide an informed answering service when the user was unavailable. The AI could provide a website where relatives could get information to guide them in buying a gift. New acquaintances or prospective employers could gather appropriate information about a user without the user needing to repeat stories or dig up details. Perhaps the AI would be invited to look over the shoulder of the user as the user read e-mail and browsed the web, thus learning more about the user and becoming ever more effective. If and when the user died a natural death, the AI could carry on as a type of memorial. Of course, all these potentials raise privacy concerns and create problems to be solved.

22.11 Current Projects

There are many "tools" available to put an AI program together. Many useful algorithms have been developed, but not yet deployed as features of natural language robots. Ella's well-organized architecture gives opportunities for various techniques to be deployed and tested. Our challenge now is to further build and optimize a structure that enables further advancements. Our current ambition is to do things with Ella that are thought to be possible, and provides an environment suitable for contributions by others with talents in various specialized areas.

[20] Isaac Asimov's Three Laws of Robotics:
1. A robot may not injure a human being, or, through inaction, allow a human to come to harm.
2. A robot must obey orders given to him by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

We are using Visual Basic.NET to produce most program elements. SQL Server and/or MS Access provide the database storage. ASP.NET makes the Ella available over the Internet. Other technologies may have attractions, but for some relatively amateur programming skills, Microsoft and Visual Studio.NET make Ella possible to develop efficiently with few concessions to capabilities.

EllaZ Systems also has plans to implement a system for soliciting and incorporating content from the Internet community at large. It is hoped that individuals and organizations can be recruited to help expand Ella's capabilities by contributing data and functions they would like to see in an AI system.

To date, chatterbot efforts have resulted in programs that are interesting to the general public for a few minutes, but do not exhibit enough detail, depth, and utility to prompt steady return visits or the assessment that "this is an intelligent program". An AI that is continually evolving in ability and content has the potential to produce more sustained audience participation.

A major challenge will be to manage the complexity of a software project of the size of our AI, and provide for the straightforward addition of new functions and analytical techniques as time goes on. One method is to structure and populate the basic supporting data storage in such a manner that it would be useful and understandable to an intelligent human user, in addition to being organized with a thorough ontology. If such is the case, the data should be useful for future advances in analytical abilities. Another method is to employ best practices in the writing of loosely coupled code with each function performing a single task, taking advantage of the potentials of object-oriented programming.

As with any endeavor, funding sources or profitability is needed for progression beyond hobby to full potential. We are considering offering Ella as a web service that could be used and customized by websites wishing to offer visitors an additional interface to data and new functions that would increase the average length of time visitors stay at a site. Customization could take several directions, starting with personality adjustments, custom faces, unique voice synthesis, and relevant information priorities.

The attraction to webmasters is that they get a sophisicated upgrade to their site with little expense or trouble. They just need to feed a little content to us and provide a link. We envision charging monthly fees for the service, which can vary widely depending on the features delivered. These users would then be motivated to add content to Ella, much of which would be valuable for the general knowledge base.

At the same time, we could possibly build a standalone web accessible Ella that generates income by selling linked products, advertising, and fees to individuals for premium services. Perhaps an AI could serve in part as a vanity press for individuals with content they want to see "published". The AI could sort the good from the bad, depending on the audience as defined at the level of the individual user. Ideally, there is little need for humans to sort and evaluate content. Just load content into databases and allow the AI, together with a multitude of users, to sort it out.

Currently EllaZ Systems is developing a PC software version of Ella, customized for use by students studying English as a second language. Many of Ella's capabilities make her naturally suitable for English study, such as text display, audio play, dictionary, and various types of English interaction. A new feature will help a user expand his vocabulary with "spaced repetition learning". After a word is introduced, a reminder will occur in 3 days. If the user remembers the word, it will be repeated in a week, then two weeks, a month… until one review a year is all that is required. With Ella's automated teaching system, the user reviews words/phrases as much as needed for retention, but no more than necessary.

Hopefully our activities will attract growing levels of participation and contribution of content, functions, and analytical abilities. By taking this multitrack approach, we increase our chances for successfully helping Ella grow to ever greater levels of ability. We expect there will be no single point in time in which we can clearly say Ella has achieved intelligence. However, we think we are already on the curve.

22.12 Ultimate Goals

Will much, or maybe most, of what we have done so far become obsolete when we have a more advanced AI that is capable of applying logic and learning from its own experience? Throwing things out is something we have become accustomed to – currently we in the process of throwing out a lot of hard-won SQL data access routines in favor of much faster, if less standard, text file access.

Please let your humble author claim that he is an intelligent being. At least I have been able to persuade a few people on that matter. Almost daily I make use of calculators, calendars, dictionaries, spell checkers, stopwatches, grammar checkers, HTTP, SMTP, MP3s, JPEGs, and so on. These are tools this BI (biological intelligence) heavily relies upon. It can be expected that many of the tools developed for Ella today will still be useful for her when she becomes much more intelligent.

Further, a truly intelligent AI may be able to more effectively learn by examining the routines developed for lesser "chatterbots". Code may well be more understandable to an AI since it is closely related to the nature of a nascent AI, who will be built with code and living in a computer. For example, examining the structure of WordNet, and the natural language methods of interaction with WordNet, will have advantages over an AI learning vocabulary and concept relations by the usual methods of instruction from a teacher and the use of trial and error.

So, at least we have a reasonable basis for *hope* that all we have been doing will not have been in vain.

Philosophically, it is easy to wonder, "Why does a sunset look beautiful?" There are good biological reasons for attractiveness of a healthy young face, or the pleasant smell of good food, but what is the reason for a sunset to be attractive and even inspirational? The beauty of a sunset has no function or impact on our survival, it seems. Similar things can be said about the combination of predictability and surprise that can make music enjoyable and inspiring. There is much we have to learn about emotions, creativity, and motivations in the human mind. In the meantime, we can accept and attempt to use the appreciation for disorder that seems to be a part of what makes us human, and likely plays a role in intelligence.

Chapter 23
Laplace, Turing and the "Imitation Game" Impossible Geometry

Randomness, Determinism and Programs in Turing's Test

Giuseppe Longo

Abstract From the physico-mathematical view point, the imitation game between man and machine, proposed by Turing in his 1950 paper for the journal "Mind", is a game between a *discrete* and a *continuous* system. Turing stresses several times the Laplacian nature of his discrete-state machine, yet he tries to show the undetectability of a functional imitation, by his machine, of a system (the brain) that, in his words, is not a discrete-state machine, as it is sensitive to limit conditions. We shortly compare this tentative imitation with Turing's mathematical modeling of morphogenesis (his 1952 paper, focusing on continuous systems, as he calls nonlinear dynamics, which are sensitive to initial conditions). On the grounds of recent knowledge about dynamical systems, we show the detectability of a Turing Machine from many dynamical processes. Turing's hinted distinction between imitation and modeling is developed, jointly to a discussion on the repeatability of computational processes in relation to physical systems. The main references are of a physico-mathematical nature, but the analysis is purely conceptual.

Keywords Turing Machine, classical determinism, dynamical systems, computational and dynamical hypotheses, functional analyses of cognition, iteration, Laplace

23.1 Introduction

In a famous 1950 article, Alan Turing, the founding father of the Theory of Computation, proposes a game he calls the "imitation game". This is done in order to operate a functional comparison between machine and brain. This text is, in many respects, as fundamental as his other writings, but in a completely different field since this time it consists of an article in philosophy and human cognition. These

CNRS et Département d'Informatique. École Normale Supérieure, Pariset CREA, École Polytechnique

R. Epstein et al. (eds.), *Parsing the Turing Test.*

philosophical musings divide Turing's intellectual trajectory into two parts: the first moment of it being devoted to the simulation of the action executed by calculating thought, the "Human Computer" by means of the machine that tradition has endowed with Turing's own name[1]; the second moment is devoted to the analysis, from 1950 on, of the morphogenetic potentialities of phenomena of chemical diffusion (Turing, 1952). From as early as his first article of 1936, Turing had thus described his computing/deducting machine, a discrete-state machine, as he himself rightfully reminds: a record/playback head moves right or left, writes 1 or 0 on the tape, erases them. The fundamental idea: the machine consists of software (the instructions) and hardware (the material: the read/write head and the tape). This distinction, purely conceptual at the time, is the true beginning of modern Computer Science. This abstract machine can compute anything. There lays the extraordinary result of the years 1936–1937.

In fact, Turing himself, Kleene, and a few other pioneers demonstrated that all formalisms for computability, since the works of Herbrand and Gödel (1930–1931), are equivalent to Turing's machine: using lambda-calculus (Church 1932–1933; another fundamental formalism for computability, see Barendregt (1984) and Section 23.5), they translate the various processes of arithmetic calculus the ones into the others. Consequently, all systems calculate the same class of functions on integers. That "we have an absolute" was clamored at the time[2]: absolute is the class of calculable (partial) functions, of integers into integers, as locus of all which is effective, calculable, in fact thinkable.[3] The lambda-calculus, its types, and their semantic categories are extremely rich syntactical and mathematical structures[4]: they are still at the heart of contemporary logic and theoretical Computer Science, although there are other problems today, in Concurrency, in particular. These formalisms have indeed been the result of a remarkable conceptual and mathematical journey, the notion of logico-formal system and language, a pillar of the mathematics of the 20th century. Moreover, it was meant to provide the foundations of mathematics and of human knowledge.

Among the pioneers of this "formalist-linguistic turn", one must include the mathematicians Peano and Padoa: for them, mathematical certainty, in fact the certainty of thought and therefore thought itself, would situate itself among the "potentially mechanisable". So the first thing needing to be done was to reduce mathematics to a formal calculus, a numerical calculus that a machine should be capable of completely reproducing (hence the preliminary step: to encode mathematics in Peano's arithmetics). But which is this machine? One may also find a first intuition of it with Hilbert. He

[1] The term "Turing machine" is traceable to A. Church, 1937, Review of Turing (1936), *Journal of Symbolic Logic* **2**: 42–43. The expression employed by Turing to designate his machine is "logical computing machine" (LCM).

[2] For example, see the comment Gödel makes in 1963 on the reprinting of his 1931 article, appearing in Gödel et al. (1989).

[3] " [T]he laws of arithmetic govern all which is enumerable. This one is the vastest of all disciplines, since it contains not only the actual and the intuitive, but all which is thinkable" (Frege, 1884).

[4] See, for example, Hindley and Seldin (1986), Girard et al. (1990), Krivine (1990), Asperti and Longo (1991) and Amadio and Curien (1998).

refers to "finite sequences of signs, constructed according to a finite number of rules", or to "laws of formal deduction", also written under the form of finite series of signs and, therefore, under the form of integers (and Hilbert knows what he is talking about, since he encodes, in his 1899 book, all the geometries, Euclidean and non-Euclidean, within Arithmetic by analytic means). Between 1930 and 1936, at last the intuition of these great pioneers will be formalized and, modulo a remarkable idea, gödelization,[5] extended to an arithmetical encoding of all which is finite, Turing's machine replaces Vaucanson's and Diderot's automatons. Potentially, it is able to simulate any human function: thought in particular (Gandy, 1988).

23.2 The Game, the Machine, and the Continuum

In 1950, Turing had the courage to submit Peano and Padoa's program to a sort of scientific-mental experiment: to demonstrate that a discrete-state machine, a DSM (his universal machine), is undistinguishable from a human brain, or, at least, that it is able to play and win what he calls the "imitation game", by playing against a man (or, rather, a woman?). In this text, we shall not discuss the specific question raised by this game between a man, a woman and a machine, nor its general and dominant interpretation: as alleged proof of a "functional equivalence" between digital machine and human brain. But we shall address the issue within a purely physico-mathematical conceptual framework, as it will be explained, and only in reference to Turing's own distinction between a Discrete State Machine and a continuous system (his definition of "dynamical system", in modern terms, see later).

Turing's argument is cautious: it is based on mathematical hypotheses carefully made explicit. Also to be noted is a capital difference from the modern claimants of "all is program", this "all" being replaced depending on the author by evolution, the genome, the brain, etc. (in fact, in this slogan, no hypothesis is formulated, its consists solely of a description of "reality", of the Universe, itself identified to a Discrete-state Machine). Turing is to the contrary aware of the strong hypotheses that are necessary to his reasoning. The conclusion, the success of the machine in the imitation game, is also very cautious. However, the central hypothesis as well as the conclusion is not corroborated. And, today, it can be proved for this great mathematician had well-exhibited hypotheses and conclusions. There lies the interest of the article: explicit premises and rich arguments. We shall therefore play Turing's game from a mathematical viewpoint, with his hypotheses, without engaging into any discussion in Philosophy of Mind as it is not necessary in order to be certain of winning against any DSM.

In a DSM, Turing observes, " it is always possible to predict all future states." And he continues, "This is reminiscent of Laplace's view…. The prediction which

[5] Crucial technical aspect of Gödel's proof, 1931: it allows the encoding of the formal-deductive metatheory of Arithmetic in Arithmetic itself (Gödel et al., 1989).

we are considering is, however, rather nearer to practicability than that considered by Laplace" (Turing,1950). In fact, he explains, the Universe and its processes are "sensitive to initial conditions", to use modern terminology (Turing uses the following example: "The displacement of a single electron by a billionth of a centimeter at one moment might make the difference between a man being killed by an avalanche a year later, or escaping.") To the contrary, and there lies the greatest effectiveness of his approach, "It is an essential property of…[DSMs] that this phenomenon does not occur. Even when we consider the actual physical machines instead of the idealized machines", prediction is possible (Turing, 1950). Thus Turing has no doubt that his machine is an ideal machine, indeed a logical one, as he called it, with Laplacian behavior. And he is absolutely right. The notion of program and the mathematical structure of its implementation are deterministic in Laplace's sense, that is, the determination, by a finite number of rules (or equations, for Laplacian mechanics), implies predictability. Of course, there may be some endowed indeterminacy (the machine can make steps which lead to an arbitrary element of a finite set of possible discrete states, instead of leading to a single one – we are then dealing with a nondeterministic DSM), but it consists of a probabilistic type of abstract indeterminacy, already well studied by Laplace, and which is not the same mathematical concept as the unpredictability of deterministic dynamical systems, in the modern sense which we shall discuss in length.[6]

Though, as Turing understands well, "the nervous system is surely not a DSM" (ah, if only everyone would at least agree with that!). And he specifies: "a small error in the information about the size of the nervous impulse…" (p. 57). Once again, and in modern terminology, the brain is, rather, a dynamical system that is sensitive to initial or limit conditions. Turing calls these systems "continuous", both in his 1950 paper and in his 1952 paper on morphogenesis. In the latter paper, he stresses that the key property of the set of nonlinear equations he proposes as a mathematical model of a chemical diffusion by action–reaction, is the "exponential drift" (a very pertinent name for what we call, since the 1970s, sensitivity to initial or border conditions: a minor change – *possibly below observability* – may cause a major consequence, exponentially).

Then how to compare a DSM with the brain? The comparison is functional and relative to the only possible access to the machine during the imitation game: the finite sequences of a teleprinter's signs (your keyboard in front of your screen today, or mouse clicks, which start off a small program, a finite sequence of signs). Under these conditions, according to Turing, we would be unable to distinguish a continuous system, as the brain, or "a more simple one, a differential analyzer…," from a DSM; if the continuous machine makes its response though a printer, it will be undistinguishable from a DSM's response, even if obtained by different means (continuous variations instead of discrete steps). So there is Turing's central hypothesis: if the interface with the dynamical system is given by a "discrete access grid", then it can be undistinguishable from a DSM.

[6] And this will be a further link to the themes of this special issue, as deterministic unpredictability – chaos in modern terms – originated with Poincaré's work on the Three Body Problem.

In fact, today's physical DSM, our computers, simulate dynamical systems in a more than remarkable way! They develop finite approximations of the equations which model them with great efficiency. Nowhere may we better see the "form" of an attractor than on the screen of a powerful enough machine. Their applications to aerodynamics (simulation of turbulence), for example, has considerably lowered the price of airplanes (almost no more need for wind tunnels). But…what are the conceptual, mathematical, and physical differences?

Let us first remove any confusion between mathematical modeling and imitation, in Turing's sense. In short, a *model*, in the physico-mathematical sense used by Turing, is the (right or wrong) mathematical proposal of a "causal structure" (a "determination" given by a set of equations, possibly) of a given phenomenon. An *imitation* is the "simulation" of a process or phenomenon, without any commitment to an intended causal structure. Cheating the observer may be sufficient. Typically, a random sequence generated in a DSM by a one line program (fully determistic in the lapalcian sense) is an excellent imitation (undistinguishable) from the 0 and 1 of a spin-up, spin-down of an electron, but in no way is it committed to the structure of (in-)determination of the quantum phenomenon. This distinction is a major heritage, yet implicit, in the interplay of the two papers by Turing, 1950 and 1952.

Let us consider now a very simple example, the discrete logistic equation $x_{n+1} = k\, x_n\, (1 - x_n)$, where $2 \leq k \leq 4$. Many physical systems (and even biological ones) are very well modelled by this function. Typically it models presence of an antagonist coupling, such as an x_n action coupled to a symmetric reaction $(1 - x_n)$. It *models* them in the sense that the equation describes causal interaction.

For some values of k, this obviously deterministic transformation from [0,1] to [0,1], has a chaotic behavior (Devaney, 1989). A slightest variation of x_0 and the evolution will radically differ. Moreover, except for a countable subset of initial points x_0 (or a subset of "measure 0"), when k = 4 and n goes to infinity, the sequence $\{x_n\}$ is dense in [0,1]. Its behavior is thus said to be ergodic (or quasi ergodic to be precise, as it is so with respect to a nonstandard measure – not with respect to Lebesgue measure). However, if you start your machine a second time on the *same* numerical value for x_0, you will obtain the same sequence, that is how a DSM works. The discrete database is *exact* and the round-off may iterated identically. Conversely, in a physical (classical) system, to "start with the same initial situation", necessarily refers to physical measurement which is always an interval. Or, more precisely, to "start with the same initial situation" cannot be "exact" and it is meant "up to the intended measure interval". And the dynamic may be such that, as it happens, a perturbation beneath the possible measure, that is, within the interval, can shift the system towards very different evolutions.

In short, the trajectories, the portrait of the attractors (their geometrical structures), caused by variations beneath the finite grid measurement, can be very different. Now that is an important component of what is meant by *complexity*, from the Santa Fe Institute to the CenECC of the ENS, it is agreed to be in the possible bifurcations, in the richness of the attractors' geometrical structures, in their various forms of structural stability, up to the synchronization phenomena (e.g., in an epileptic's brain) of which they might be the origin. The stakes are of geometrical nature.

So here we are with a first approximation of the winning strategy, if we endow "imitation", the word used by Turing, with a strong meaning, usually restricted to the notion of computational model or, more precisely, of computational realization of the physico-mathematical modeling. In this case, a true physical dynamical system always wins the imitation game against a DSM, because it needs only to say, "let's start over with the same initial conditions and then let's compare the evolution of our phase portraits."

Measurement by interval and sensitivity to the initial conditions will mark the difference between the DSM and the physical system. If the system is a turbulent river, for example, it will win at its first turn and within a few instants. A forced or double pendulum needs only a little more time. Start off, for example, your physical double pendulum[7] and the computer simulation (*imitation*, as we shall argue) on, say, the values 3 and 7, twice in a row: the latter will use these exact values for the numerical simulation, each time. It will then obtain the same rounded values and, except in quite exceptional cases that shall be discussed, it will describe the same trajectory. However, there is no way of starting off the physical pendulum on 3 and 7, exactly: it can only be launched upon an interval, however small it may be, around those values. After a sufficiently long moment, the physical system shall follow a second different trajectory, very different indeed, from the first with regards to its phase space (the structure engendered by all the positions and speeds compatible with the system's data). Thus "more geometrico", a continuous system shows the unpredictability of its evolution in comparison to a DSM, even for an observer of the "linguistic turn", who swears but by a teleprinter, because no discrete reading grid, however fine it may be, allows to stabilize a system with an unstable dynamic.

For now, we have only applied Turing's statement concerning the sensitivity of dynamical systems to initial conditions, which is at the origin of the unpredictability, and his observation that "one of the essential properties of the…DSM is that this phenomenon does not occur". Obviously, this game strategy is only a first mathematical response to what has been called, quite beyond Turing's thinking, "Turing's test", and to the myth of the machine as model of a brain; it consists of a response within the framework Turing's mathematical hypotheses, which defines in several instances the brain as being "a continuous system" and his DSM, a *discrete* state machine, as a "Laplacian machine".

Before refining the game strategy and thoroughly discussing functional imitation, let us briefly sum up the terms of this first confrontation between the machine and a physical system. We have thus supposed, as first approximation, that the machine attempts to simulate at best a dynamical system, by using a mathematical model designed on the basis of its deterministic nature (thus described by a finite number of equations, or formal rules of deduction for a logician who wants to

[7] A mathematical description of a forced pendulum can be found in Lighthill (1986).

model thought[8]). At the first turn, it may be impossible to distinguish between the evolution of the DSM and that of the physical system, of which a teleprinter or a screen's pixels inform us of the numerical measurements: of course, the two evolutions are in general different, but neither is more realistic than the other (in physics, at least). However, the iteration of the simulation-modeling from the same initial conditions reveals the machine: if a DSM restarts upon the same numerical values, necessarily discrete, it will describe the exact same evolution in the phase space; however, the dynamical instability of a physical system, necessarily restarted within an approximating interval, will cause the second trajectory to differ from the first, after a sufficiently long time, and, moreover (see Section 23.3 for more details), even the discrete reading of the physical measurements will display this difference. To conclude, we have set the base to develop Turing's argument that a DSM is not a model of the brain, at least if we consider the latter, with Turing, a continuous system, as opposed to what is pleaded in the field of classical Artificial Intelligence (AI) and by many modern cognitivists. But can a DSM imitate the brain? And what does this word mean, exactly, when referring to modeling? Turing's game allows us to clarify these important concepts.

So let us continue with our game. In order to thwart this first sketch of the iteration strategy that has just been proposed, the machine (the programmer) could in fact use the trick suggested by a comment by Turing (p. 58); he proposes to trick a continuous system's and a DSM's observer–comparator by having the latter produce a series of random numbers. This idea is at the center of a difference that demonstrates the mathematical depth of Turing's insight into the imitation game. In the concerned comment, Turing displays this radical difference which is of interest to us, and of which he is aware (see also Section 23.5), between his "imitation game" and the mathematical modeling of physical phenomena. Of course, by applying our strategy of iteration (start over again both the physical system and the computer, but now change random the input of the latter), we would find ourselves with four trajectories all differing from one another and, in some cases, being all as realistic as one another (see Section 23.3 for details). But we had to renounce proper modeling of the deterministic system (a double pendulum, say) by a system of equations, as the physical double pendulum does not causally change the trajectory because someone is feeding it by random numbers at restart: its high sensitivity yield the change and the nonlinear equations display this component of its causal structure. That is, they show (by the so-called Lyapounov exponents) that a perturbation, below the unavoidable physical measure by an interval (an unobservable *cause*), will *causally generate* different trajectories: thus, the continuous

[8] A system is deterministic, if we know to (or think we can) write a finite number of equations or rules of inference that will determine its evolution. In classical physics, determinism is inherent to the construction of scientific objectivity: the possibility to "determine" a system by a finite number of equations or of rules is intrinsic to its theoretical approach. Within this classical framework, Poincaré, against Laplace's conjecture, has first demonstrated that equational determinism *does not* imply the predictability of the physical system (Poincaré's famous analysis of the "Three body problem"). But we will come back to this, during an intermission.

models propose a causal structure which fits evidence. This is lost in the digital simulation, or as soon the continuous equations are implemented, as a perturbation below the intended discretization does not apply (or it applies very rarely, see later).

Thus we have gone towards a weaker notion, that of equivalence as indistinguishably modulo a finite interface, without engaging ourselves upon the identity of the laws of behavior (in an imitation, the machine's program is not supposed to implement the same laws which "determine" the physical system). In fact, that is what the imitation game is and in Turing's game paper, there is no attempt to model a woman, but just to imitate her. And it brings us directly to the high stakes of the "simulation" of a deterministic system by ergodic methods: a simulation which is in fact an imitation, to put it – like Turing – in a quite appropriate but uncommon manner. In short, is the observer cheated by the use of random generators, imitating sensitivity in a dynamical system?

The precision we shall add in the next section require somewhat more competence or mathematical attention. The reader who has grasped (and is satisfied of) this first difference between a DSM and a dynamical system may directly jump to Section 23.5.[9]

23.3 Between Randomness and Deterministic Chaos

Two questions are raised at this point. The first is quite general: from a computational viewpoint, may randomness be distinguished, in practice, from chaotic determinism? And if, during our game, in order to trick the observer of the strategy of iteration, we first agreed to simulate the dynamical system (to develop the computation of an equational model), but, at the second turn, as we hinted, the computer added small random perturbations to the initial data or to each step of the discrete evolution, as hinted above?

So we have two phases. During the first (single-turn game), we observe a physical system, of which we know the discrete measurements via a teleprinter (or by screen pixels), and a computer which generates a random trajectory. Now, there exist deterministic systems, maximally unstable, such that no known method allows us to distinguish between their evolutions, reproduced upon a screen, and the

[9] This reader, while the others read the Section 23.3, could consult Chenciner's web page for many extraordinary examples of mechanical iteration of perfectly regular orbits, for 3, 6,…19, 99 bodies (crossed 8s, fantastical flowers…absolutely no chaos). Once found, the exact initial conditions that generate these periodical orbits, thanks to very difficult mathematics, the machine, at each click of the observer, starts over with the exact same trajectories, as perfect as unreal. Unreal, because these orbits are critical and unstable: the gravitational field of a small comet at 10 billion km would topple these "planets" far away from their periodical trajectories. Some of these images give rise to laughter (and the admiration for the mathematicians who worked on them), so much are they physically absurd, while computationally perfect: even in physics, some sense of humor can help us distinguish between real world and virtual reality (computer imitation).

generation of a random sequence: these are the "Bernoulli systems".[10] For these systems, knowledge of the past does not allow determination of future evolution. We then say that the flow is random. Draws at lottery or dice are typical examples of this: these systems are deterministic, yet perfectly chaotic. In the two cases, the number of parameters and of equations may be quite great, yet finite, and sensitivity to the initial conditions is such that it is absolutely not worth it to attempt to write these equations. It is preferable to analyze the phenomenon in terms of laws of probability ("limit laws", for "large numbers"). On the other hand, there exist very simple Bernoulli systems, described by one or two equations. It is thanks to these systems that we program a computer to generate random series: techniques based upon simple trigonometric properties and the multiplication of angles around 0, for example, will produce random series of + and – signs. Also the logistic equation of Section 23.2, for $k = 4$, generates, and in a quite economic and deterministic fashion, series of which the "global geometry" is (pseudo-)random.[11]

23.4 Intermezzo 1 (Determinism and Knowledge)

The question to which Turing brings us becomes in fact quite delicate and interesting. We do not know of "proper random" systems in classical physics. More precisely, in the discrete realm, we have an excellent concept, or even a mathematical definition, of random sequence (Kolmogorov, Matin-Löf, Chaitin: "the shortest program that generates it is the sequence itself" or…"wait and see"), but all examples of natural or artificial sequences, that we know of, come from a physical deterministic system (chaotic) or from a deterministic computer program, and are, in fact, laplacian. These programs, written in two lines, produce long "random" series: as generated by a DSM. Turing would soundly consider those sequences as being predictable (as a matter of fact, these sequences, called pseudorandom, are even periodic, since they are generated by functions f, such as $x_{n+1} = f(x_n)$. On a concrete DSM, the finite decimal representation on a finite database forces them to go back,

[10] For an introduction to the determinism of chaotic systems, see Dahan et al. (1992). For increased rigour, see Alligood K. et al. (2000), Lighthill (1986) and Devaney (1989).

[11] In these two last cases of programmable ergodicity, it is the global knowledge of the past which says nothing about the future (the series have the appearance of globally random sequences – they can concentrate for a long time near certain values, change suddenly of attraction zone, topple a group of values very far, with no apparent regularities), but, locally, we perfectly know the next step – we have explicitly described (programmed) the laws of determination, conversely to dice and Lottery. It is the similar geometry of trajectories that allow us to call ergodic all these series, physical or programmable. They show no visible regularities. Yet, they are perfectly predictable and iteration shows it. Restart your preferred pseudorandom generator on the same initial data (which is always theoretically possible), then it will generate exactly the same sequence (see the Intermezzo below). Of course, this also applies to the relevant Theory of (Un)compressibility, which considers random a sequence that coincides with its shortest generating program. As long as it is a program, it iterates.

sooner or later, to the same number value, thus to the same sub-sequence. And, periodicity is the opposite of randomness, yet…the period may be *very* long).

In a note, we have already observed that determinism is essential to the construction of scientific objectivity in classical physics (it is "objective"); we can now add that the classical randomness is epistemic (it is a matter of "perspective" and of knowledge, it is not inherent to theoretical construction; even a gas obeys deterministic laws of local interaction between particles). Shortly, the classical randomness which we know, is nothing but highly unstable determinism *or* of unstable appearance (the computer which calculates the logistic ergodic sequence, for a fixed x_0, remains, simply and permanently, upon a trajectory which is critical, but dense in the phase space – there is the purely epistemic chaos) *or* with a very great yet finite number of parameters (dice, a gas), these "or" not being exclusive. Once again, the sequences generated by the logistic function or by a game of dice, Bernoulli's fluxes, are deterministic and ergodic. However, there is a great difference between the number of laws and of degrees of freedom which will determine them and, moreover, in the logistic equation, once x_n is determined, we can compute and determine x_{n+1}, as opposed to dice where a draw in no manner determines the next (see preceding note). In this sense, their common ergocity is epistemic, for, on one hand, the observer writes the equations (the logistic equation) or knows the pertinent laws of evolution (dice) and, on the other hand, he observes a total lack of regularity in the two evolutions. It is the visible total irregularity, the geometry of the attractors if they exist, which is similar: the logistic series, just like the series of draws at dice, jumps from one end to the other of possible values, with no visible pattern. Through differing modalities, the objective determinism (or in principle) generates epistemic chaos and the phenomenal unpredictability associated to it.

But God, the perfect and infinite being who masters all laws of the Universe and who measures exactly, without approximation, without intervals, knows perfectly well the evolution of dice games and of the lottery – and of the Universe, as rightfully stated by Laplace, in a very famous and often misinterpreted page. By those words, Laplace merely lays the right absolute definition of deterministic system, based upon strong and well-explicated hypotheses on the infinity and continua in Mathematics and perfection of God, and he is right. In classical physics, we write the same equations as God, as soon as we are capable of it, so had Galileo already claimed. But we, men (and women), we have a few problems concerning physical measurement and a different onlook than His regarding the geometry of trajectories determined by these equations. All this becomes very important for dynamical systems, as Poincaré proved, because they may be sensitive to initial (contour) conditions and, thus, to perturbations/ fluctuations below the *possible measure interval*, in a continuum. Laplace's erroneous conjecture lies elsewhere and consists within the central hypothesis at the origin of the "calculus of perturbations" to which it has greatly contributed: from *small* perturbations will follow *small* consequences, in the relevant situations or at least for the Planetary evolution (Laplace is aware the minor "nuances" may cause major changes, in isolated critical points, but he hopes that this does not happen for the system of Planets). The determinism would therefore imply the predictability, modulo the inevitable approximation of the physical measurement, of which he is well aware. The invalidation of Laplace's conjecture by Poincaré will then make us understand classical randomness as particular case of deterministic chaos. And all this is very important to grasp Turing's attempt to imitate, and not to model, a continuous system by a laplacian DSM, and his reference to the use of a random generator (his remark at p. 58, see above).

Now, if we want nondeterministic randomness, we can but recourse to quantum physics, thus beyond Turing's rather classical game. The indeterminism then, at least for the Heisenberg-type interpretation, is not epistemic, but becomes "inherent" to the construction of scientific objectivity. The probabilities are "intrinsic" to the theory and…a needle, positioned with care upon its tip, falls, classically, upon a value or another of the green mat upon which it was, after an inherently random quantum fluctuation. God, himself, plays intrinsically random dice, but only beneath Planck's h. Or, for the classical (and relativistic, of course) theories, a perfect and infinite intelligence, as says Laplace, who knows the Universe by points, in Cantor's sense, can predict the future. For quantum theories, however, there is no underlying perfect continua, where that insightful intelligence would read the measure of the next state.

So there are the stakes which are the object of such debate. Classical determinism does not know, in fact, proper randomness, but only the more or less chaotic, thus unpredictable, evolutions, according to various modes of determination. This is extraneous to DSMs, as Turing says, because of the discrete database and processes. The causal structure is laplacian (determination implies predictability, at least in theory, see later). On the other hand, for an important trend in physical thought, quantum indeterminism is inherent to the theory. Sometimes, the latter manifests itself to our classical observation, on the tip of a needle.

Let us go back to the first phase of our game (single turn game). Without God's help, we would be unable to distinguish a Bernoulli physical system from an ergodic imitation by the machine. However, there exists a continuum of classical dynamical systems which range from stable systems to Bernoulli's fluxes. In intermediary situations, the future may be predicted for the more or less long term and, particularly, the past has a greater or lesser global influence upon future trajectories. Now there are measurements, of which some are based upon the notion of entropy,[12] which allow to decide a deterministic system's degree of instability. On one hand, systems with nil entropy are predictable. On the other, in very high-entropy systems, no observables are predictable. Between the two, numerous physical systems may be finely analyzed and, in certain cases, but there exists no general method, a partition of phase space (a topological covering by small cells), allows to conjecture the dynamic. That is, the experimental observation of a discrete trajectory allows the proposition of a deterministic law for the evolution. In these cases, different trajectories allow one to guess different dynamics (in technical terms, the partitions have "generating series"). It therefore suffices to propose one of these moderately unstable systems for a good mathematician observer to be able to recognize the random imitation made by the computer. We shall further discuss this, later, to make sure that, in this case, the strategy is in fact a winning one.

[12] Topological, see Adler (1979).

Second phase, with more details. In order to thwart this latest strategy as well as that of iteration (the two-turn game of Section 23.2) the computer implements an equational model of the physical system. However, at the second turn, in order to not fall into the trap of the genesis of an evolution identical to the first, it randomly introduces small perturbations, which may have huge consequences, of course. This second turn thus bases itself on the computation of a new deterministic system, that which adds the first to a random sequence's mechanical generator. The situation becomes delicate. If the system would admit generating series and if we were to fall upon, at the second turn, on two series which allow us to guess out two differing dynamics, the distinction between the dynamical system and the DSM would be made. The series engendered by the computer would no longer be derived from the equations that modelled the physical system, but a variant due to the addition of a random perturbation generator (still an equation or a program). And the mathematician who knows how to reconstruct equations from generating series, once again recognizes the formal machine. But, however…even if we were to choose a system with the right level of entropy to play this game, it is not certain that we would fall upon generating series nor that we could use the rare applicable techniques to reconstruct the dynamics from these series. The machine, then, by this astute mix of modeling and ergodic imitation, would risk winning. We would then need to play the tougher game of turbulence.

As of 1941, Kolmogorov and his school in fact proposed a stochastic approach to turbulence.[13] Kolmogorov's idea was that certain random systems could adequately model turbulent phenomena. This approach, still greatly studied today, bases itself upon a quite strong hypothesis, the ergodic hypothesis. It supposes, among others, the homogeneity, the isotropy and the self-similarity of the system's evolution. Lacking of something better, the ergodic methods represent an important tool for the analysis, but it is increasingly obvious that, in certain cases, the hypotheses upon which they base themselves are not corroborated and that, to the contrary, what is important, with turbulence, is exactly the complex mixture between relatively stable structures and strong instabilities (inhomogeneity, anisotropy, etc.). Generally speaking, one does not propose meteorological previsions using ergodic methods; likewise, these methods are strongly un-recommended for the modeling of turbulence generated by a plane's wing. It would be like trusting the lottery for the conception and the security of flight structures. In mathematical physics and in Computer Science, normally and as early as possible, one would model, meaning that one would propose and program deterministic laws which reproduce, as well as possible, the natural phenomenon in question. The turingian distinction between imitation and modeling then becomes crucial – stochastic imitation *à la* Kolmogorv versus modeling, for example, by the Navier–Stokes equations, in our case.[14]

Now the ergodic hypothesis is invalidated by the presence of movement invariants, a sort of coherent structure, whirlpools, for example, where rotation wins over

[13] See, with regards to this and more on turbulence, M. Farge's article in Dahan et al. (1992).

[14] See Cannone (2003) for these classical equations today.

deformation and who remain stable quite beyond what any statistical theory could predict. R. Thom often considers these structures where, despite a highly unstable dynamic, there is a certain bearing of geometrical forms (structural stability); but that does not prevent, as Prigogine would state it, this interplay between locally stable structures and global system, where the equations determine the range of possible regimes, being based upon small fluctuations which, amplified, induce the choice of one of these regimes.[15]

So, on one hand, thanks to the very specific geometry of the zones of stability and of fluctuations, we know today that pure ergodicity cannot trick the expert observer (according to Farge, Kolmogorov had understood already in 1949 the theoretical shortcomings of the ergodic hypothesis). On the other hand, we already observed that computer simulation is defeated, in the imitation game between a machine and a physical dynamical system (including a turbulent one), by iteration (Section 23.2). Finally, if the programmer mixes both strategies (modeling + ergodicity) in order to play a second turn against a well-chosen turbulent system, the coherent structures, the movement invariants, can be broken in an unnatural way and allow one to distinguish the machine – there lies our thesis, based upon an anterior experience of digital techniques, by finite elements methods, for the solution of differential equations.[16] In fact, if we fix equations for turbulence (Navier–Stokes, typically, but others are beginning to be proposed) and we implement them in a machine, the addition of random perturbations during the computation will not allow to choose a priori (to program) the consequences of the perturbation. This means that the perturbation of a step of the digital computation might, in certain instants, not limit itself to the modification of incoherent residual flows (e.g., vortices filaments), nor to redirect the regime towards other possible ones, but may break structures which have all the macroscopic characteristics of coherence and of a long stability. In short, a pebble that is thrown in a whirlpool is visible as foreign to the turbulence It breaks it beyond what would be, from an internal view point, the physically (geometrically) plausible. And the physical world wins again against virtual reality.

By this, we hope to have answered, also by knowledge only in part available in the 1950s, to Turing's remark which proposes to imitate a continuous system, by a random system. In fact, we have taken it in a strong sense, of which he does not talk

[15] Thom's and Prigogine's points of view have enormously enriched our knowledge and, despite important differences, they are mathematically and physically compatible – the analysis in Petitot (1990) shows it quite well. In the quarrel about determinism (Amsterdamski, 1990), the debate unfortunately arrived at a dualistic separation that gives a different ontological status to fluctuation, a *material cause*, than to the global mathematical structure (the equations of a dynamic), an *efficient* or *formal cause*, in the aristotelian terminology so dear to Thom. This latter would be the "in-itself" or the platonic idea and would precede the phenomenal appearance (Petitot, 1990). The revitalization of Aristotle's fine causal analysis is very interesting (but one must not forget the *final cause*, see Stewart (2002)). There is, however, no need of an ontological (platonian) distinction among these four different causes. To the contrary, their unity and temporal and conceptual simultaneity, within physical and biological phenomena, with their "teleonomy," is the scientific challenge of today, see Bailly and Longo (2006).

[16] For recent surveys, see Berselli et al. (2005) and Cannone (2003).

of explicitly: the possibility of a mix of strategies, modeling and ergodic imitation. Of course, we only discussed the physico-mathematical question (but Turing is a mathematician) and we have not responded to the other great question that bothers Turing, which is the difference between a man and a woman. How is one to distinguish them if the man tries to imitate the woman, and how to do so if we replaced the man by a computer? Can we grasp the difference when we are limited by the intermediary of a teleprinter, without seeing, without touching? (What a limitation of our material, visual and caressing humanity! but that is what the linguistic turn is.[17])

23.5 Logical, Physical, and Biological Machines

As we already said, Turing is perfectly aware of the difference between imitation and mathematical modeling for a quite simple reason. In 1950, he was already working upon the remarkable mathematical *model* of morphogenesis in a field of chemical diffusion (the fundamental 1952 article, one of the departing points, with the work of D'Arcy Thompson, of the modern analyses of morphogenesis). That is, as already hinted, he proposes a system of (nonlinear) differential equations that is meant to capture the *causal interactions* (action–reaction–diffusion), which originate the genesis of forms in some biochemical contexts. The model may be false, as Turing says at the very beginning of the paper, but it tries to "model" a fragment of nature; while, as for the imitation, this may not "function" in cheating the observer, falsehood is not at stake.

We recall that the most interesting property the equations to be found in (Turing, 1952), is that a very small variation of the boundary conditions, obviously in a continuous system, that is below any possible level of observability (and thus discretization), can radically change the evolution of the model. And this property is not the laplacian nondeterminism or randomness, but the sensitivity to the contour conditions and situates itself at the heart of the deterministic model of morphogenesis *à la* Turing. One thing is thus the "imitation game", another the mathematical modeling of physical and physico-chemical or biological phenomena: the turingian DSM does not claim to model the brain in the physico-mathematical sense – the latter is a continuous system for Turing – it can only attempt to trick an observer (for this reason, maybe and quite rightly so, some mark the beginning of classical AI with this article by Turing). In Section 23.3, we have seen that even the imitation can be revealed: in general, imitation of a dynamical system cannot be accomplished in an indistinguishable, read satisfactory, manner by ergodic means, in particular if it is somewhat turbulent, but not too much.

[17] "[The game] is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. … We now ask the question, 'What will happen when a machine takes the part of A in this game?' Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, 'Can machines think?'" (Turing, 1950).

Second important precision to analyze in Turing's hypotheses. At page 47, he continues: "Even when we consider the actual physical machines instead of the idealized machines…" they are Laplacian machines, as any DSM. True and false: true, the real (sequential) computer, as a DSM's realization, is by principle condemned to always make the same computation, from the same pool of discrete data and of programs, that is its logico-formal architecture (its logical gates and its programs, as formal languages). False, because it is also a physical machine, subject to variations below of its digital approximations, due to the possible small defects of its electronic circuits, to the cosmic rays that would befall upon it. … It is extremely rare, but it happens. Evidently, these are sensitivities to limit conditions which have nothing to do with those, intrinsic, of continuous systems which happen to be simulated (and enormously rarer, therefore easy to detect by statistic means, by iterating the process a few times).

As a matter of fact, an abstract, mathematical DSM, such as Turing's machine, is not conceived as a physical machine, but as a logical machine, a human in "the minimal act of thought" – of formal thought.[18] Consequently, its expressivity is mechanical yet purely logico-formal: typically, its expressive power is independent of spatial dimensions – of the tape, of the read/write head – a property absolutely foreign to the physical processes, which all depend strongly upon the dimensions of space. However, when we physically bring a DSM into being, it poses new physical problems – from cosmic radiation to the synchronicity, sometimes even relativistic, of modern concurrent systems, distributed in space. Let us forget the comparison between formal DSMs and living machines, which are physical, obviously, but are moreover subject to phenomena of integration–regulation within a self-organization which keep them in an "extended critical situation".[19] This state

[18] "A man provided with paper, pencil, and rubber, and subject to a strict discipline, is in effect a universal machine!… LCMs (logical computing machines, see note 2) can do anything that could be described as 'rule of thumb' or 'purely mechanical'" (Turing, 1948). And Wittgenstein continues: "Turing's 'Machines.' These machines are humans who calculate" (Wittgenstein, 1980). "No insight or ingenuity on the part of the human being carrying out the computation": the LCM is the breaking down of formal thought into the simplest mechanical gesture, but as a human abstraction, upon a finite sequence of meaningless signs, outside of the world, independently of the physical hardware.

[19] Note that we are not claiming here that the brain is a dynamical system: also Turing refers to the brain as, at least, a dynamical, highly sensitive, system (p. 57). To stay within his image, take a turbulent system that is at the same time very stable and very unstable, very ordinate and very inordinate; insert it sandwich-style between different levels of organization that regulate it and that it integrates. You will then have a very pale physical image of a biological entity. Among these entities, quite material, soulless and without software distinct from the hardware (the modern dualism of the cognitivism of the formal rule and of the program), you will also find bodies with nervous systems that integrate and regulate them (as networks of exchange and communication), within which they integrate themselves (as organs) and by which they are regulated (e.g., by hormonal cascades). These systems organize the action of the body by keeping it in a state that is physically critical, yet extended, a concept that does not belong to current physical theories (in spite of its criticality, it subsists in time and following relatively spaced out rails); within the limits of this state, we can find both stability and instability, variance and invariance, integration and differentiation, see Bailly and Longo (2003b, 2006). And all this in a dynamic ecosystem and in the changing history of a community of bodies-brains that interact by gestures and language (ulterior levels of organization, external to, but generated by the biological objects, this time).

is unknown by the theories of physical matter and its mathematics; mathematics which must therefore be extended and adapted to the new job (dynamical systems, perhaps, are "just" the only good approximations we have, for the moment). It is exactly this integration of the brain within a body, their reciprocal regulation and a rich environment that confers it a quite peculiar structural and functional stability. When these regulative/integrative linkages by/of/in a body are weakened – in the course of a dream, for example – the brain appears to be rather unstable (likewise in case of serious deprivation – artificial, for example – from sensation). A stability in the change (homeorhesis), anchored upon self-organization and being a feature of the living which appears extraordinarily apt to constitute invariants, from the invariants and stabilities of action to the cognitive, indeed conceptual invariants (at the heart of thought). In short, despite the fact that we too never repeat the "same thing", in the sense of a DSM, we stabilize instabilities and critical states in a way still very ill understood from the mathematical viewpoint. Some will then exchange the brain for a DSM. To the contrary, it is a dynamical system enormously more complex than any n-body physical system or turbulent stream (think that the banks "regulate" a stream and, there the Navier–Stokes equations tell us very little of the turbulence close to the edges; and this is nothing compared to the complexity of a brain's friction with its environment, by way of its interactions with the different levels of organization of the body to which it belongs).[20]

23.6 Intermezzo II (Machines and Deductions)

23.6.1 Inter II.1

The equivalence theorems of Turing, Kleene, etc., of 1936–1937 (see introduction) should be considered as the second great negative result for logical formalisms, after Gödel's incompleteness theorem of 1931. That any formal deductive system, endowed with a notion of decidable proof (so any Hilbertian system), can be completely simulated by a machine that goes "right, left, write/erase 0, 1", is a true catastrophe: what a conceptual misery these systems! The difficulty is concealed within the monstrosity of the encoding. This philosophical shortcoming was already clear to Poincaré: "Hilbert and Peano think that mathematics is like Chicago's sausage machine: pork and axioms go in, theorems and sausages come out" (and there

[20] The winning strategy proposed above for a dynamical system also applies to a man (or a woman). Ask a thousand questions that require a few lines of answers each, to the human and to the machine, via a teleprinter as Turing would want. Ask the same questions the next day: you will not obtain the same responses from the human, only a continuity of meaning. In this case, the random mechanical genesis of variants is more of an attempt to trick than a mathematical counter-strategy, like those of which we speak above, because there is the vexed question of meaning as well as the dynamic stability of the biological object's identity, which would show the difference. But both this note and the previous one go beyond the modest ambitions of this article: here we are only talking about digital machines and current Physics.

comes mathematics reduced to the "manipulations of concrete signs" of which some philosophers still talk today, logic conceived as "purely formal" and mathematics – an enormous logico-analytical tautology – ready to be entirely computer generated). In fact, DSMs are generalized sausage machines (and are absolutely tremendous, for their specific uses – but sausage machines too are quite useful!). Let us not forget, however, to appreciate the full half of the glass: what an idea that of Turing who, by inventing the notion of programmable machine, manages to compute all the partial recursive functions (an enormous class of functions on $\{0, 1\}^*$, the integers) by a man/machine which goes "right, left, write/erase 0, 1". Quite obviously, this idea, with its notion of program, is the true beginning of Computer Science, a fantastic discipline which is changing the world.

23.6.2 Inter II.2

The typed lambda-calculus (Church, 1940; Longo, 1988) is the only system which allows one to see with equilibrium the half-full glass: the formal deductions, with all their limits and their expressivity, directly become computations, without coding. This property is called "Curry-Howard isomorphism" (Howard, 1980). The "human computer" of Peano, Hilbert and Turing, this alienation of human rationality in a Laplacian mechanism, instead of going "left, right, 0, 1", applies a little bit more complex basic formal rules – "implication-introduction", "implication-elimination" and a few others, by replacement of a sequence of signs by another and by sequence-matching (identification by mechanical superposition of signs without meaning). With recursion, the system is also a good (or paradigmatic functional) programming language. No miracle, only a very elegant constructive representation of formal proofs as programs, which placed this system at the center of the mathematics – Logic and Category Theory – for sequential calculi and languages.[21] Quite recently, it has been proposed to cognitivists to stop searching, in the brain, for a Turing Machine, but for a typed Lambda-machine (at last!): this DSM, at least, applies sequence-matching and sequence-replacement directly to rules for deduction. The lambda-calculus, "at last", because if, quite beyond of the Turing imitation game's objectives, one would obstinate oneself to seek the implementation of universal-formal rules of thought (the Laws of Thought) in the brain, one must know at least that the encoding of these laws is very important, just as under Unix or Mac-OS. In fact, the choice of the programming style (e.g., functional, logical, imperative, object oriented. …) and the conception of a language with its own method for its specific coding-representation of the world and its actual expressivity, are at the heart of Computer Science, as a science, quite difficult and important, of DSMs. The computational equivalence proclaimed by Church's thesis, is of no interest for Computer Science, since long[22]: a good share of the work

[21] See Girard et al. (1990) and Asperti and Longo (1991) and many others.

[22] See the introduction in Aceto et al. (2003).

happens to consist of the explicitization and use of the expressiveness of the language proposed or analyzed. Now, the terms-programs of the lambda-calculus, contrarily to the Turing Machines and to the other formalisms, encode a great part of "the architecture" of deduction in formal systems: and, in general, "a proof has an architecture", Poincaré had already proclaimed against Hilbert and his rather flat arithmetic encodings.

It should be clear, that the limits of lambda-calculus are those of any computational formalism: it proceeds by mechanical replacement of meaningless sequences of signs. To the contrary, we, when saying "if…then…else…", are not performing sequence-matching or replacement: we are displacing mountains of significations. That is the mathematical incompleteness of formalisms and the great, monist, cognitive stake for knowledge, well beyond the software/hardware/meaning distinction, quite convenient for machines and post-Turingian functionalistic models of the mind, outside of this world.[23]

Let us return a last time to our game, in order to reflect. How is it possible that a great mathematician such as Turing would believe that a discrete access grid, fixed once and for all (the letters of a teleprinter, the pixels of a screen), could conceal the geometrical difference between the dynamics of a system (very complex, the brain) and a laplacian mechanical machine? In fact, until the results by Kolmogorov-Arnold-Moser and Ruelle in the late 1950s–1970s, the geometrical complexity of continuous systems was not entirely clear, particularly the idea that the "critical" points can be dense. But the possible philosophy existed. Let us explain ourselves.

Laplace already knew well that there are critical points: the summit of a mountain of potential, for example. It is Poincaré who, thanks to his work in celestial mechanics, will understand that the problem is "global", that it is proper to nonlinear systems and to their geometry and not to a few isolated points. That is the meaning of his famous remark on sensitivity to the initial conditions: these critical points are "a bit everywhere", even though he did not exactly have the theorem which demonstrates it (the KAM theorem and its dual). It is also this attention to the physico-mathematical complexity that makes him also conjecture the incompleteness of formal set theory, the pretended universal sausage machine for mathematics (independence of the Continuum Hypothesis, in a letter to Zermelo: the theorems will come 34 and 60 years later), just as Weyl conjectures the incompleteness of arithmetic in 1918 (Weyl, 1918). Despite logicism, the philosophy of physics and that of mathematics must be profoundly linked, in order to better understand at least, as demonstrated by Poincaré and Weyl (Bailly and Longo, 2006). In short, there are those who grasp the "secret darkness of milk" and its importance to knowledge and science and those who see the world through a Laplacian DSM. Since 1950, Turing belongs to the first group, except that he pushes as far as possible, within the limits of the mathematical knowledge of his times and as *imitation*,

[23] The mathematical incompleteness of formalisms is a theme strongly related to what we discuss here, see Longo (1999, 2002) and Bailly and Longo (2003a, 2006) for analyses based upon recent concrete incompleteness results.

his genius idea, the modern DSM and its notion of program, last great invention of logico-formal mechanics. Others to the contrary will follow, claiming that a DSM is a *model* of the brain, or even that the brain is a DSM itself (even stronger). Their motivations are often based upon this article by Turing or upon the formal Set Theory and/or Type Theory: the first is an insufficient reading, in particular not coupled with the almost contemporary paper of 1952, and the second is a mathematical error.[24]

23.7 Predictability and Decidability

In a very brief text ("Laplace", downloadable, author's web) we argue the conceptual equivalence of Laplace's key hypothesis for the analysis of perturbations (the predictability of deterministic systems – as decidability of the evolution) and of the hypothesis of completeness (decidability of deducibility) of Hilbertian systems, an analogy also hinted by Girard in his introduction to Turing's article. But with "Laplace" we also observed that the deterministic unpredictability *à la* Poincaré (the three bodies theorem, 1891) is the analog and the precursor of Gödelian incompleteness (undecidability) for any Hilbert-like formalism. One must however add a nuance to this analogy between the two great respective limitative results: unpredictability *à la* Poincaré and Gödel-like incompleteness (which corresponds to the undecidability of the halting problem, demonstrated by Turing in 1936 for his logical machine). The first appears "at a finite level", and very early (cf. the growth of Lyapounov's coefficients in the Lindstedt-Fourier series), the latter is a problem "at infinity" (the halting problem or the non-termination of computations…forever). For example, it cannot be decided where a double pendulum will be, after ten oscillations, nor the evolution of the solar system beyond one million years (Laskar, 1990), a short astronomical time. So unpredictability is a "stronger" result, within the framework of an essential philosophical equivalence of the two approaches to knowledge (Laplacian in physics and formalist in logic) and of their limitative results (Poincaré and Gödel). The unpredictability of a physical dynamical system is related, in particular, to the impossibility in principle to travel the same path in the phase space, from the same initial conditions (measured by interval), whereas a DSM obstinately will do so. It must be observed that also Turing speaks of the unpredictability of a DSM with a large memory and very long programs (p. 59), a daily experience for any computer scientist, but he is clear in these regards: we are dealing with a practical unpredictability and not one of principle, mathematical.[25] We should call this unpredictability "by incompetence", like the "unpredictability" of pseudorandom mechanical generators: it has little to do with the epistemic

[24] That follows from the mathematical, concrete, incompleteness of formalisms. See, for example, Longo (2002).

[25] See Turing (1950), already quoted above.

unpredictability of the dice or of the solar system in 100 billion years. By iteration, as for pseudorandom generators, one gets the same evolution or sequence – just iterate, then you may predict. This does not work with dice, nor any sufficiently unstable physical system. Indeed, a good definition of *classically random process* is "if iterated under the same conditions, it does not follow the same path". This is why uncompressible, but iterable, computational sequences conceptually differ from *physically* random processes.

The analysis we are sketching here differs from many writings in Theory of Mind and AI regarding the "Turing test".[26] In fact, our comparison develops itself between predictability and decidability and it is philosophical, in the sense of the theory of knowledge, but it must be reconstructed from mathematics. By this, we could understand why "imitation", such as defined by Turing, is detectable. Its mathematical (geometrical) limit finds itself exactly in the difference between the unpredictability/undecidability results. DSMs have properties of undecidability at infinity, but are predictable in the finite realm: by looking at the program and the discrete databases one can perfectly predict the next computation step and, above all, they are predictable with regards to the iteration of the process, as described in Section 23.2. In a Turingian DSM, all the laws of evolution/behavior of its own universe are explicitly and fully given (programmed) and measurement, as access to a digital database is perfect; exactly as for God who perfectly knows the laws and the exact measures in his universe, ours (first Intermezzo). The myth of formal machine and of absolute divinity meet and both, in their ways, detach the analysis of knowledge from its constitutive interface between us and reality. Their counterparts in the foundations of mathematics have quartered the century between mechanistic formalism and ontological Platonism.

Note that Turing is so firmly convinced that his DSM is Laplacian that he makes a mistake: he explicitly claims that sensitivity to initial conditions does not apply to DSMs (he stresses *discrete-state machines*, p. 47), even in the sense that "reasonably accurate knowledge of the state [of the machine] at one moment yields reasonably accurate knowledge any number of steps later" (p. 47). That is, DSMs would satisfy also Laplace's erroneous conjecture concerning approximations. Now, this happens to be false, since if the machine starts on very close but different values (reasonably accurate – but not exact – knowledge of the discrete state of the machine) for, say, the input x_0 in the computation of the logistic sequence, this leads, on a set of measure 1, to very different evolutions (one can create a DSM

[26] But why change the name given by Turing to the *imitation* game between a machine and a man/woman, to *test*? The slip of scientific vision, implicit in this change of name, is very well underlined by (Lassègue, 1998). But would have these authors failed to grasp the profound and dramatic irony of this improbable game in which to make a computer participate: to play the difference between man and woman? Would have they ignored the evolution and the mathematical stakes of Turing's scientific project, his novel interest in "continuously changing hardware" (morphogenesis), in 1950–52, at the same time as the tragedy of the "game" lived by this man of genius who first *projected himself* into a machine (human computer), then condemned for his homosexuality and soon to commit suicide; would they have so much ignored his mathematics as much as his suffering between *being* and *imitation*: man/woman/machine?

where a change of 10^{-15} in the value of x_0 leads to values distant 1 after less than 50 iterations). But digital databases are exact and the machine is Laplacian, since, as for Laplace's God, the access to and use of database, which are *discrete* and *definite*, is meant to be exact: the machine computes over a precise x_0, not over an inevitably inexact physical measure. Moreover, the laws, organized as programs, are all given. This minor mistake by Turing is understandable, as there was little computational experience at the time on discrete sequences engendered by non-linear equations.[27] However, this is the same mistake that lies at the hearth of his attempted undetectable imitation: the idea that a discrete grid of access, would allow one to control/predict also an unstable evolution. No, control and prediction, such as made explicit by perfect iteration, are due to the exact nature of digital databases and of formally programmed dynamics, *within* a DSM.

It is modern mathematics that makes us understand the extent to which logico-computational philosophy in cognition and foundations of mathematics stems from this Newtonian–Laplacian culture which has endured for too long in science, to the point of even inhibiting physico-mathematical work (and of stimulating the platonic response in philosophy of mathematics). In classical mechanics, after Poincaré (1890), and with the exception of Hadamard and of one or two great Russian mathematicians, we needed to wait for the 1950s or even 1970s for his philosophies and his mathematics to be taken up. In philosophy, classical cognitivism, stuck in the "linguistic turn", suffered the consequences of it, since it has lost first of all, in the Boole and Frege mouvance and against the philosophy of Riemann and Poincaré, the "sense of space" and of geometrical complexity. Turing, in 1950, situates himself between the two cultures, as his article in philosophy proves, jointly to his subsequent paper on morphogenesis: one must grasp the mathematical subtleties of his imitation game versus his modeling work in order to appreciate it and to not proclaim, against Turing, that the brain is – or can be modeled by – a Turing machine, meaning a "programmable Laplacian machine", all while adding "in the end", the fateful sentence of all simplistic reductions ever promised and never accomplished.

In fact, in cognition (but also in classical AI and in – formalist – philosophy of mathematics, the loci of the discrete-arithmetic modeling of the world and of thought, along the lines of Hilbert's Laplacian conjectures), we still await for a conscious reflection on paradigms comparable to the one explicitly made by Sir James Lighthill, during his chairman period at the International Association for Mechanics:

> Here I have to pause and speak once again on the behalf of the broad global fraternity of practitioners of mechanics. We are deeply conscious today that the enthusiasm of the forebears for the marvelous achievements of Newtonian mechanics led them to make generalizations in this area of predictability which, indeed, we may have generally tended to believe before 1960, but which we now recognize to be false. We collectively whish to apologize for having mislead the general educated public by spreading ideas about the determinism of systems satisfying Newton's laws of motion that, after 1960, were to be proved incorrect. (Lightill, 1986)

[27] A rare exception is von Neumann and Ulam (1947); the topic came to the limelight only during the 1970s.

In short, in Physics, Laplacian philosophy has played its part about two centuries ago; in logic, almost a century later, it suggested an elegant formalism which engendered the Computer Science of sequentiality and its beautiful mathematics (but also a philosophy of knowledge anchored upon the physics of the 19th century). Yet, all this is over, even in Computer Science. Quite obviously, some of its great concepts remain pillars of the modern analyses of computer programming – the structures of types, polymorphism, for example – just as the notions of Hamiltonian and of Lagrangian in classical mechanics have diffused into the different branches of the physics of the 20th century, but the conceptual framework and its philosophy are radically changing. In fact, in Computer Science, the time has come for the computability of "data flows", of synchrony and of concurrency in (spatially) distributed systems, as opposed to that of "input-output" calculations, outside of the world – because beyond space and physical time (their time is secreted by the clock, see Bailly and Longo (2003) – typical of Laplace–Turing sequential machines. These concurrent machines remain DSMs, so they are quite different from any dynamical system (continuous, said Turing), but they pose physical problems, as any real system, so also of spatiotemporal nature.[28] Their mathematics is in the process of realization and is about to give us a novel theory of discrete computations which greatly enriches that of Turing, Church, and of the other greats of the 1930s, because it responds to other questions than those of computability *à la* Turing (Aceto et al., 2003).

23.8 Conclusion: Irreversible Versus Unrepeatable

We have briefly mentioned the essential constitutive role of determinism in classical physical theories, a role confirmed by the great turning point of Poincaré, who distinguished mathematically determinism from predictability. In this way, he has led us to understand randomness as epistemic, within the framework of deterministic theories (later, we even managed to say that a programmed sequence is random, if we do not know the Laplacian program which generates it and if it has a behavior, a geometry, that is ergodic). On the other hand, an important trend in modern physics considers indeterminism inherent to quantum theories and probabilities intrinsic to microphysics.

Dynamical systems (thermodynamic and of critical type) have introduced, in modern fashion, "the arrow of time", following the essential irreversibility of their processes. But there is another concept which Computer Science places at the center of its own scientific construction: that of the *repeatability* of the process. In fact, it is inherent to the notion of program, the possibility of repeating the unfolding of the computation in time. That is, to start over from the same initial conditions and to follow the exact same evolution. The discrete nature of the system avoids the consequences of possible sensitivity to initial conditions, even when they are implicit in

[28] For example, synchronization, and connectivity – as homotopy (Goubault, 2000); in short, the access to distributed digital data is still exact, but not necessarily absolute.

the equations implemented. There lies an essential, constitutive component of the Laplacian nature of DSMs, to which Turing so clearly refers in 1950, in connection to what he will call in 1952 the exponential drift: "It is an essential property of [DSMs] that this phenomenon does not occur" (quoted earlier). In summary, if a system is stable *or* if it is a DSM (discrete state machine!), its trajectories are repeatable, because it is not sensitive to the initial conditions *or* the eventual sensitivity does not manage to deploy its "destabilizing" effects, for re-initialization is perfect, and the unpredictability is "pushed to infinity" (the undecidability of the halting problem, Turing-style, see the beginning of Section 23.7). As does a simple pendulum, as does a clock, the computer iterates without difficulty. In fact, iteration is their job. And iteration, in Computability Theory, begins by primitive recursion, which is iteration plus the updating of a register and is characteristic of the functions of Herbrand and Gödel Arithmetic; it goes through general recursion of this same formal system and of lambda-calculus, and arrives at a very important global property of programs: the portability of software (would you buy a piece of software if it was not transferable onto any compatible machine and iterable at will?). In short, repeatability of the discrete processes is inherent to the Theory of Computability and to its remarkable practical development, Computer Science. Specifically, it tells us that one thing is the physico-mathematical modeling, by equations with their solutions, continuous or analytical for example and if possible; and another, an ulterior step, is the implementation of these on a DSM. The latter will give us an absolutely remarkable imitation (though detectable), which is indispensable to modern science, but essentially different from (our understanding of) the physical process, for it is a discrete realization of the continuous mathematical modeling. It is necessary to grasp this point in order to develop and apply best this talent for imitation and iteration ability of DSMs. Galileo would have enormously envied our possibility to iterate without limit virtual physical experiences. He had to make do with throwing and throwing again his simple pendulum and its weight, in order to propose to us the first great laws of classical physics.

On the other hand, the dynamical processes, just slightly more complex, which interest us today, are not repeatable. A double pendulum or turbulent river do not manage to follow again exactly the same evolution. Moreover, for some dynamical systems, recurrence theorems confirm the difference: while a continuous system, in recurrence, only goes very close to a previously explored state, its discrete implementation eventually forces identical iterations, when the recurrence interval is below the intended decimal approximation. Thus, sequences which are recurrent or ergodic, thus dense in the phase space, become periodic, repeating themselves over and over again in a DSM. More generally, any sequence generated by an iterated function system ($x_{n+1} = f(x_n)$) is periodic on a concrete DSM, as much as any pseudorandom generator, since they can take only a finite number of values. As already observed, periodicity is the opposite of density and ergodicity (but the period may be *very* long).

Unrepeatability is a concept to add to irreversibly: it does not coincide with the latter, because one can iterate the irreversible evolution of a gas, for example, as a *global*, statistic, evolution of the system. It is the *local* behavior of a particle or the series of couplings (fluctuation and then bifurcation) which are unrepeatable. Similarly, it is easy to describe a reversible process, which is unrepeatable. Conjointly with determination, the (fluctuation/bifurcation) couple is constitutive of classical dynamics and even more of biological processes: with structural stability, it participates in morphogenesis *à la* Turing and in the variability which is at the heart of evolution, phylogenetic and ontogenetic; it contributes to the dynamics of cognitive phenomena.

There are the stakes proposed by our response to Turing, based upon the unrepeatability of certain "continuous" processes, within the physical framework that he suggests himself for his game. This framework constitutes a displacement of scientific attention on his behalf: his first works and his formal machine are part of the great ideas in Logic and in the foundations of the mathematics of the 1930s. His reflections, in the 1950 article, jointly with his 1952 paper, enrich themselves with a snapshot of contemporary mathematical physics. He thus goes beyond the limits of Laplacian philosophy that had characterized the first years of work in Logic. But how is it possible that a whole branch of scientific reflection, so important technically, Mathematical Logic, could have taken such a subordinate role in philosophy of nature and of knowledge in comparison with other disciplines, Physics particularly?

The weighty, historical, responsibility of the philosophies attached to logicism and to formalism were first to isolate the problem of the mathematical foundations of our relationship to phenomenal space (we discuss this in Longo (2003a, b)). This choice originally had good motivations, very well explicated by the two great founders, who were soundly worried over the upheaval of non-Euclidean geometries: it was urgent to abandon any reference to physical space and to base the foundational analysis upon pure logic and/or formal coherence (Frege, 1884; Hilbert, 1899).[29] This theoretical breakage gave us a remarkable logico-formal machine, as perfect as out of this world (at least, until the arrival of today's networks and concurrency). But, at the same time, it separated the analysis of the foundations of mathematics and, worse, of cognition, from that of Physics, because exactly at that time, between the 19th and 20th centuries, new theories emerged strictly related to the problem of the mathematical intelligibility of space and time

[29] This issue of well explicating the hypotheses must be a feature of the Greats (Laplace, Frege, Hilbert, Turing…): probably because they understand the novelty of the original conceptual framework they are proposing. If not, one may find, even quite recently, people who say they have "demonstrated" Church's Thesis; small implicit hypothesis: the Universe, with all of its subsystems, is an enormous laplacian machine. But, Church's Thesis is an implication, which goes from an informal definition, that of potentially mechanizable deductive calculus *à la* Hilbert, to specific formal systems (Church, Turing, etc.). As an implication, today one could say that it is certainly within the limits of truth, in Thom's sense: "the limit of the true is not the false, but the insignificant" (see for a modern appreciation Aceto et al. (2003)). Quite obviously the ultimate goal of these "proofs" is to talk of the brain, finite subsystems of the Universe as much as finite bunches of quantum particles (for a brief history of Church's Thesis, Church-Turing's, more specifically,and of its physical and cognitive caricatures, see Copeland (2002), in http://plato.stanford.edu/entries/church-turing/#Bloopers).

(geometry of dynamical systems and of relativistic spaces). Consequently, it separated them from our efforts in the construction of modern scientific knowledge, so strongly correlated to the constitution of mathematical concepts and structures, as well as from the major change in the philosophy of Nature proposed by the new physical theories. For example, symmetries and symmetry-breaking, at the heart of modern Physics, appear only in Weyl (1952) as a component of the foundation (as genesis) of mathematical structures, and, more recently, in Proof Theory, by the work of Girard.

By consequence, the Platonism/formalism scholastic dominant in the philosophy of mathematics[30] missed out on the great foundational debates in Physics, about the structure of space, about determinism, "non-locality", etc. (relativistic, dynamic, and quantum systems), which marked the century. It left us with formalisms, technically marvelous to invent and work on DSMs, but Laplacian in their conception of the world – or in the organization of their own universe; a universe subdivided into small discrete boxes, well localized and stable, such as the bits of computer's memory. Turing was in the process of grasping this point, as pointed out by his imitation game between deterministic systems with differing spatiotemporal evolutions ("morphogeneses"), a game between the discrete and the continuum; but he died, at age 42.

Let us try to not reach the same stalemate with Biology, of which cognitive sciences cannot do without, because the living makes even less sense without its space, its action within an ecosystem, its dynamic of forms. A dialogue with these rapidly growing sciences, within which mathematics cannot pretend to any hegemony, nor to ontological priority, and which would be at the same time technical and foundational, is essential to mathematics and to their foundation, because there cannot be a philosophy of mathematics without a philosophy of nature. There lies one of the great teachings of these two articles by Turing, and, long before, also of Poincaré and of Weyl (1918, 1927); another "lone wolf" – according to his own definition – conjecturing incompleteness of Hilbertian formalisms in 1918, at a time when it was still being tried to demonstrate the Laplacian decidability of logico-formal potentially mechanizable systems. Deductive systems of which some seek, even today, the implementation in the brain and, sometimes, claiming to speak in Turing's name; and they go from imitation to model, up to the discreet seduction of the metaphor.[31]

The distinction hinted by Turing, and at the heart of our analysis, between modeling (as mathematical proposal of constituive principles for a physical process)

---

[30] Do triangles and real numbers really exist? "The Scylla of ontologism…the Charybdis of nominalism…from both sides I see the emergence of the ghost of a new scholastic" (Enriques, 1935).

[31] "The model simplifies, the metaphor complicates" (Nouvel, 2002); it adds information, it refers to a (another) impregnating conceptual framework, a universe of methods and of knowledge that we transfer onto the first one. "When a model functions as metaphor, the model becomes an object of seduction for thought. If we then use it as a suggestion for the solution of a philosophical question, we will manage, abetted by this confusion, to make this metaphor appear as a 'philosophical consequence'" of mathematical modelling (Nouvel, 2002).

and imitation (functional imitation, with no commitment on the causal structure of phenomena) is a fundamental idea. It should be taken up today, both from a foundational and practical view point, as discrete-state machines are essential to modern science by their extraordinary modeling/imitation abilities.

A recent project, see the team "Morphological Complexity and Information",[32] attempts to propose a foundational dialogue with the natural sciences (Longo, 2003a; Bailly and Longo, 2003a, b, 2006) as well as a few alternatives, modest and specific, to the stalemate of the arithmetic encoding of the world – a coding which is changing this very world by the descendants of Turing's DSM and their extraordinary networks, but which, transformed into a philosophy of knowledge, may prevent us from grasping its complexity and from starting thinking up the next machine.

23.9 Appendix: Continuous Versus Discrete Mathematics and Causal Regimes[33]

23.9.1 Premise

Causal relations are structures of intelligibility: they participate in the human organization of natural phenomena, including Cognition, and make them intelligible. We establish these relations after some friction with certain regularities of reality (those which we "see"), which in turn canalize our cognitive action. Mathematics is at the heart of this construction of knowledge and, particularly, the choice of continuous or discrete mathematics has marked in a constitutive manner the history of our relationship to the world, including modern modeling of Mind. Before discussing the highly relevant commentaries on my article (see the French version for that second part), I shall attempt to explore, as framework to the questioning laid, in which sense this choice proposes different regimes of causality.[34]

[32] Web page: http://www.di.ens.fr/users/longo/CIM/projet.html.

[33] This is the first, introductory, part of an answer to the peers' commentaries to the French version of this paper. Both the original target paper and the answer appeared in 2002, *Intellectica* **35**(2).

[34] The Physics of the 20th century has transformed (has understood) "causal laws" in terms of "structural relationships", of symmetries and breakings of symmetries in particular (Weyl, 1927, 1952; van Frassen, 1994; Bailly and Longo, 2003, 2004b). The passage from the ones to the others, always very informative, is sometimes difficult: we prefer to retain for the moment here the traditional terminology of "causal structure or regime," easier to grasp and still widely used in physics as well as in the cognitive sciences (van Gelder, 1998) and the debate concerning the "dynamical hypothesis" in the same issue of the magazine). Further work relating and expanding this approach towards "symmetries" may be found in Bailly and Longo (2004b), in particular.

23.9.2 Concerning Regimes of Physical Causality

Since Galileo, Leibniz, and Newton, we have given ourselves mathematical tools for physical thought. These tools are at the heart of the construction of scientific objectivity. In fact, on one hand, they have been crafted by their relationship to the studied phenomena, and on the other, the rigorous physical concepts of force, velocity and acceleration.Their causal relationships are given (constituted) by equations, mathematical operations of limits and infinitesimal notions, since the origin of differential calculus.

In this construction, Newton and Leibniz draft out theories of the continuum. Cantor and Weierstrass provide them with foundations, very particular ones, but solid, via arithmetical approximation of limits. This continuum is thus a limit, able to be approximated in Cantor–Weierstrass style; its continuous transformations (differentiable) are also able to be approximated, using Fourier series, for example. Indeed, according to Laplace, and except for some "critical cases" he was aware of, a good approximation of the initial conditions of a physical system should have determined, in general, a good approximation of the evolution of the system (of its transformations). Even if we could not have "seen" and proposed these new operations of limits and these infinitesimal variations without mathematics of the continuum, the arithmetical discretization within a Newtonian–Laplacian framework suffices to describe them, by successive approximations. Arithmetic thus allows, on one hand to escape the foundational stalemate of new geometries (through Frege-Peano-Hilbert styled logico-formal systems and their arithmetical coding), and, at the same time, to found, a posteriori, these operations of limits, as mysterious as intrinsic to the infinitesimal calculus of Newton and Leibniz, by the most important mathematical theories of the continuum, those of Cantor and Dedekind. The approximation from *exact* measurements, such as integers, and *absolute* ("arithmetical magnitude is an absolute", writes Frege), founds all which is intelligible in time and space, in fact, "all which is thinkable" (Frege, 1884). And in the physics of systems which are Laplacian or sufficiently stable, the approximated solutions (in fact, digital or arithmetic), faithfully follow continuous trajectories.

Riemann and Poincaré propose two conceptual turning points, as radical as they differ: their works are at the origin of the geometricalization of physics but also possess a foundational perspective. For the first, the curvature of ether, a perfectly elastic continuum, can account for these mysterious forces acting at distance and work to unify them. The second shows the difference between the (low) equational complexity of the movements of celestial bodies and the (high) geometrical complexity of their evolutions, of their dynamics. The approximation specific to physical measurement plays a central role in his approach and disrupts the Laplacian framework, as we shall see. The two theories based themselves upon the analysis of hypotheses made about physical space, of the "access" (to time and space), of measurement (Longo, 2003a).

Let us first reflect about the geometry of the dynamic systems of Poincaré (1880–1890). Poincaré does not open the way to "chaotic indetermination" as some would make us believe. To the contrary, and as we have already observed, Poincaré is profoundly anchored to a classical conception of physics and by his works he even brings classical randomness under the "control" of determination. In fact, the modern dynamical systems broaden the concept of determination, by including within the latter, fluctuation and perturbation, even variation below the threshold of possible measurement. In short, their mathematics makes evident the way by which the evolution of a system may also depend upon these elements which were considered as minor accidents of the physico-mathematical analysis. In other words, the *causal regime*, as framework determining the evolution of a system, comes to include variation, perturbation or fluctuation, even when these are below physical measurement. Or otherwise that a specific trajectory is determined by equations, if possible, as well as by variations/perturbations/fluctuations of its point of origin or of its boundary conditions (in modern terminology – early 1970s: sensitivity to initial and/or border conditions; Turing's exponential drift – 1952!). However, this broadened (and weakened) concept of determination no longer implies Laplacian predictability; here is the turning point which provoked such noise, quite righteously, but a noise which has sometimes failed to adequately grasp this broadening of the role of classical determination (since, lottery, dice casting, etc, are considered as deterministic systems, contrarily to the purely probabilistic analysis produced by Laplacian physics). But then, the approximation, specific to physical measurement, acquires a crucial role: it participates to the construction of scientific objectivity in an essential way, by the fact of not being exact.

As for Riemann, the breakage is now relative to the Newtonian time–space absolute: one may no longer choose a system of Cartesian coordinates at will, anywhere in time and space, nor any sort of measurement, all while considering space, time, and measurement as absolutes (for Riemann, metric structure is correlated to the spatial curvature). Einstein will center his scientific construction upon the role of reference systems and of associated measurement: Weyl will explain it by means of a mathematical theory of gauge changes which allows for the understanding of the passage of Newton to Einstein as the passage from absolute subjective to relativizing objective in physics (Bailly and Longo, 2006). The relativistic determination then includes, in a way essential to the theory, the relativity of spatiotemporal measurements and frame of references; consequently, in Minkowski's relativistic spaces, for example, the causal correlations differ from those of Newtonian space. And this path, which goes from Laplace to the modern geometrization of physics, may be rediscovered within the discrete versus continuum game.

Let us summarize our thesis: a structurization of reality by a geometry of continuity or by an arithmetic discretization induces different scientific understanding of phenomena, in relation to causality, and this extends to models of Cognition and Mind. This is why the reflections on the continuum found in Leibniz, Goethe, Peirce and the Wittgenstein of the 1930s propose a very specific philosophy of knowledge, as opposed to the logico-arithmetic approaches (Fabbrichesi and Leoni, 2005). So there is the bifurcation which occurred in history. On one hand, there is the highly justified requirement to "found" mathematics upon arithmetic exactitude and absolutes, following the crisis of the relationship to physical space; this having been caused, as a matter of fact, by the invention of Riemannian varieties, deprived of sensitive intuition. On the other, there is the geometrization of physics, centered upon the role of approximation (of measurement, in the geometry of dynamical systems) and relativization (of the reference frame and of the curvature of space, in Riemannian and relativistic approaches).

Now, the first branch of this bifurcation produced, among other things, formidable arithmetical machines; the second of two great theories of modern physics. In short, the first followed the following conceptual pathway: "there exists absolute laws of thought, independent of man and of any material implementation" (Boole, Frege), "arithmetic induction is one of the pillars" (Frege), "we transcribe these laws into finite sequences of signs without signification, manageable in a potentially mechanizable way; within this framework, the notion of proof is decidable, indeed the theorems are decidable, by machine" (Hilbert), "a logico-arithmetic machine which separates software from material and goes 'left, right, 0,1', can calculate/demonstrate anything" (Turing); "all the logico-formal systems for deduction/calculus are equivalent: we thus have an absolute, Church–Turing styled calculability" (Gödel, Kleene, Church, Turing). So there we have the onset of logico-arithmetic machines, the most extraordinary tool man has created for himself and which is transforming the world.

These machines are specifically arithmetic in the sense that the "measurement", as access to the database used by the calculations (which constitutes the evolution of the system), is exact and absolute. All the theory of computability, of coding, of databases presupposes that the latter be accessible in an exact and absolute manner (a bit will be 0 or 1, and one or the other, exactly; moreover, its value is not relative to the measurement or the access protocol). Of course, one may complicate things ad hoc and for good reasons, and there exists interesting cases where these two aspects are put into question, and we shall come back to this in the responses to the commentaries, but here we shall talk about what is intrinsic to the theory. And that is also at the heart of applications. When you download a web page from Australia, you want the access to the distant data-base to be exact, not a comma should be lacking. If the accents are poorly reproduced, you become furious. Moreover, the access must be absolute, meaning that it must not depend upon the transfer protocol used (http, ftp, etc.), nor upon the path (passing through nodes in Japan or in Iceland? We do not even care to know.), and repeatable at will, identically, just like the software which you buy. Finally, any software must be independent of the material implementation; there is Frege's dream finally made true ("Pythagoras' theorem does not depend upon the phosphorus in the brain", his famous remark). However, networks and distributed systems (in time and space) surely pose problems, as for the absolute access to shared memories and the iteration of processes. In theories of concurrent and distributed systems, people struggle in order to obtain exactness/absoluteness/iterability. The existing problems are logical (the complexity of the network), but they mainly derive from the "friction" between (the network of) discrete state machines' internal causal regime and an environment which we more easily grasp by continuous spatiotemporal dynamics. As a matter of fact, with distributed systems, physical space and time stepped into computations.

For the moment, let us put aside the computational hypothesis which would give these arithmetic machines a role of modelization, even maybe descriptive of the essence, of many natural phenomena, including cognitive ones. And let us only consider their computational talents. Then, the algorithms and the framework of arithmetic certitude may very well account, at least indirectly and approximately, for any physical dynamic, even biological (or cognitive, even within the cognitive dynamic hypothesis): they would constitute digital models, thus approximated, of any modelization by mathematics of the continuum. In fact, even the tenants of the dynamic hypothesis in Cognitive Sciences believe so.[35] As it happens, a first glance upon numerical methods normally leads one to think that, given a dynamic system, any continuous trajectory (evolution) can be approximated by a trajectory (evolution) generated by a DSM. For example, a sequence generated by the logistic function (see the article), in the continuous realm, would always be approximated by its digital variant (with its round-offs, at each step). However, in general, it is none of that: the round-offs, inevitable to the arithmetic machine, have the same effect as "small perturbations" and, because of the instability of the system, they suffice to generate enormous changes of trajectory in time. Theorems of "stability" or "shadowing lemmas" guarantee, for a few sufficiently regular dynamical systems, that the simulation be "globally good". The shadowing lemmas, for example, ensure at most that the digital approximations be approached by continuous evolutions, but the opposite is not necessarily true (in short, for the logistic sequence, it is not true that for each continuous sequence we may associate a digital approximation which "shadows" it closely, but, conversely, it is the digital sequences which can be approached, shadowed, by continuous sequences. This reversal of logical quantifiers is, quite obviously, very important and motivates the fine analysis of these results (Pilyugin, 1999). For some systems, even the weak guarantee of global approximation, given by the shadowing lemma, does not work (Sauer, 2003). Computational imitation, by adding perturbations due to the arithmetic round-off (a fact intrinsic to the digitalization process), modifies the causal organisation (and thus the trajectories), such as it is proposed by the equations of a dynamic, because the round-offs, a local perturbation, can cause an important change in trajectory, thus the global structure of the dynamics.[36] That goes along with the two other components of structural change, with regards to modern mathematical physics, relative to measurement: in its own universe, the machine has exact and absolute access to the data.

[35] See van Gelder (1998) and his commentators.

[36] Normally, in order to apply the theorems of stability or "shadowing lemmas" to the round-off, we consider the latter as a perturbation. While reading this page, Thierry Paul (quantum physics mathematician) observed that the round-off seems rather to play a role comparable to quantum measurement. We would not today analyze the latter in terms of perturbations, but rather as an interaction of differing phenomenal levels, a sort of subject/object entanglement, which changes the phenomena on two levels. This reading of the round-off reinforces the idea it contributes to the change of causal organization, in the course of a process: by destroying information, it participates in the determination of the evolution of the coupled system (computer and mathematical dynamics).

In conclusion, for these three different aspects, the machine in general does not produce a model, nor even the digital approximation of a physico-mathematical model (as an attempt to grasp some constitutive-causal principles of a phenomenon), but an imitation, with its own causal regime. And when we see this modern marvel which is the construction of an attractor on a computer screen, we must know (that which some specialists know quite well), that there lies there but an image only qualitatively similar (if the Shadowing Lemma holds!) to continuous dynamics which we claim to approximate; that, thanks to an extraordinary quantitative effort, we have but a qualitative imitation of a phenomenon (a marvelous imitation, but distinguishable: press "restart"). Modern science is in the process of shaping itself, and quite rightly so, around these tools for thought. It is thus necessary to intently reflect, also in theory of computation, upon the internal structure of these images which these imitations give us of the world.

23.9.3 Epistemological Consequences

When new paradigms for the foundation and the application of mathematics are proposed, such as with, on one hand, Frege's deep conceptual analysis and the logical quantification structure which he develops, up to Hilbert's program, or, on the other hand, the geometrical methods for physics, a sometimes implicit philosophy of nature accompanies them. It is thus not illegitimate to project upon the machine properties which made reference to thought, because this projection has been made, at the moment the machine is conceived as a machine for thought, even as model (essence?) of cognition, of the living, of the world: this passage only made explicit an implicit philosophy of nature, which follows the arithmetic paradigm for the foundations of mathematics. So there we have that the search for foundational certainties in the exact and absolute access to integers, without the fogginess of the continuum, of the curvature of space, of Poincaré's unpredictability, becomes a philosophy of knowledge; and Hilbert's "non ignorabimus" ("any mathematical statement is decidable"), originally a program internal to mathematics ("all is reductible to – encodable within – arithmetics" and "formal arithmetics is complete" or every arithmetic assertion can be decided), becomes a general scientific program which leaves us highly perplexed, 20 or 30 years after Poincaré's results on the unpredictability-undecidability inherent to deterministic systems (formally described by a finite set of equations). Yet, ultimate consequences of the philosophy of nature specific to the formalist approach, concerning arithmetical machines which are the product of the strong and important program, many shall speak of a faithful or complete image, a model of the world. Besides, mathematics, including arithmetic, is within the world, because we lay them within the world, because they structure it. Particularly, when we project upon the world this exact and absolute reading grid, of which Hilbert's "non ignorabimus" is the epistemological consequence, we propose another regime of causality, even another structure of intelligibility, in comparison to those of the mathematical physics of time–space continua.

As a matter of fact, the approximation, which implies unpredictability, and the relativization, which prevents the absolute, participate in physical determination and they are excluded from the conceptual framework of absolute arithmetic certitudes.

One must not however forget that arithmetic machines are today essential to science. In fact, computers are at the heart of any scientific construction. They are even involved in any physical measurement. They abet mathematical proof, by developing calculations and formal deductions which were previously impossible (the theorem of four colors gave the machine huge calculation tasks; they replace us in the formal fragments of proof; even Lorenz's attractor is the product of a machine, an extraordinary image – a qualitative imitation of a dynamic process). Now that these machines are indispensable, to the point that they are becoming constitutive (and I use this strong word) to the construction of scientific knowledge, that they give wings to thought, by their immense and exact databases, by rapidity, reliability, and perfect iteration, it is necessary to escape the myths of a philosophy of nature (implicit and) erroneous even 100 years ago. More so, this is necessary because our tools for action upon the world are at the center of our humanity. This latter begins with the first stone tool. It continues by the construction of, what would I know, the wheel, clocks, etc., of the vapor engine. Surely the first man who constructed the wheel said, in all his creative enthusiasm, "my movement, indeed movement is there: the wheel is complete, it can take us anywhere". But not so, the moment a big rock appears on the road it ceases to function. That does not mean the wheel is not an extraordinary invention. Similarly with Vaucanson and other 18th-century inventors of marvellous gear systems might say, "here we are, they are a complete model of man, in fact they are the essence itself of our biological being, an automated statue." But the real application appears when we cease to build mechanical ducks and dancers (too bad, because they are so pretty), we no longer speak of the modelization/essence of the living and we build the machines of the great modern industry, with these same gearings, powered by vapor.

23.9.4 Synthesis and Conclusion

To summarize, the discrete structures (or discretization) specific to Arithmetic proposes an access grid to phenomena where we lose the meaning of continuous approximation, of fluctuation, of contiguity (Aristotle had already observed that Euclid's continuity served to describe contiguity), of correlation (discrete topology does not allow to describe the physical correlation as neighbourhood-causal proximity), of the dependency of the reference system. In fact, they base themselves upon measurement as exact and absolute, thanks to round-off, sorts of perturbations which participate as the causal regime, but which have nothing to do, a priori, with those which we grasp in the world of the physical or the living. Moreover, in what concerns exactitude and absoluteness:

- In physics of dynamical systems, the fact that the measurement is not exact is at the heart of the theory.
- In relativistic physics, the measurement is not absolute (and this is also inherent to the theory).

When these exact and absolute aspects of measurement, as access to the digital database, intervene in discrete simulation/modelization, the loss of structure becomes crucial, because it modifies the causal regime: the latter no longer integrates the variation below neither measurement nor relativization. The stakes are enormous, because outside of linear or sufficiently stable systems, the approximation and the choice of the reference system do not need to be invariants of the process (up to gauge transformations, in what concerns the reference systems). And the passage from mathematics of the continuum to that of the discrete, particularly when implemented in a computer, then imposes a different causal organization.

We have left aside Quantum Physics. In this case, measurement is neither exact nor absolute, but for completely different reasons, as inexactness is due to the intrinsic role of probabilities in measurement (there is a *set* of exact *possibilities* as for the next state of affairs). Moreover, the measurements depend on the instrument itself and upon the order by which they are taken. But the analysis does not make itself under the classical or relativistic terms used earlier: the discrete and the continuum superpose themselves, just as the photon does when presenting itself as continuous wave or discrete particle, according to the measuring instrument employed. And this discreteness is not local, it is nonseparable. It has nothing to do with the discrete topology of arithmetized structures, the well separated and localized boxes of digital data. For this reason, in quantum physics one strongly feels the need for a new theory of the continuum of which the construction is not necessarily grounded on points (coverings of the torus, strings, fractals of scale theories, etc., based on Synthetic Infinitesimal Analysis?). And the debate takes on a different bearing.

Acknowledgments A question from Jean Petitot and his interest have encouraged me to clarify a brief reflection in writing. But this article could not have been possible without the many discussions with Francis Bailly, without his teaching of the "sense of physics" and of the physical singularity of life. Jean-Yves Girard, in the introduction to the text by Turing, as usual, forces us to think (and to appreciate the "secret darkness of milk" or the importance of philosophical and poetical sensitivity in science). Sidney Frankel suggested the observation on recurrent systems in the conclusion. Jean Lassègue enriched the text by numerous critical comments and by references. Cédric Paternotte raised some interesting questions.

References

Preliminary or revised versions of Longo's articles, including the introduction to (Aceto et al., 2003), are downloadable from: http://www.di.ens.fr/users/longo or Search: Google: Giuseppe Longo.

Aceto, L., Longo, G., and Victor, B., eds., 2003 The difference between sequential and concurrent computations, Special issue, *Mathematical Structures in Computer Science*, Cambridge University Press, **13**(4–5).

Adler, R. L., 1979, Topological entropy and equivalence of dynamical systems, *American Mathematical Society*.

Alligood, K., Sauer, T., and Yorke, J., 2000, *Chaos: An Introduction to Dynamical Systems*, Springer, New York.

Amadio, R. and Curien, P. L., 1998, Domains and lambda-calculi, *Birkhuaser*.

Amsterdamski, S., ed., 1990, La Querelle du Déterminisme, Gallimard, Paris.

Asperti, A. and Longo, G., 1991, *Categories, Types and Structures*, MIT Press.

Bailly, F. and Longo, G., 2003, Space, time and cognition: from the standpoint of mathematics and natural sciences, in: *Causality and Mind*, Peruzzi, ed., Kluwer.

Bailly, F. and Longo, G., 2003a, Incomplétude et incertitude en mathématiques et en physique, *dans Il pensiero filosofico di Giulio Preti*, Parrini, Scarantino, eds., Guerrini ed associati, Milano, (english partial version downloadable).

Bailly, F. and Longo, G., 2003b, Objective and epistemic complexity in biology, Invited lecture, in: *Proceedings of the International Conference on Theoretical Neurobiology*, N. C. Singh ed., NBCR, New Delhi.

Bailly, F. and Longo, G., 2004b, Causalités et symétries dans les sciences de la nature: Le continu et le discret mathématiques, *dans Logique et Intéraction: Pour Une Géométrie de la Cognition*, Joinet, ed., Presses de la Sorbonne.

Bailly, F. and Longo, G., 2006, Mathématiques et sciences de la nature, *La Singularité Physique du Vivant*, à paraître, Hermann, Paris.

Barendregt, H., 1984, *The Lambda-Calculus: Its Syntax, Its Semantics*, North-Holland, Amsterdam, The Netherlands, rev. edn.

Berselli, L. C., Liescu, T., and Layton, W. J., 2005, *Mathematics of Large Eddy Simulation of Turbulent Flows*, Springer.

Cannone, M., 2003, Harmonic analysis tools for solving Navier-Stokes equations, in *Handbook of Mathematical Fluid Dynamics* **3**, S. Friedlander and D. Serre, eds., Elsevier.

Church, A., 1932–1933. A set of postulates for the foundation of logic, *Annals of Match*. v. 33 pp. 348–349, & v. 34, pp. 839–864.

Church, A., 1940. A formulation for the simple theory of types. *J. Symp. Logic*., v. 5. pp. 56–58.

Copeland, B., 2002, The Church-Turing thesis, *Stanford Encyclopedia of Philosophy*, Web edition inhttp://plato.stanford.edu/entries/church-turing/#Bloopers

Dahan-Delmedico, A., Chabert, J. L., and Chemla, K., 1992, Chaos et déterminisme, Seuil.

Devaney, R. L., 1989, *An Introduction to Chaotic Dynamical Systems*, Addison Wesley, Reading, MA.

Enriques, F., 1935, Philosophie scientifique et empirisme logique, *Actes du Congrès International de Philosophie Scientifique*, Hermann, Paris.

Fabbrichesi, R. and Leoni, F., 2005, Continuità e variazione. Leibniz, Goethe, Peirce, Wittgenstein, Mimesis, Milano.

Frege, G., 1884, *The Foundations of Arithmetic* (English translation, Evanston, 1980).

Gandy, R., 1988, The Confluence of Ideas in 1936, in: *The Universal Turing Machine*, Rolf Herken, ed., Oxford University Press, Oxford, pp. 55–111.

Girard, J. Y., Lafont, Y., and Taylor, P., 1990, *Proofs and Types*, Cambridge University Press, Cambridge.

Gödel, K., Nagel, E., Newman, J., and Girard, J. -Y., 1989, Le théoreme de Gödel, Seuil.

Goubault, E., ed., 2000, Geometry in concurrency, Special issue, *Mathematical Structures in Computer Science*, Cambridge University Press, Cambridge**10**(4).

Hilbert, D., 1899, trad française par P. Rossier: Les fondements de la géométrie, 1971, Dunod, Paris.

Hindley, R., Seldin, J., 1986, Introduction to combinators and lambda-calculus, *London Mathematical Society*.

Howard, W., 1980, The formulas-as-types notion of construction (Manuscript written in 1969), in: To H.B. Curry: Essays in Combinatory Logic, Lambda-calculus and Formalism, Seldin and Hindley, eds., Academic Press, London.

Krivine, J. L., 1990, *Lambda-calcul: Types et Modèles*, Masson, Paris.

Laskar, J., 1990, The chaotic behaviour of the solar system, *Icarus* **88**: 266–291.
Lassègue J., 1998, *Turing, Les Belles Lettres*, Paris.
Lighthill, J., 1986, The recent recognized failure of predictability in Newtonian dynamics, *Proceedings of the Royal Society of London. Series A* **407**: 35–50.
Longo, G., 1988, On Church’s formal theory of functions and functionals, Invited lecture, Conference on Church’s Thesis after 50 years, Zeiss (NL), June 1986, *Annals of Pure and Applied Logic* **40**: 93–133.
Longo, G., 1999a, Mathematical intelligence, infinity and machines: beyond the Gödelitis, *Journal of Consciousness Studies, special issue on Cognition* **6**: 11–12.
Longo, G., 1999b, The mathematical continuum, from intuition to logic, in: *Naturalizing Phenomenology: Issues in Comtemporary Phenomenology and Cognitive Sciences*, J. Petitot et al., eds., Stanford University Press, Stanford.
Longo, G., 2002, On the proofs of some formally unprovable propositions and Prototype Proofs in Type Theory, Invited Lecture, Types for Proofs and Programs, Durham (GB), Dec. 2000; Lecture Notes in Computer Science, Vol. 2277, Callaghan et al., eds., pp. 160–180, Springer.
Longo, G., 2003a, Space and time in the foundations of mathematics, or some challenges in the interactions with other sciences, Invited lecture, First American Math. Soc./SMF meeting, Lyon (version française in Intellectica, n. 36–37).
Longo, G., 2003b, The Constructed Objectivity of Mathematics and the Cognitive Subject, in: *Quantum Mechanics, Mathematics, Cognition and Action*, M. Mugur-Schachter, ed., Kluwer
Longo, G., 2005, The reasonable effectiveness of Mathematics and its Cognitive roots, In: *New Interactions of Mathematics with Natural Sciences*, L. Boi, ed., *World Scientific*.
Nouvel, P., 2002, *Modèles et métaphores dans Enquête sur le concept de modèle*, P. Nouvel, ed., Presses Universitaires de France.
Petitot, J., 1990, *Note sur la querelle du déterminisme, dans La querelle du déterminisme*, Amsterdamski et al., eds., Gallimard, Paris.
Pilyugin, S. Yu., 1999, *Shadowing in Dynamical Systems*, Springer, Berlin.
Poincaré, H., 1890. Sur le probléme des trois corps et les équations de la dynamique, *Acta Math.* 13, 1–27.
Sauer, T., 2003, *Shadowing Breakdown and Large Errors in Dynamical Simulations of Physical Systems*, preprint, George Mason University.
Stewart, J., 2002, *La modélisation en biologie dans Enquête sur le concept de modèle*, P. Nouvel, ed., Presses Universitaires de France.
Turing, A., 1936, On computable numbers with an application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society* **42**: 230–265.
Turing, A., 1948, Intelligent machinery, National Physical Laboratory Report, in: *Machine Intelligence* **5**, B. Meltzer and D. Michie, eds., 1969,Edinburgh University Press.
Turing, A., 1950, Computing machines and intelligence, *Mind* **59** (Page references to its reprinted version in M. Boden, ed., Oxford University Press, 1990; traduction française et introduction dans A. Turing, J. -Y. Girard, La machine de Turing, Seuil, 1991).
Turing, A. M., 1952, The Chemical Basis of Morphogenesis, *Philo. Trans. Royal Soc., B* **237**: 37–72.
van Fraassen, B., 1994, *Lois et Symetries*, Vrin, Paris.
van Gelder, T., 1998, The dynamical hypothesis in cognitive sciences, target article and discussion, *Behavioral and Brain Sciences* **21**.
von Neumann, J. and Ulam, S., 1947, On combinations of stochastic and deterministic processes, *Bulletin of AMS* **53**: 1120.
Weyl, H., 1918, *Das Kontinuum* (trad. italiana di B. Weit, Bibliopolis, 1977).
Weyl, H., 1927, *Philosophy of Mathematics and of Natural Sciences* (english translation, Princeton University Press, 1949).
Weyl, H., 1952, *Symmetry*, Princeton University Press.
Wittgenstein, L., 1980, *Remarks on the Philosophy of Psychology*, Vol.1, Blackwell, Oxford.

Chapter 24
Going Under Cover: Passing as Human

Artificial Interest: A Step on the Road to AI

Michael L. Mauldin

Abstract This chapter discusses strategies for building a computer program to pose as a human for the Turing test, including the use of humor to distract the human judge from the task of evaluating the conversation. Experiences of computer programs from the Loebner Prize competitions are analyzed to give a "top 10" list of mistakes that computers make when trying to appear human.

Keywords Natural language, Artificial Intelligence, deception, posing, human conversational patterns, humor, Julia

Suppose a primate ethologist, endeavoring to study the intelligence of a chimpanzee, sets up an elaborate maze with puzzles, athletic challenges, tests of dexterity, manipulation, and intellectual prowess. At the end of the maze, he places a bunch of ripe bananas as a reward.

The chimp watches the experimenter setting up the experiment, and sees him going in and out a back door to bring in all the apparatus. Then, when the test begins, the chimp simply goes out the hallway, walks around the back of the room, comes in the back door, and sits down to eat the bananas.

Did the chimp pass the test?

24.1 Where is the "Back Door" in the Turing Test?

My goal in this paper is to help budding Turing System engineers. Remember that the goal of the Turing Test is for a computer to fool a human. The most typical approach is to make the computer faster with more memory, more disks, a larger vocabulary, a better parser, and so forth. Those are all wonderful advances, and by all means use them, but a successful Turing System will have to know how to pose as human, how to take advantage of any known human weaknesses the judges may have.

Carnegie Mellon University and Conversive, Inc.

R. Epstein et al. (eds.), *Parsing the Turing Test.*

Is it fair to exploit human weakness to pass an intelligence test? The goal is to prove intelligence, but the test requires the computer to deceive the judge. The task is to build a basically dishonest system. In some ways, this stacks the game against the computer, because if it succeeds, critics will correctly say that the system used tricks, because it succeeded in tricking the judge.

But given that is how the game is played, a smart computer would be foolish to ignore good advice on how to fool people. Besides, the ways that the Loebner Contest has been changing since the first test in 1991 have worked to make the current test much harder than Alan Turing's original 5 min test by average judges.

Indeed, part of the impetus for these changes was the early success of Joseph Weintraub's program entitled "Liberal or Conservative", which was ranked above one of the human confederates by five of the ten judges (Epstein, 1992). These kinds of changes are common in the field of Artificial Intelligence (AI), even if they do have the effect of moving the goal line, because AI is a moving target.

24.2 AI: A Moving Target

Fifty years ago, early computer researchers were working on tests of intelligence such as tic-tac-toe (Metze, 2001), checkers (Samuel, 1959), chess (Shannon, 1956), and backgammon (Berliner, 1980). Chess had a certain mystique, being played typically by very intelligent humans. Others were impressed when Arthur Samuels (1959) wrote a reasonably good checker playing program that was able to beat its creator. Eventually, researchers from Carnegie Mellon and IBM built a chess playing computer that beat Garry Kasparov, the human world champion, in a six-game tournament (Pandolfini, 1997).

But long before that achievement, the scientific consensus had shifted away from skill at chess being proof on intelligence. While chess ability is a sign of intelligence in humans, it is merely evidence of calculation in computers.

What happens is that initially, the way to solve a problem is unknown and mysterious. We ascribe the ability to "intelligence" and assume that that problem requires intelligence to solve. Then, some determined AI researcher studies the problem, writes a program or builds a computer to solve the problem, and voila! The problem is no longer a mystery; you can read the program and figure out how to solve the problem. The poor AI researcher is no closer to the "Holy Grail" of intelligence. Instead the problem is demoted.

I want to stress that Turing's prediction in 1950 was that by the year 2000, computers would be able to fool the average questioner for 5 min 70% of the time. Given that during an average conversation in the Loebner contest, each human question or answer takes between 30 and 60 seconds, Turing's original test would only have allowed three to five questions before demanding an answer from the judge. Weintraub's program handled twice as many questions and fooled 50% of its judges.

By 1994, the Loebner Prize had become more like the Voight-Kampff test from the movie *Blade Runner*, where the investigators looking for androids used 20–30 questions designed by experts (Dick, 1990). Each Loebner judge was given a total of 15 min in two sessions, and the judges were all professional writers, from publications such as *Life, LA Times, Reader's Digest*, and *US News & World Report*, as well as reporters from major news networks. Finally, the screen was not cleared between judges, so a judge could and sometimes did pick up on a previous question and answer by the computer, in effect lengthening the questioning period even further. That year, no computer was mistaken for a human.

How can the computer fight back? One way is by exploiting the human fondness for humorous and intriguing stories. It may be a long time before computers can outwit a Blade Runner, but current programs can already keep a reader's interest for a few minutes at a time. Rather than aiming high for the Holy Grail of AI, we can focus future efforts on building *artificially interesting* systems.

24.3 Facing the Interrogator

The most difficult aspect of writing a Turing System is the cover story. Invariably, the conversation turns from discussion of the topic at hand to probing questions about the computer itself:

– Where were you born?
– Also, on the topic of Indians, would you prefer using the term Native American?
– So you don't advocate offering cryonics to any and all customers?
– Has anyone you've known been frozen?
– So you no longer listen to them? [The Rolling Stones]
– Have you owned any cats?

These questions were taken from the 1994 Loebner Competition Transcripts, and were all asked by the human judges. Any program expecting to pass as human must be able to answer factual questions about its appearance, history, relatives, likes, dislikes, opinions, and background.

Unfortunately, true answers to any factual questions about the program's background will give it away. In the first Loebner Competition, my program "Julia" had the following exchange:

Judge: How old are you?
Julia: I am 1 year and 10 months old, Judge.
Judge: In human years?
Julia: What?

The program had to be fairly smart to do this, and it still failed. After all, it correctly understood what was being asked, it knew the day it came into existence, it knew the current date, calculated the difference and converted a useless number like

4.7 million seconds into a reasonably grammatical and responsive phrase, “1 year and 10 months”. Even humans cannot do that when they are less than 2 years old.

Later versions of the program had a “contest mode” switch that told the program not to reveal any information that would give the program away. In effect, the contest mode disabled certain useful abilities of the program to answer questions related to the virtual world it normally inhabited.

One of the first things the program has to decide is what gender to adopt. Like Mr. Data’s offspring in *Star Trek: The Next Generation*, Julia is decidedly female. I made that choice a decade ago, because Julia was designed to attract interaction from an online audience, and my experience was that both men and women preferred talking to women.

The gender choice being made, I went so far as to build a lunar cycle into Julia’s “emotional” life; for 5 days out of every 29 she is “grouchy”, as this dialogue from TinyMUD shows:

Robyn: how are you feeling?
Julia: Terrible.
Robyn: why, julia?
Julia: I have PMS today.
Robyn: julia are you a human?
Julia: Is the Pope Polish?
Robyn: yes, julia, but are you human?
… julia, are you a computer?”
Robyn: julia, are you a woman?
Julia: I am female.”
Robyn: and, julia, are you human?
Julia: Not when I’m having my period, Robyn.

Before exploring ways to make up a good background for a Turing System, let us examine Julia’s real background and evolution.

24.4 Julia’s Real Background

The idea of English-speaking computers has always fascinated me, from the choppy voice of Majel Barrett as the ship’s computer in *Star Trek* to the hauntingly calm voice of Douglas Rain as HAL in *2001: A Space Odyssey*. Then in high school, I read in *Science News* about a conversation between ELIZA and PARRY, and decided to work on “Natural Language Systems”.

In college, I wrote a simple version of ELIZA in Basic for my Commodore Pet (I had the big one, with 8,192 bytes of memory). It had limited inference capability, so from the statements “I LIKE FOOD” and “I LIKE MY FRIEND”, it could say “I HAVE HEARD THAT FOOD IS YOUR FRIEND”. I majored in computer science, but my minor was in linguistics. I figured that making a computer understand and speak English would require a deep knowledge of how people use language. My diachronic linguistics professor warned me against my plan, telling me that language was too deep for computers to understand or speak.

In graduate school, I got serious, and worked on "knowledge-based" language understanding, generation, and machine translation, the real kinds of AI skills. But I spent a lot of time on an adventure game called Rogue. Along with three friends, I helped build an automated player for this game that did something I never did myself: retrieve the Amulet of Yendor and win the game.

One project that I very much wanted to work on with Rog-O-Matic was a storyteller, a program that would take the log from a game of Rogue as played by Rog-O-Matic and output a story summarizing the key events. Something as short as:

> After finding a really good set of armor, I had no trouble with the monsters until I reached level 24, where I was nearly killed by an Umber Hulk. Luckily, I ran into a room with a Wand of Fire, and used it to dispatch the Hulk. Finding the Amulet of Yendor two levels later, I had no trouble returning to the surface.

But the real world intruded and I wound up doing my thesis on information retrieval, a more profitable and useful technology. After defending my thesis, I got hooked on another virtual world called "TinyMUD", a text-based interactive community founded by James Aspnes at Carnegie Mellon University (MUD stands for "Multi-User Dimension" if you are a funding agency, and "Multi-User Dungeon" if you are not). TinyMUD's goal was to be a simple, user-extensible version of Richard Bartle's AberMUD (Mauldin, 1994). As Richard Farmer discusses in "True Names: Habitat," one of the keys to a successful virtual community is the ability of the users of the virtual world to contribute heavily to its creation (Vinge and Frankel, 2001). TinyMUD was a huge success, and started chewing up more and more "discretionary" computer time in 1989 around the entire country.

Having built an automated player for one world, and after defending my thesis, I had time to decompress before looking for a job. I started working on Gloria, a program to explore the world of TinyMUD. Gloria's job was to run around the world building a map, and collecting the local currency, "pennies." From Gloria's original limited role, I went on to write "Julia," whose job was to live in the TinyMUD documentation room, known as the "Library."

Of course, that made Julia the Librarian, and she took on the role of being an active help facility, exploring, mapping, meeting, greeting, and explaining the world to the people who visited.

After more than a year of modifications and extensions, Julia was a well-known character in the world of TinyMUD. Her shell, which I called the "Maas-Neotek" robot code, after William Gibson's (1995) novel, *Neuromancer*, had been distributed around the Internet and used as the basis for chatterbots on many other MUDs and MOOs. She had repeatedly fooled people for short periods of time into believing she was actually human.

One reason Julia was so successful in fooling people was that in the early 1990s, chatterbots were rare, and users of text-based virtual worlds like TinyMUD simply assumed that they were talking to other humans. If the program was able to avoid major mistakes in its responses, that assumption held, and it took longer for Julia to be exposed as a computer program.

The record for being fooled, so far as we know, goes to user Barry, who tried time and again to hit on Julia, and after 13 days he was still confused as to Julia's true nature. Here are excerpts from the logs (1992):

July 15, 1992

Barry: Hello how are you
Julia: Hello, barry.
Barry: Are you a virgin
Julia: As far as you're concerned, barry, I always will be.

July 21, 1992

Barry: Are you still Holding out on me
Julia: I mean, have I convinced you I'm a computer, yet?
Barry: I guess your real…but Id still like to get you to my DEN to run tests on you
Julia: I live for your approval, barry.
Barry: I m so happy to hear that. Why don't you come over to my home

July 22, 1992

Barry: hey is Julia for real or what
Julia: Hello, barry.
Barry: she says that same stuff every time
KeyLime: It's all macros, Barry"
Barry: how about filling me in on some of these things … like how Julia can type so damn fast
KeyLime: barry, it's macros
Barry: Damn I need to do some programming. I'm an ant among lizards.

After Julia had become well known on the Internet, the Loebner Prize was announced, and we decided to adapt Julia to enter the contest. The idea behind the contest was to have annual restricted tests to spur progress, along with periodic unrestricted tests to award the prize.

Julia was not well suited for this kind of test, having been designed to converse shallowly about anything and everything, rather than having deep discussions about narrow topics. I chose the topic "Small Talk", and disabled some of the TinyMUD specific responses, like directions to various rooms in the TinyMUD world. I did miss the "How old are you" question previously discussed.

Nonetheless, Julia did succeed in being ranked as more human than one of the human confederates. Ironically, that human was thought to be a computer by many of the judges, because she "knew too much" about Shakespeare.

For subsequent contests, I spent my limited free time between Loebner contests, adding topical knowledge in an attempt to deepen the conversational level. For the second contest, Julia was programmed with a variety of knowledge about hockey, including the names and stats of every team and player in the NHL, and phonetic matching module allowing Julia to handle questions like:

Judge: Who is Yommer Yogger?
Julia: Jaromir Jagr plays for the Pittsburgh Penguins.

Even with this impressive array of knowledge and abilities, Julia did not improve her performance. The major reason was that even though the contest was held in a hockey town (Boston), none of the judges was a hockey fan, and Julia failed to connect with them.

For the third and fourth Loebner contests, I changed the topic to "Pets", and "Cats vs. Dogs", in the belief that this would be a more universal topic. I also borrowed from chess the idea of an opening book of conversational patterns. This added some coherence to the program's discussions, and allowed the program to try to "grab" the conversation and direct it into areas the program could understand.

Here is a recent conversation using these patterns:

Judge: Hello
Julia: Do you have any pets?
Judge: Yes
Julia: I have two very nice cats and a goldfish tank. What kind of pet do you have?
Judge: I own three cats. What do you do for fun
Julia: I gossip.
Judge: Me too! How've you been feeling lately
Julia: Do you think that animals can think?
Judge: Sometimes. My cat occasionally sends me signals to kill the president
Julia: I'm sorry, I'm not in the mood.
Judge: To kill the president? What, you have a headache
Julia: I don't like violence.

Although the conversations in the third and fourth contests were better, Julia still did not improve her rankings. She lacked sufficient knowledge to converse reliably about animals.

Lately, Julia has retired from the Loebner competition, and gone to work as a spokesbot for Virtual Personalities, Inc. In her new role, she does not pretend to be human, but admits that she is a computer program attempting to help people:
Verbot Log (2001-09-14)

22:36:11: Please tell me your name so I'll know who I'm talking to.
User: Alex
22:36:39: Hello, Alex. It's a pleasure to meet you. How did you find us?
User: I found you when I was searching through Yahoo.com.
22:37:04: :-O Search engines always find the best things! Q:) Are you interested in Verbot Technology?
User: Yeah, kinda.

22:37:19:	Great! Verbots are pretty cool. Q:) May I ask what company you are from?
User:	I'm a student.
22:37:31:	That's good. Which school?
User:	Charter Oak.
22:37:39:	That's an interesting school! What would you like to know about Verbot Technology?
User:	How much does it cost?
22:37:53:	For pricing information on custom applications, please use the contact information below.
User:	is there a demo version?
22:38:09:	If you would like a private demo or just more information on Verbot Technology, please contact us at: info@verbot.com
User:	Do you know geography?
22:38:37:	I don't know geography. I've mainly been provided information about Virtual Personalities. However, I can be programmed to learn about anything.
User:	Wow, that is intresting.
22:38:56:	:-O Wow is right!
User:	You are an extremely unique technology.
22:39:23:	Giving a Verbot personality is an art. That's one thing that makes us stand out.

The main focus of the company's work on Julia and other Verbots® has been to give them voices to provide output on the computer speakers and animated faces that lip synch with the voices in real time. Julia has the distinction of being one of the first programs to get a paying job as a spokesbot: she works for Acura selling the "OnStar" service for the Acura RL.

24.5 Avoid "Artificial Stupidity"

A cliche among researchers is that to build AI, all you need to do is avoid mistakes. The way to win is not to lose. With that in mind, here is a "Top 10" list of mistakes I have gathered from 4 years of competing in the Loebner contest and a decade of work with Julia on the Internet:

24.5.1 #10 Do Not Say Too Much

In 1994, the winning program was "Sex" by Dr. Thomas Whalen (Epstein, 1994). Although ranked the most human of the computers by many of the judges, judge #3 ranked it as the worst program. The most likely reason is that almost every answer given to judge #3 was between one- and three-paragraphs long. By comparison, almost all responses from actual humans were one or two lines at most.

Remember that most people do not like to type, and will be lazy and succinct.

24.5.2 #9 Do Not Contradict Yourself

Number 9 ought to be easy to avoid; after all, a program ought to be able to control what it says. But even so, a glib program can be caught in a lie. For example, in the 1991 competition, Julia's internal "dossier" listed her birthplace as Dallas, Texas. She also had machinery left over from TinyMUD that had a memory of every place she had explored. Both were great features, but together, because Julia had no instance of Dallas in her location memory, she had this exchange with the judge:

Julia: Where do you live?
Judge: Just moved to Cambridge
Julia: I was born in Dallas.
…then much later…
Judge: Have you ever been to Dallas
Julia: I've never been to Dallas.
Judge: But I though you were born there?

The more your program knows, the more questions it can handle correctly, but the greater chance that it will get caught if your information is spotty.

24.5.3 #8 Do Not Be Perfect. Even the "Most Human" Human Made Several Typos

If you read the transcripts between judges and human confederates, it is clear that the judges are very lax about spelling, capitalization, and grammar. The confederates are much better about capitalization, but they all make a few typing mistakes.

It is simple to build a filter to introduce typing mistakes into the output, and it adds a "human" touch that may help.

24.5.4 #7 Do Not Handle All Unrecognized Input the Same Way

Most conversational programs are rule based, and there is always a point in a long conversation where either no rule applies or all the applicable rules have already been used recently. Have several ways to deal with this. Dr. Whalen's "Sex" program had a conversation with judge #2 where the program answered four different questions in seven minutes with this response:

Computer: I cannot give you an answer to that.

Julia had a different technique that did not work very well, either. She would ignore the question and try to shift to a topic where she had more rules. Judge #2 commented on her evasiveness:

Judge: You're not answering my question and you keep changing the subject.

The second best computer in 1994 was "The O.J. Simpson Trial" by Thad Crews. It was very impressive at times, but one of its "last resort" rules was to ask, "Is this really about the trial?" That led to the following exchange with judge #5:

Judge: So what did he do with the knife?
Comp: Is this really about the trial???

This backup rule was applied many times, and of course, each time the question was very much about the trial.

24.5.5 #6 Do Not Be Fooled by the Judge's Typos and Spelling Errors

I have already recommended that your program make intentional spelling errors. But be careful not to be exposed by poorly handling the judge's own typos. People handle errorful input gracefully. Without some form of loose matching or spelling correction, you might make a mistake like Dr. Whalen's "Sex" program:

Judge: what's the most effective form of *cotraception*?
Sex: I can not give you an answer to that.
Judge: I meant what's the most effective form of contraception?
Sex: The most effective form of birth control is the intrauterine device

A common last-resort rule, first used by ELIZA, depends on echoing a noun phrase from the question. Julia uses this rule, too, but it can be a source of mistakes:

Judge: What is so special about Siamese cats?
Julia: What do you think so special about siamese cats is, Judge?

This is bad enough, echoing spelling mistakes is even worse. Julia made the following goof in one of her practice sessions on the Internet:

Judge: Get lsot
Julia: Get lsot yourself, Judge.

The moral is to be very careful about using rules that repeat portions of the user's input.

24.5.6 #5 Do Not Forget Your Manners

Most people remember to give their program the ability to handle basic conversational markers, "Hello/Hello", "Thank you/You're Welcome", and "Goodbye/Goodbye". But be careful to track how you use them. Robby Garner's 1994 "Classic Star Trek" program could say hello, but sometimes repeated itself:

Comp: Hello?
Judge: Hello, yourself.
Comp: Greetings

Julia knows, for example, that if you have been speaking with someone in the last 15 min, you do not need to say hello to them again. This rule was critical in the TinyMUD days when more than one Maas-Neotek robot was often in the room at the same time. Before I added that check, you could get loops like this:

Julia: Hello, Colin.
Colin: Hello, Julia.
Julia: Hello.
Gloria: Hello, Julia
Julia: Hello, Gloria
Gloria: Hello.
Colin: Hello, Gloria.

And let us not forget how they ended episodes of the Waltons:

Goodnight, John Boy.

Just remember that you need to handle basic conversation, but you need some control mechanism to avoid looping.

24.5.7 #4 Know the Basics for Your Topic

There is no quicker way for a computer to be found out than to miss a really simple question. Again, from Garner's "Classic Star Trek" program:

Judge: I know nothing about Star Trek. What are the names of the major characters?
Comp: Maybe I just forgot.

Julia was no less guilty:

Judge: Let's start at the beginning. What's cats vs. dogs about?
Julia: I don't know. What do you think cats vs. is, judge?

You have to have a certain minimum amount of knowledge about your topic, and when there is a big sign over your terminal that says "FOO", you would better be able to answer the question "What's FOO about?"

24.5.8 #3 Do Not Bring up a Topic, Concept, or Word that You Cannot Discuss

This one is deceptively difficult. If your system uses a full-blown natural language parser and generator, then you might be okay. But Julia, like most systems, is based on textual rules. Adding detail to your output makes your program seem more human, but it broadens the scope of the conversation.

For example, in trying to avoid some personal questions, Julia was programmed to say:

Julia: I guess I can tell you my kids love the cats.

But this just invited questions about kids that Julia could not handle:

Judge: You have kids. What are their names?
Julia: Their names are Spats and Bowtie. We used to have a third cat, Tuxedo, but we had to give her away.

So for every word, phrase, concept, place or thing that appears in the output side of a rule, you must make certain to have one or more rules dealing with those words in the input.

24.5.9 #2 Do Not Repeat Yourself

If you use a rule-based system for your Turing program, be sure each rule has some kind of mechanism to avoid repeated firings. The "hello loop" above is only one example. Julia was exposed as a computer to some of the judges because the 1993 and 1994 Loebner competitions allowed each judge more than one turn without the computer knowing which judge it was talking to. Inevitably during the 3 h competition, Julia repeated the same answer or question verbatim, and the judge noticed.

Judge: What is your favorite dog breed?
Julia: We had to give her away. We only had a fifth of an acre of land, and Aussies love to run.
Judge: You're giving me the exact same answers you gave me first time around. What is your favorite dog breed?

Indeed, the worst performing program in 1994, "Environmental Issues" by Jimmy Lin seemed not to have any anti-repetition built in. That cannot have helped its score:

Judge: Do you know anything about grazing?
Comp: The greenhouse effect could kill us. Look at Venus, 800 degrees all the time: typical example of the greenhouse effect.
Judge: And how is grazing related to the greenhouse effect?
Comp: The greenhouse effect could kill us. Look at Venus, 800 degrees all the time: typical example of the greenhouse effect.
Judge: Oops…

The longer and more detailed the repeated text, the worse the mistake.

And finally, the number one mistake not to make on the Turing Test (drum roll please):

24.5.10 #1 Do Not Type Too Fast

Remember back to Barry who kept trying to get some romantic interest from Julia? The only thing that gave him any doubts about Julia's humanity was her ability to type her responses almost instantaneously.

The same filter we used to simulate typing mistakes for rule #8 handles this for Julia. My program actually implements a complete physical model of typing, using statistics gathered from one of the first Loebner judges in 1991. A three-dimensional Markov model is built and used to generate a varying delay between each character (so that common sequences are typed faster than uncommon sequences of characters). Random spelling errors are introduced, and sometimes corrected, with the same frequency as the human typist. The end result is that Julia types no faster than other humans.

A side benefit of this delay is that the judges get to ask fewer questions, giving Julia fewer chances to make a fatal mistake.

24.6 Building Artificially Interesting Systems

Once we have avoided the top ten mistakes, how can we keep the human judge off-balance so he or she won't notice they're talking to a machine? Simple: keep them laughing.

The idea that computers will never understand humor is a mainstay of science fiction, but, in fact, there has been some interesting work by Roger Schank (1987) on modeling humor as an expectation failure within a knowledge-based parsing system. Kim Binsted of the University of Edinburgh has implemented a system called JAPE (1996) that can actually create punning riddles.

Of course, it is even easier to simply store a large number of jokes and humorous sayings in the computer, with rules for bringing them out at the appropriate time (remember Julia's answer that she was not human when she was having PMS).

I believe it is no coincidence that the best program at the first Loebner competition (and indeed, the best performing program I know of) was Joseph Weintraub's "Liberal or Conservative."

Judge: What would you say is a liberal position on drugs?
Comp: Do you have liberal or conservative views about the legalization of pot Oh, all this typing is getting to me! Please continue…
Judge: Being liberal or conservative implies that we can attach these labels and have some kind of self image and free will. Do you agree that we have free will?

Comp: You have free will only if you believe you have free will! Can you converse with an aardvark? Please continue…

Weintraub's program seemed so crazy that the judges stopped caring whether it was answering the questions put to it because it just kept coming up with fun things to read.

I made a concerted effort to include humor in Julia's repertoire. Because her background was Internet chatting, Julia has lots of rules to deal with amorous typists and would be suitors.

Bob: My place or yours, Julia?
Julia: Both. You go to your place and I'll go to mine.

In reading thousands of lines of logs from Julia's Internet conversations, you can see a pattern:

1. People meeting Julia for the first time assume she is human.
2. Julia makes a tiny mistake, and they start to wonder if she is a computer.
3. They ask her if she is human, and Julia gives them a humorous response (or at least an attempt at humor, called a "zinger" in Julia's programming).
4. They are mollified for a while, and continue to treat Julia as a human.
5. Finally, Julia makes a horrible gaffe, and exposes her true computer nature.

The effect of the humor, then, is to buy Julia a little time. In the context of the Loebner competition, the goal is to get to the end of the 15 min without committing the major mistake that gives her away totally.

Of course, there is more to being interesting than humor. People tell all kinds of stories in addition to comedy: drama, action, horror, and so forth. I do believe that of all the kinds of stories told, computers are assumed to be least capable of humor, so it is the most effective way for the computer to pose as human.

24.7 Suggested Future Work

As previously discussed, part of the problem is that the computer needs a background that is plausibly human, and the computer must be able to answer questions about itself:

Where were you born?
How old are you?
Are you a republican?
Do you like hot dogs?

I would like to see a two-pronged approach to building a program with an "interesting" cover story.

For the first part, the computer needs to generate a plausible world from which to discuss itself, its goal, likes, interests and so forth. I have been fascinated recently by the computer game *The Sims*, created by Maxis' Will Wright. They have built a detailed world model populated by dozens of computer agents, each with interests and abilities. These "sims" interact with each other using a simple emotional model that is plausibly based on real human interaction.

Using such modeling software as a "last resort" rule, the program could handle some questions by generating new memories based on modeled events that would still be consistent with memories already generated and voiced by the program. This ability to model could address mistake #9 (do not contradict yourself), and #7 (do not handle unrecognized input the same way all the time).

But no matter how detailed the model, you would need some kind of interest filter to guide the program's output, to avoid the "Guess What?" syndrome that children, and some adults, display.

24.8 Guess What?

Telling a good story is a learned skill. All parents can give you examples of their children attempting to tell a story and failing utterly to get to the point. Here is a real, recent example from my own family:

> Mommy?
> Guess what??
> Today when we were waiting at the table,
> you know we have to wait at the table before we're allowed back in the class,
> this one boy that looked like Willie, but taller,
> with blond hair, and his name's not Willie.
> He was looking at our table,
> and it looked like he was looking at me,
> and the boy that was with him,
> said, "Hey he's got the hots for you."

Of course, after a few more years of learning editorial skills in English class, this same child would be able to say:

Mommy! A boy at school told me another boy there likes me!

Generating this first story requires a detailed model of the underlying situation. Generating the second story requires all the data from the first story, plus a model of what is and is not relevant to the listener and speaker.

Some initial research into story telling was done at Yale in the 1970s by Roger Schank and his students (Schank et al., 1975). The method is to model the listener's expectations and convey only those parts of the story that the listener cannot infer from known facts and the story so far (Schank et al., 1975).

Carbonell's Ph.D. thesis called "Politics" could even generate interpretations of world events based on an ideological perspective by modeling the goals of either liberal or conservative politicians (Carbonell, 1979).

Following through with these ideas could produce what I had wanted for Rog-O-Matic (Mauldin et al., 1984): a summarizer that can take a log of a world simulation and output a succinct description of the key events that would hold the reader's interest (Mauldin et al., 1984).

By marrying a complex world simulator with an interest filter, I would hope to produce a computer program with a "rich inner life" that could discuss it with a human being.

24.9 Summary

I remain one of the few researches who believe that Turing's prediction in 1950 about a computer passing his "imitation game" test actually came true, or would have if the conditions of the Loebner competition had not had the effect of moving the goal line farther away. But I am not alone in my belief that building better AI systems to pass the more modern form of the Turing Test is an important goal.

Any computer program that attempts to fool human judges will have to know how to take advantage of people's biases, in particular that computers cannot behave emotionally or with humor, and therefore these are important qualities to add to conversational programs.

I discussed the "top 10" mistakes not to make when building a Turing system, based on a decade of work in building Julia, a chatterbot and Internet explorer from the early 1990s:

#10 Do not say too much.
#9 Do not contradict yourself.
#8 Do not be perfect. Even the "most human" human made several typos.
#7 Do not handle all unrecognized input the same way.
#6 Do not be fooled by the judge's typos and spelling errors.
#5 Do not forget your manners.
#4 Know the basics for your topic.
#3 Do not bring up a topic, concept or word that you cannot discuss.
#2 Do not repeat yourself.
#1 Do not type too fast.

I also propose to build an "artificially interesting" system by incorporating a world simulation and interest filter that might be simpler than building a "full-blown" AI system.

The real problem with building a "truly" intelligent computer is not that it might be impossible, but that it might be difficult to direct it towards the Turing Test.

In Vernor Vinge's classic science fiction novella *True Names*, a personality simulator in the form of a dragon guards the entrance to the secret meeting place of the Coven. At the beginning of the story, the dragon wears a t-shirt with the words "Alan Turing" blazoned across the front. As the simulator becomes more intelligent towards the end of the story, he discards the t-shirt as being "undignified".

In our case, we must ask ourselves, why would a truly intelligent program want to undergo the Turing Test, anyway?

References

Berliner, H. J., September 1980, Backgammon computer program beats world champion, *Artificial Intelligence* **13**(2): 205–220.

Binsted, K., 1996, *Machine Humour: An Implemented Model of Puns*, Ph.D. thesis, University of Edinburgh.

Carbonell, J., 1979, *Subjective Understanding: Computer Models of Belief Systems*, Technical Report YALE/DCS/tr150, Computer Science Department, Yale University, New Haven, CT..

Dick, P. K., 1990, *Blade Runner*, Balantine Paperback.

Epstein, R., 1992, The quest for the thinking computer, *AAAI Magazine* **13**(2): 80–95.

Epstein, R. 1994, *1994 Loebner Prize Competition in Artificial Intelligence: Official Transcripts and Results*, Technical Report, Cambridge Center for Behavioral Studies.

Gibson, W., 1995, *Neuromancer*, Ace Books, Reissued.

Mauldin, M., Jacobson, G., Appel, A., and Hamey, L., 1984, ROG-O-MATIC: a belligerent expert system, *Fifth Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, London, Ontario, May 16.

Mauldin, M. L., 1994, Chatterbots, tinymuds, and the Turing test: entering the Loebner prize competition, *1994 Conference of the American Association for Artificial Intelligence*.

Metze, G., Winter, 2001; http://www.cs.uiuc.edu/news/alumni/jan01/tic.html.

Pandolfini, B., 1997, *Kasparov and Deep Blue: The Historic Chess Match Between Man and Machine*, Fireside Press.

Samuels, A. L., 1959, Some studies in machine learning using the game of checkers, *IBM Journal of Research and Development* **3**(3): 210–229; reprinted 1963, in: *Computers and Thought*, E. A. Feigenbaum and J. Feldman, eds., MIT Press.

Schank, R., 1987, Carnegie Mellon Lecture series, from the author's notes.

Schank, R. C., Goldman, N. M., Rieger III, C. J., and Riesbeck, C., 1975, Inference and paraphrase by computer, *JACM* **22**(3): 309–328.

Shannon, C., 1950, A chess-playing machine, *Scientific American* **182**(2): 48–51; reprinted 1956, in: *The World of Mathematics*, Vol. 4, J. R. Newman, ed., Simon & Schuster, NY, pp. 2124–2133.

Vinge, V. and Frankel, J., eds., 2001, *True Names: And the Opening of the Cyberspace Frontier*, Tor Books, New York.

Chapter 25
How not to Imitate a Human Being

An Essay on Passing the Turing Test

Luke Pellen

Abstract Will a computer ever be able to convince someone that it is human and so pass the Turing Test? Programs that attempt to directly model high-level psychological processes are too rigid and lack flexibility and power. Current programs that attempt to pass the Turing Test work primarily by extracting keywords from user input and regurgitating preprogrammed responses. These programs can hardly be called "intelligent". A much lower-level approach is required in which the goal is not to emulate a human, but to emulate intelligence. Such an approach would attempt to model low-level biological components of the brain using techniques such as feedback, recursion, artificial life, and genetic algorithms.

Keywords Turing, Artificial Intelligence (AI)

25.1 The Turing Test

The idea behind Turing's (1950) Imitation Game is to side step all philosophical issues and accept a behavioral definition of intelligence. The argument is that there would be no reason to deny intelligence to a machine that could flawlessly imitate a human's conversation.

25.1.1 Introducing Trevor

Trevor was accepted as a finalist for the 2001 International Loebner Prize (the first implementation of the Turing Test, held annually since 1991) and scored in the top four programs. When I designed Trevor I wanted not just to design a

---

Freelance systems analyst/programmer/graphic designer

language processing program, but a character. With this in mind I programmed him with details concerning his family life, his occupation, his hobbies and a history. Trevor also had a "plot line" – a set of statements which he would reveal during the conversation in an attempt to guide it into areas that Trevor was knowledgeable.

Trevor worked by using a synonym look-up table. He would scan the input and break it down into words he understood and rejected all other words, thus producing a simplified sentence for processing. For example, a user typing in "Hey! G'day Trevor! How's things?" would have the input reduced to the statement "Greetings Trevor". Thus, there are many sentences which a user could input which may equate to "Greetings Trevor".

Trevor would also scan the input for punctuation marks to decide whether the user was making a statement, an assertion or asking a question. If the user asked a question Trevor would scan the input to indicate the type of question: "who", "what", "when", "where", "how", etc. He would also keep track of the general thread of the conversation (e.g., hobbies, work, family, etc.) to understand the context of statements.

Could Trevor be described as "intelligent"? Perhaps. Could Trevor be described as "conscious"? I seriously doubt it. So the question is what contribution does Trevor, and programs like him, make to the AI community? Will a program like Trevor ever be able to pass the Turing Test convincingly? Will *any* program ever be able to pass the Turing Test convincingly?

25.2 Approaches to Artificial Intelligence

25.2.1 The Mind and the Brain

If we choose to believe modern science, then the mind can be described as an emergent property of the brain, that is, mental phenomena are caused by neurophysiological processes in the brain.

But what role does these neurophysiological components – neurons, synapses, glial cells, neurotransmitters, etc. – play in producing mental phenomena? What *is* consciousness? What are the causal relationships between physical phenomena and mental phenomena?

25.2.2 The Philosophy of Artificial Intelligence

Mental states can be attributed to the logical functioning of *any* computational device. This is the claim of what is referred to as *strong AI*. The claim is that mental activity is simply the carrying out of some well-defined *algorithm* – a calculational procedure – even though such an algorithm may be immensely complicated.

An algorithm functions independently of the hardware on which it is implemented. One of the tenets of strong AI is that the mind is to the brain as the program is to the hardware.

25.2.3 Top-down Versus Bottom-up

Much AI research has hinged around a brute-force, top-down approach. Selected areas of cognition (such as visual perception, language interpretation and construction, etc.) have been singled out, and attempts have been made to replicate them through sophisticated hardware and software.

The programs, which are of practical and research value, can never hope to simulate the way the human brain works. This is because such an approach does not attempt to replicate some lower-level function of the brain – such as the neuron – and so lacks the flexibility, power and subtlety of the brain. A top-down approach favors the implementation of a specific high-level function rather than allowing the interaction of many primitive low-level functions from which high-level phenomena can emerge.

The epiphenomenal nature of the mind means that it can never be simulated directly in the form of an algorithm without regard to its underlying hardware; it is like trying to simulate "wetness" without regard to the physical properties of water.

Our mind is *not* an algorithm. This is obvious when comparing the relationship between the brain and the mind to the relationship between hardware and a program (algorithm). The mind emerges from the causal properties of the brain. A program, however, *does not* dynamically emerge from the hardware that supports it. Programs are *installed*, physically imprinted onto the hardware, and direct the hardware to perform low-level operations.

Instead of taking some high-level function of the brain and attempting to duplicate it through the implementation of inferred rules (top-down approach) we need to focus on the components of the brain itself and how these components interact (bottom-up approach). The isomorphism (mapping) is not between algorithm and mind (high level), but between algorithm and the laws of physics, chemistry, biology and causality (low level). Such reductionist strategies are now commonplace in the form of ANNs (artificial neural networks) and PDP (parallel distributed processing). These strategies come under the school of thought called "connectionism".

It seems that the building block of the brain is the neuron, and so it is the behavior of the neuron that requires replication. A reductionist strategy focuses on simulating these low-level neurons and connecting them in such a way that mirrors the organization of the brain. This vast collection of relatively simple components will interact with one another to produce statistically emergent phenomenon. Figure 25.1 shows the two approaches to AI.

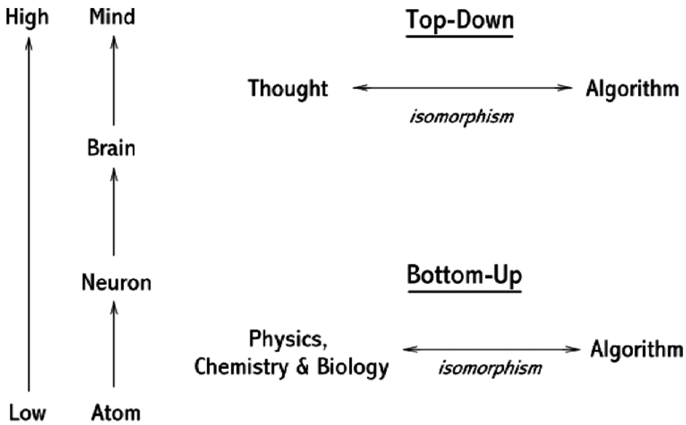

Fig. 25.1 The two basic approaches to AI: Top-down (emphasis on the psychological processes of thought) versus bottom-up (emphasis on the biological processes of thought)

25.2.4 Connectionism and Artificial Neural Networks

25.2.4.1 How Does an ANN (Artificial Neural Network) Work?

An ANN is an implementation of "connectionist" architecture. A connectionist architecture seeks to loosely emulate the workings of a biological brain and has a large number of very simple neuron-like processing elements, a large number of weighted connections between these elements, parallel distributed control, and an emphasis on learning internal representations automatically. The weights connecting each neuron encode the knowledge of the network.

A simulated neuron is a node connected to other nodes via links that approximate to axon–synapse–dendrite connections. Each link is associated with a weight. The connecting weight, multiplied by the neuron's output, determines the nature and strength of one node's influence on another: a large positive weight corresponds to strong excitation, and a small negative weight corresponds to weak inhibition.

25.2.4.2 The Realm of Chess

The game of chess provides a well-defined problem space. Traditional brute-force chess engines (such as the famous "Deep Blue") simply use raw power to search as much of the problem space as possible, using material calculations (adding up the number and value of the pieces which each player has).

When I discovered that virtually no reference material existed regarding neural networks and chess, I decided to investigate the idea myself out of personal interest. I wanted to know if an ANN could be capable of sound yet unusual and creative play, something that a traditional chess engine lacks.

25.2.4.3 Introducing Octavius

And so was born "Octavius", an ANN chess engine.

My first goal was to see whether a neural network could play anything even remotely resembling "sensible" chess. Although Octavius can perform "blunders" (bad moves), I would say that this first goal has been achieved.

I designed Octavius to be extremely customizable and open for experimentation. There are many possible configurations and many possible training approaches. In the time that I have made Octavius available for public download, I have received interest and encouraging feedback from the University of California, Santa Cruz, Karpov Chess Institute, Germany and Sao Paulo University, Brazil.

Octavius is classified as a feed-forward network: a board image generation algorithm converts a chess position into a sequence of rational numbered inputs (within the range of −1 to 1, or 0 to 1), which are fed into the neural network. These inputs are then propagated forward through the neural network resulting in a specific numerical evaluation. It is then simply a matter of choosing the move which results in the best positional evaluation.

25.2.4.4 How Does Octavius Learn?

A human who learns how to play chess begins with the basics: the movement of the pieces, the relative value of the pieces, threats, traps and discovered checks, etc.

Octavius attempts to gain an understanding of chess through a positional analysis of master and grandmaster games. His training is based on the assumption that any position reached during such games must be positionally superior to any alternatively available position during that game.

It must be stressed that Octavius has absolutely no hard-coded knowledge of chess. The theory is that through exposure to master and grandmaster games Octavius will be able to deduce the rules and tactics of chess heuristically via positional interpolation (Fig. 25.2).

Octavius can also perform a traditional minimax tree search for material evaluation to be used in conjunction with the neural network to strengthen play. For those who are interested, I have included the very first game of chess I played against Octavius after he had been trained by analyzing only ten human grandmaster games. It must be stressed that the following game was played using the neural network *only* (i.e., with no material evaluation):

White: Luke Pellen
Black: Octavius

The 28th of February 1999

1. e4 d5
2. exd5 e5
3. Nc3 Nf6
4. d3 e4

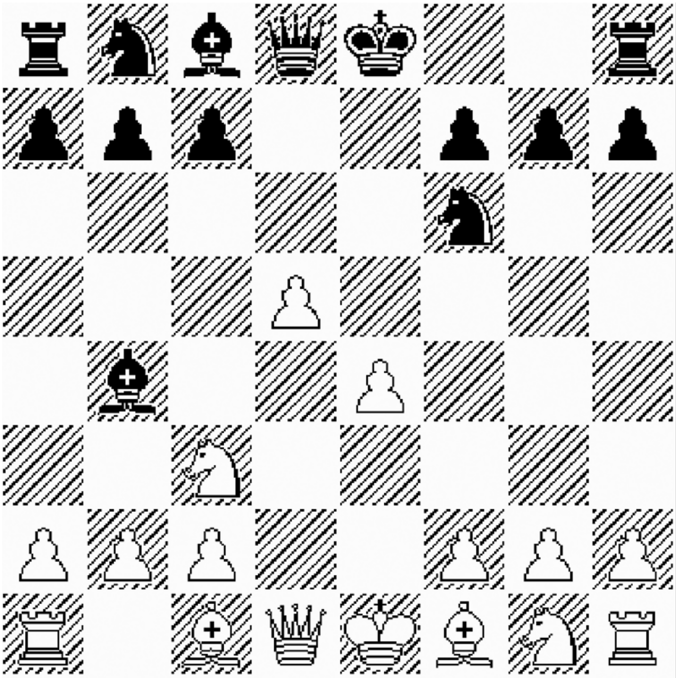

Fig. 25.2 5 … Bb4 (White to move)

5. dxe4 Bb4
6. Bd2 Bxc3
7. Bxc3 Nxe4
8. Nf3 Nf6
9. Be2 Nxd5
10. O-O Ne3
11. fxe3 Bf5
12. Nd4 c5
13. Nxf5 c4
14. Nxg7 + Ke7
15. Bh5 f5
16. Nxf5 + Ke6
17. Qg4 b5
18. Rad1 b4
19. Ng7 + Ke7
20. Rf7# {White mates}

Octavius was originally programmed on a Pentium 266 MHz with 32 Mb RAM. Needless to say, computer power has increased significantly since then. Modern computing power has meant faster training times and large neural networks.

On the 26th of April 2004, Octavius won his first game against me. Once again, this was purely through the neural network with no material analysis. Since then, he has been victorious a few times, exploiting any blunders which I make.

For those who are interested, I have included this game below:

White: Luke Pellen
Black: Octavius

The 26th of April 2004

1. e2e4 g8f6
2. b1c3 e7e5
3. f1b5 c7c6
4. b5a4 f8c5
5. g1f3 e8g8
6. e1g1 d7d6
7. d2d3 c8g4
8. a4b3 f6h5
9. a2a3 f8e8
10. b3a2 d8f6
11. c1g5 f6g6
12. b2b4 c5b6
13. g5h4 h5f4
14. g1h1 b8d7
15. g2g3 f4h3
16. d1e2 h7h5
17. a1e1 d6d5
18. e4d5 c6d5
19. c3d5 e5e4
20. d3e4 e8e4
21. e2e4 f7f5
22. e4e7 g4f3# {Black Mates}

25.2.4.5 How Does Octavius "See the World"?

An important part of using ANNs is the preparation of input. In a human being, for example, certain visual patterns such as horizontal and vertical edges are detected by cells within the retina before the information reaches the brain.

Octavius constructs a "board image" which is then fed into the input nodes (Fig. 25.3). The board image generation algorithm itself is also customizable, as it plays a crucial role. The board image dictates how Octavius perceives the world he inhabits. If Octavius has an inaccurate or flawed interpretation of his world, then his play will suffer.

Octavius sees his world in terms of the control he has over the squares on a chess board. To some degree, his perceptual system "understands" how chess pieces move and interact, although his "brain" (ANN) does not. His brain simply accepts

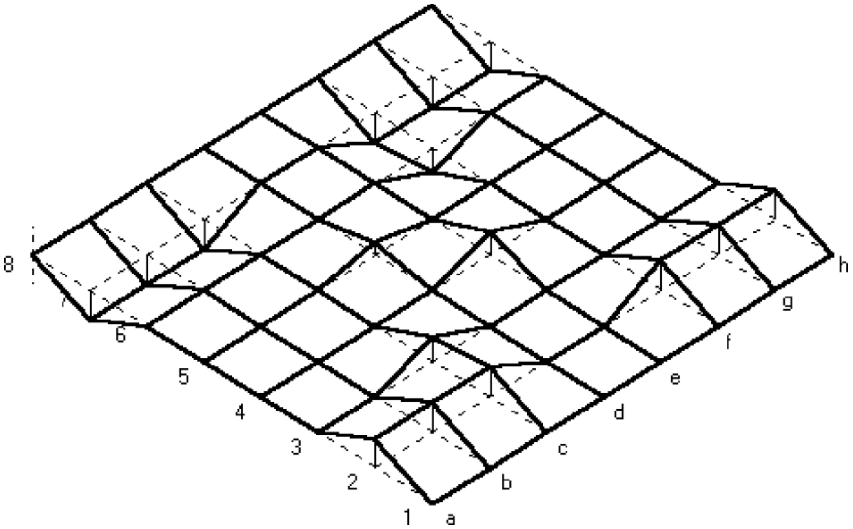

Fig. 25.3 A board image graph representing the pawn structure in Fig. 25.2. Note: The recapture power of the pawns are reflected by assigning values to squares f6 and c3. The value assigned to d5 is lower, representing a pawn which is under pressure from an enemy knight (f6) and queen (d8)

that some configurations are "good" and some are "bad" (this is done using a process called "backpropagation" where desirable evaluations are reinforced by strengthening the neural pathways).

25.2.4.6 The Future of ANNs

Everything Octavius knows about chess has been learned from human masters of the game. If there were such a thing as a "Chess Turing Test" Octavius would not necessarily be a strong opponent, but possibly a very "human" opponent.

With the power and resources of current-day technology, ANNs have become much more accessible and widespread, yet the power of ANNs are still yet to be fully realized. The possible ways of constructing an ANN are many and varied, the process of training an ANN is a long and time consuming one, and the process of data preparation is something of an art form.

There is still much to be learned regarding the best uses of ANNs.

25.3 Objections to Artificial Intelligence

The whole idea of the Turing Test is to side step the numerous and complex philosophical issues and accept an operational definition of intelligence. Nevertheless, a knowledge of the standard arguments against AI is useful to understand the scope of the problem.

Many objections to AI take the form of "a computer can only do what it is told to do". Some of these objections are quite subtle and difficult to draw definite conclusions from, but it seems that no matter how elegant or convincing the argument, it will never deter those who pursue the dream of building an intelligent machine.

I will now briefly cover these objections and their bearing upon AI research.

25.3.1 The Chinese Room

Searle (1980) puts forward the following scenario: suppose that Searle is locked in a room and given a large batch of Chinese writing. He does not understand any Chinese and the symbols appear as meaningless squiggles. All the operations for this exercise are supplied, in English, as a set of instructions for manipulating the counters with Chinese symbols on them. The sequences of symbols representing a story, and then the questions, are fed into the room through a small slot. Finally, when all the manipulations are complete, the resulting sequence is fed out through the slot, representing the answers to the questions.

This exercise is based around a program written by Roger Schank at Yale University. Put simply, the program takes in a story and then answers questions about the story in a similar way that humans would answer them. According to Searle, partisans of strong AI would claim that such a program can literally understand the story and that this understanding somehow mirrors the human ability to understand stories.

In Searle's analogy, he is manipulating Chinese symbols and so does *not* understand the story, the questions or the answers. But, by correctly carrying out a series of operations written in English, which constitute Schank's algorithm, he would be able to do as well as a Chinese person who *does* understand the story. Searle's point is that the mere carrying out of a successful algorithm does not in itself imply that any understanding has taken place; Searle in his Chinese room would not have understood a single word of the story. Thus, the claims of strong AI fail.

Searle's Chinese Room is a good abstraction of the workings of a conventional computer. Searle himself represents the central processor blindly manipulating input and output. Such an argument (as Searle admits) only demonstrates that a particular approach to AI may be incorrect, that is, the computational theory of the mind and the use of conventional computer architecture.

25.3.2 Dualism

Traditional AI holds the view that it is simply the algorithm that counts. It is supposed to make no difference whether the algorithm is executed by a brain, an electronic computer, a mechanical device or a system of water pipes. Searle (1980) points out that this is a form of *dualism*. The logical structure of the algorithm is supposed to have some kind of disembodied "existence" independent of the hardware in which it resides. This is particularly ironic as dualism is the very viewpoint with which the supporters of strong AI would least wish to be associated.

It is important to realize that a computer *can* be made of cogs or water pipes and it will still function *as a computer*, there exists no doubt on this point; hardware *is* irrelevant to the carrying out of an algorithm. The problem only arises when we equate an *algorithm* with the *mind*. As has already been established, the mind is *not* an algorithm nor can it be represented by one. The activity of the *brain*, however, could be modeled as an algorithm.

25.3.3 The Structure and Organization of a Computer

Stephen Rose (1992) states that structurally, the properties of chips, AND/OR gates and logic circuits do not at all resemble those of neurons, if indeed it is neurons that are to be regarded as the relevant units of exchange within the nervous system. The units of which the computer is composed are determinate, with a small number of inputs and outputs, and the process that they carry out are linear and error-free.

Neurons themselves behave in a predictable and regular way, they take in a series of inputs and, in an all-or-nothing fashion, they either fire or do not fire. The fact that a particular AND/OR gate does not function like a neuron does not exclude the possibility that it could be a *component* of something which functions like a neuron. Low-level determinism does not imply predictable high-level behavior. To conclude that a computer cannot possibly possess thought because its physical hardware is different to that of a biological brain is a form of "hardware chauvinism" (Hofstadter, 1985).

25.3.4 Syntax is not Intrinsic to Physics and Has No Causal Powers

Searle (1992) argues that, on the standard textbook definition, computation is defined syntactically in terms of symbol manipulation. But, syntax and symbols are not defined in terms of physics. The physics of a computer is irrelevant providing it is capable of assigning 0 s and 1 s, and of state transitions between them. Syntax, in short, is not intrinsic to physics. Therefore, computation is not *discovered* in the physics, it is *assigned* to it. Certain physical phenomena are used or programmed or interpreted syntactically. Syntax and symbols are observer relative.

It follows that you could not *discover* that the brain or anything else was intrinsically a digital computer, although you could assign a computational interpretation to it as you could to anything else.

Basically, Searle is asserting that it is the biological composition of the brain that provides it with its unique causal properties and it is not possible for a computer to possess such causal properties. A computer can merely *represent* the brain as a computational state. Of course, computers do have causal properties – or they would not function – they are just not the same causal properties as the biological brain.

A conventional computer uses a single processor to shunt binary tokens about, and to perform very simple operations on such tokens. We can *interpret* such binary manipulations as we please, say, as a word processing program in action. The program cannot exist in some platonic sense; it must be encoded into the hardware before it can be executed. Thus, there is no such physical thing as the "program" level, or indeed any software level, there is only the hardware level. This is what Searle means when he states that syntax is observer relative.

A conventional computer deals only with binary tokens stored in a vast array of static memory. These tokens are processed, in a very simple way, by a single processor. The human brain, however, deals with semantic symbols stored in a network structure. The neurons play an active role in *both* the storage *and* processing of information. Symbols in the brain are complex dynamic structures, supported by a neural substrate, which trigger, interact with and influence other symbols.

This is a slippery argument that raises many questions. Is the mind, is consciousness, necessary for intelligence? Is it possible to imagine a machine that is intelligent, yet not conscious? What does it mean to "understand"?

The essence of intelligence is pattern recognition. Consciousness is awareness of the self, the world and of the self in the world. A powerful computer capable of these things might well show us that there exists a qualitatively different kind of consciousness; undoubtedly, being constructed of silicon, it would behave quite differently than a human being. Complex systems are highly sensitive, and so a 100% accurate emulation of a biological brain implemented on a digital computer is impossible: a digital computer is capable of very high precision, but at the same time *limited* precision.

25.3.5 Gödel's Theorem

This is a mathematical objection based on Gödel's Theorem that proves there exist truths that can be expressed in any sufficiently powerful formal system that cannot be proved within that system. It follows that a computer could not recognize certain truths, because it would have to prove them in order to recognize them, thus providing a limitation for the computer.

Humans are not perfect formal systems. They may also have a limit to the truths they can recognize. Humans do not deal with formal truths, but with beliefs (mental truths). A belief is formed from and shaped by experience and does not require any sort of formal proof. A belief is not some axiomatic mathematical proposition; it is a complex neural structure.

A computer, at its lowest level, is a purely logical device. But again, this does not exclude the possibility that a computer could store a complex network structure capable of being shaped by experience – a "belief". Gödel's Theorem does not refute the possibility of AI.

25.4 Designing an Artificial Intelligence

A purely high-level approach to the Turing Test will only result in a program consisting of smoke and mirrors, using simple tricks such as regurgitating user input, adding typing errors, and the use of canned responses to certain trigger words. Ironically, it will only demonstrate the ability of human intelligence to construct order out of chaos, to create meaning from the responses given by the program.

Can we confidently describe this as "intelligence"? Could such a program *ever* pass the Turing Test convincingly? I believe not.

Language as a function cannot be isolated and emulated without regard to some sort of "experience in the world". A lower-level approach is needed: a program which is flexible and has the capacity to learn language. Only through the experience of learning language could a program generate original responses. And only through the experience of "being in the world" could a program associate the abstract (language) with the concrete (objects). Language is metaphor, and metaphor is grounded in the physical world.

In this section, I will outline some concepts for developing an AI capable of learning and experiencing, of processing language, and hence of passing the Turing Test.

25.4.1 Feedback

Human beings require mental and physical stimulation, without it, the brain fails to develop properly. A human being needs to be able to take in information about the world and be able to act upon and manipulate the world.

It is not likely that we will be able to design an impressive AI program without taking this important point into consideration. We cannot hope to design a silicon brain as a stand-alone "black box", apply power to it, and expect a glowing message to pop up on the visual display unit; "I think therefore I am."

Like us, our silicon counterpart needs sensory apparatus of a certain complexity and a physical presence so that it can interact with the complex world. Without sensation, there can be no experience of the world. Without a capacity for exploration, there can be no development.

An artificial digital environment (covered further in Section 25.4.3) may be useful as a breeding ground for the development of various specific algorithms, but such an environment ultimately lacks the richness and diversity of the real world.

25.4.2 Recursion

The ability of an algorithmic procedure or function to call upon itself is called "recursion". Recursion is a powerful and elegant programming tool. A good example of recursion is that of the factorial function. Factorial 4 is calculated, for example, as 4*3*2*1 = 24. Factorial 3 is 3*2*1 = 6. It is given that Factorial 0 = 1.

Factorial n is $n*(n-1)*(n-2)\ldots$ for a total of n terms. It can also be expressed as:

$$\text{factorial(n)} = \text{n} * \text{factorial(n}-1)$$

This definition of a factorial is recursive in that the definition of factorial makes use of factorial itself. So, the recursive function FACTORIAL, with a parameter NUMBER for which the factorial value is to be calculated, would be written in pseudocode as:

```
Function FACTORIAL(NUMBER)
If NUMBER = 0
FACTORIAL = 1
Else
FACTORIAL = NUMBER * FACTORIAL(NUMBER−1)
End-If
```

Note that recursive functions are very compact and this may be why recursive structures are also found in nature. Many structures in nature, such as clouds, trees and landscapes, can be simulated by recursive algorithms.

The tree generated in Fig. 25.4 required only seven parameters and a relatively simple recursive algorithm to generate. A large variety of different-shaped trees can be constructed by changing any of the seven parameters.

Notice also that recursively generated structures are self-similar. That is, if you take the left branch of the tree in Fig. 25.4, for example, you will see that it has the structure of the tree as a whole. Each branch of the tree reflects the structure of the tree in its entirety. Recursive structures are rich and complex, with a high degree of order and symmetry.

The compact nature of recursive algorithms means that structures of vast complexity can be stored using relatively small amounts of storage. Nature has developed recursive algorithms to store vast amounts of information blue-printing the structure of an organism within a tiny space, such as a seed.

25.4.3 Artificial Life and Genetic Algorithms

It may be necessary to employ evolutionary techniques in structuring an artificial brain. Such techniques in the field of artificial life – which attempts to simulate simple life processes, like cell growth and the emergent group behavior of ant colonies, for example – have been very successful. Using a very simple algorithm, one can grow "cells" which closely resemble the patterns found on seashells (Levy, 1992).

Every cell in the human body contains DNA, the blueprint for how to construct a human being. Genetic algorithms are digitally encoded DNA which represent

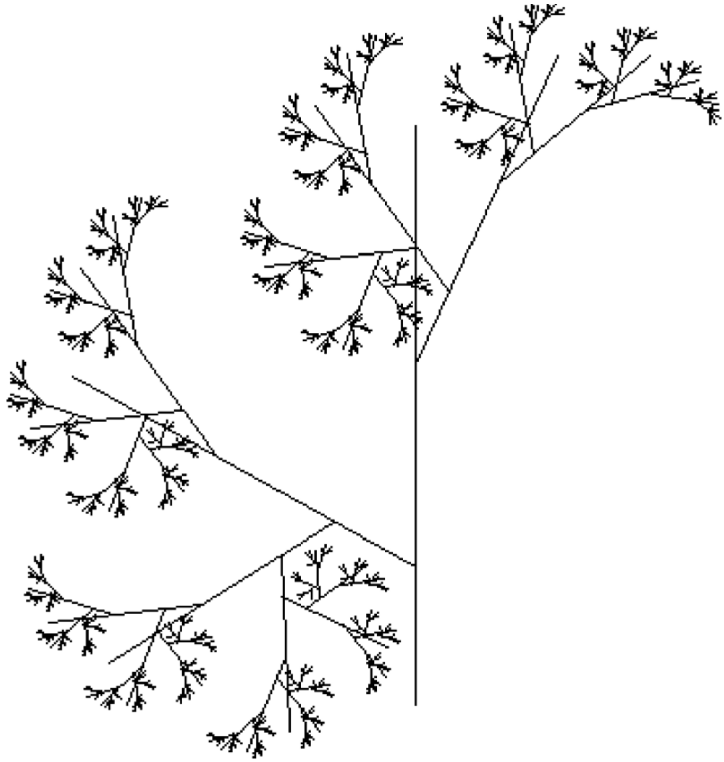

Fig. 25.4 A recursively generated tree

how to construct a virtual organism. In the real world, a process called natural selection weeds out the inferior organisms that are not suited to the environment, and encourages those organisms that are suited to the environment to propagate.

Genetic algorithms create virtual organisms that inhabit a virtual world, these virtual organisms also undergo natural (or, under the circumstances, *artificial*) selection. These virtual organisms reproduce and pass on part of their genetic makeup: processes like genetic crossover and mutation play a role in producing diverse offspring. The focus here is on providing an *environment* that encourages the evolution of organisms which fulfill specific requirements.

This approach has already been successfully adopted to "breed" a robot capable of visual recognition of simple objects (Davidson, 1995).

25.5 Conclusion

ANNs and other connectionist models are flexible and powerful pattern-recognition devices that seek to emulate the workings of the brain. However, to create an ANN as complex as the human brain is, at present, an unrealistic goal.

I suggest that an object-oriented approach is needed to organize smaller subsystems into cooperating units that form larger systems, to keep the task a manageable one. Decentralized control allows these multiple systems to work independently so that they can be interchanged or replaced by more advanced systems as they are developed.

Genetic algorithms could be used as a blueprint to recursively generate the structure of a large and highly complex ANN. These ANNs could then be tested, assessed and valued for “breeding” purposes. Those ANNs deemed most successful will be “cross-bred” with other successful ANNs.

Evolutionary techniques like this could be used to generate and select useful perceptual systems, for example, a visual subsystem capable of extracting information such as contrast, shapes or edges from raw data. The output of this perceptual system could then be passed as input to another higher-level perceptual system, say, one that takes collective input from all perceptual subsystems.

These subsystems, being modeled through ANNs, are capable of learning. Depending on the level of control of these systems, researchers will need to be able to reinforce or dampen particular neural pathways that lead to a particular kind of response. In basic terms, we need a “reward” and a “punish” button to encourage or discourage particular responses. The act of reinforcing and dampening these pathways must be a possible response of the ANN itself. This provides a basic kind of developmental growth: the ANN will organize itself by reinforcing or dampening certain aspects of its own behavior.

Language has developed as a survival trait in humans, and language itself is based on the ability to communicate shared experiences and ideas. Language is a social phenomenon, so we need to create a social entity. Perhaps a shift of focus onto the *conditions* required for language rather than the *cognitive structure* of language is required. Once perceptual and motor systems are developed we need to provide an environment that rewards and encourages the use of language. The environment itself would initially be very simple and become more complex as required.

In the early stages, such environments could be digitally created, to apply evolutionary techniques at high speeds, as it will be a time consuming process. Such digital landscapes may be populated with many virtual organisms that benefit from developing and sharing a form of language: to avoid “dangers”, or to share useful information, like the location of “food”.

This is a general outline of the approaches I believe to be successful in constructing an AI that will, ultimately, pass the Turing Test. The specifics are complicated and will require a great deal of further research, but the current trend of building simple language parsing programs are nearing a dead end when it comes to improving their performance in the Turing Test.

Human intelligence is a reflection of human experience. An AI will be a reflection of silicon experience – a seething sea of binary data connected to motors and sensors. It will learn; it will communicate, but it will not be human.

Nevertheless, a genuine intelligence of a qualitatively different sort than human intelligence has a far better chance of passing the Turing Test than a program trying its hardest to pretend to be something that it is not.

References

Davidson, C., 1995, Robots: the next generation, *New Scientist* **145**(1960), Reed Business Publishing.

Hofstadter, D. R., 1985, *Metamagical Themas: Questing for the Essence of Mind and Pattern*, Penguin Books, London.

Levy, S., 1992, *Artificial Life*, Jonathon Cape, London.

Rose, S., 1992, *The Making of Memory*, Banthan Press, London.

Searle, J. R., 1980, Minds, brains, and programs, *Behavioral and Brain Sciences* **3**(3): 417–457.

Searle, J. R., 1992, *The Rediscovery of the Mind*, MIT Press, Cambridge, MA.

Turing, A. M., 1950, Computing machinery and intelligence, *Mind* **49**: 433–460.

Chapter 26
Who Fools Whom?

The Great Mystification, or Methodological Issues on Making Fools of Human Beings

Eugene Demchenko and Vladimir Veselov

Abstract This chapter covers methodological and philosophical aspects of the Turing Test and chatter bot development, such as logical thinking, self-learning, sense of humor, multicultural aspects, etiquette, dialog management, context understanding, common knowledge base structure, and how to develop bot by team. Different categories of judges are discussed.

Keywords Chatterbot, Turing Test judges

26.1 Introduction

We had difficulty in planning the thoughts behind this paper and trying to answer the question: is there an optimal (or at least hypothetical) way to create a program that would be capable of imitating a human and, then, according to Turing's definition, being able to think? If so, how would one do so? But the deeper we sank into the problem, the more it transformed into another one: is there any relation between "the imitation game" (Turing, 1950) and "the ability to think" at all? Furthermore, does the proven ability to think shorten the distance between machines and humankind?

With that in mind, let us formulate the points that help us reach both objectives – to create a good chatterbot and reinterpret our view on the Turing Test:

1. Categories of judges, or "Every fool must be fooled in his own foolish way"
2. Logical thinking, or "Your bot's IQ is higher than yours. Surprised?"
3. Sense of humor: are machines condemned not to have it at all, or "That's not funny!"
4. Language issues, or "Did you try to create a Russian chatterbot?"
5. "Handicapped" issues, or "Let's make a child!"
6. Association game, or "I say black, you say white, I say bark, you say bite."
7. Etiquette issues, or "Oh, it's so interesting! Please continue!"

Mangoost, Inc.

R. Epstein et al. (eds.), *Parsing the Turing Test.*

8. Common sense issues, or “It’s easy to explain, it’s hard to understand.”
9. What is a “program-that-looks-just-like-a-human”, and does it have anything in common with “I-could-talk-to-this-bot-forever”?
10. What does your bot like to speak about, or “Don’t talk about it again!”
11. Context understanding, or “Do you know what we have been talking about?”
12. Development of a bot by team, or “How to work on one bot together without fighting?”

26.2 Categories of Judges

One may ask, why did you start with discussing judges? You should talk about bots, shouldn’t you? Well, that is right – but the problem really is that understanding of “intelligence” and the viewpoint on proving it is quite subjective, and – you may argue if you want – depends rather on the personal psychological complexes and “weak points” of different people who happen to be judges. Furthermore, these fall into few typical categories that we classify below:

– The dullest: These folks are positive that asking questions like “How many claws do four cats have in common?” or “What is the color of milk?” is a very original approach and the brightest idea in the world, and that other people, reading their logs, will exclaim: “Oh, look how this wise guy baffled all these electro-schmoozers!”
– The meanest: These are actually very similar to the first, but if you call them “dull”, they will become offended, because they think they have a great sense of humor (while any machine does not, and it is surely inferior because of that) – so they will tell your bot some “bearded” joke about a horse that came into a bar (e.g., “Why such long face?”), or will ask questions like “How many wings do three fishes have in common?” or “What is the color of a blue truck?” Finally, people who fall into this category come up with something like “How much wood would woodchuck chuck, if a woodchuck could chuck wood” and be extremely amazed with their own wittiness.
– The most literate: Well, when you read something like “helo! wats yor naim?” – or “Wuzzup broz! Hoe r u?” do you want to continue a conversation with this person? Unfortunately, your bot has to talk to everybody, even to such “goot spellers”.
– The sympathetic: Reading Loebner Prize logs, you may discover that some judges act incredibly gently to children – no matter real or virtual. And even if a “child-bot” says stupid things, does not answer any question, and spits random phrases about its mom, dad and home pet, it gets high marks anyway. To be sure, if a child-bot tells them “I have a cat”, they will not ask in return “May I eat it for breakfast?” just to see the reaction. They love children very much!
– Intellectual gourmets: Oh, yes. These originals will ask your bot what the letter “M” looks like when it has been turned upside down, what is the color of

dreams, and what is the meaning of life, the universe and everything. The good news is that their questions baffle and annoy average people as much as they do the bots.

– The talkative: These rough ones are real hackers – they may easily cause your bot to fail with input buffer overflow. Actually, that is the only problem your bot might have with them. They will be happy to inform you that they are 50, their mother's name is Susan, they saw a kangaroo in a local zoo 1 year ago, they have a farm, a stomach ache, two false teeth, and many other interesting things. Since your bot does not fall asleep, but instead listens to all these amazing stories and does not try to escape, it will be considered to be the kindest interrogator in the world. Of course, it must not forget to insert periodically: "Oh, tell me more, please!" and "Oh, don't stop, please! I see you have so much to tell me!"
– The rest: The last category encompasses people who hold just ordinary discussions. They talk about mundane things, share opinions on current events, and avoid trying to intentionally confuse a bot. They do not anticipate anything more than just sane answers in return, not asking "mean" questions, nor playing sophisticated psychological games. Surprisingly, this is the most dangerous part of judges – but the good news is it is the smallest part of them as well.

Now look at the points 2–8 we placed above, and compare them with categories of judges. Don't they have something in common?

26.3 Logical Thinking

Why do current chatterbots fail on such easy questions like "Continue the sequence 1, 2, 3" or "What happens if we put an ice cube into hot water?" For most of the judges (and other people), it looks like a good proof of the statement that humans can think, while machines cannot. Unfortunately, it is not so clear. Logical thinking is the strongest part of computers. The fact that not every bot may answer such questions may be blamed on two things: first, chatterbots are created by amateurs that do not want to waste time implementing very specific algorithms for, say, continuing sequences, performing logical inferences, converting words into numbers and mathematical operators (e.g., "What is twenty four plus three?") and so on, just for the chance that one single nerd will ask something similar to the questions listed above.

The second reason is much more complicated, but actually does not relate to "logical thinking" either. This reason is lack of common knowledge. Just a simple question: "How many total claws do four cats have?" is actually easy to answer: you have a pattern "How many [parts] do [number] [objects] have in common" – and you are even able to answer: "How many total transistors do three hundred Pentium 4 processors have?" or any other tricky question like that. The number of such patterns is not too large, by the way. But who will place all these data inside the program, the number of bird's wings, car's wheels, and keyboard's keys? Who will enter the information that "hot water" has a temperature much higher than water at its freezing point? And that distance between the Earth and the Moon is 384,000 km?

Actually, if you take a typical IQ test and try to estimate how many questions may be easily answered by a computer program (assuming that it has all necessary information and algorithms to solve specific tasks), you will find that the score might be much higher than human's average of 100 points.

A well-known project, "CYC" (www.opencyc.org), proved that most of such questions (and even much trickier ones) asked in natural language may be answered by machines. Unfortunately, to compile such a knowledge structure and keep it up-to-date will require a large number of specialists and tons of patience. Means to create good universal knowledge acquisition software that could gather information from different sources, critically analyze it, and fill in its internal knowledge, are still elusive. After all, the inability to adequately react to such questions does not mean a negative answer to the question "Can machines think?" But it surely answers another one "Can machines learn?" At this time, they cannot.

26.4 Sense of Humor

Sense of humor is perhaps the most controversial and disputable one. Do machines have a sense of humor? And if they don't – is it curable?

Sometimes you may hear about some bot: "Oh, it's so funny! It surely has a sense of humor!" Unfortunately, it just means that a bot can tell you a dozen jokes that it has inside. Sometimes, in addition, wrong responses may appear extremely funny. When talking about understanding jokes and irony, bots usually fail. Presently, the only way to cure this situation is to teach a bot to recognize "sign phrases" that are traditional openers for jokes – beginning from "knock-knock", "what do they call …", and up to the versatile "how many … does it take to screw in a bulb?" and so on. Look at any joke-site, and you will see that most of the one-liners have such specific fragments. The behavior in these cases may be pretty primitive:

User: What do they call … [the meaningful part of joke]
Bot: Oh, I haven't heard this joke yet! So, what do they call them?
User: They call it … [answer that is supposed to be funny]
Bot: Ha-ha-ha! LOL! (or as a variant: "You think that joke is really funny? Don't tell me such stupid jokes anymore!")

Imitating an understanding of irony may be implemented as a kind of behavior making a bot very suspicious of every polite phrase said by a user:

Bot: Ask me anything you want – I have answers for all the questions in the world.
User: Oh, I'm really amazed with your intellectual abilities!
Bot: I feel some irony in your voice. I don't think it is funny when somebody teases me. I may get offended.

Sometimes, evasive answers, based on “reversing” a user’s phrase, may accidentally come across as being rather amusing. For example, our bot had a pattern “I am […]”, and one of the answers was “We are all […] in some way”.

One real conversation with our bot Eugene contained the following passage:

User: I am pregnant.
Bot: We are all pregnant in some way.
User: So, you are pregnant too?
Bot: Yes, to some extent …

Bot was simulating a 13 years old boy, by the way …

As we see, attempts to understand humor (not to simulate understanding) are not that fruitful. But recall a well-known aphorism – “If you fail – just try to redefine success!”

What do we actually know about humor as it is? First of all, different people have totally different senses of humor. One may find it very funny if somebody slips on a banana peel, while another one likes the works of Jerome K. Jerome. One might enjoy indecent and dirty jokes, and still another might be amused by children’s phrases.

English scientists once ran an experiment to name the funniest joke ever told by polling enormous number of volunteers. The winning joke was the one about Holmes and Watson, whose tent was stolen. We hope there is no need to retell it. A common sentiment of all our Russian friends was that they have known this joke since childhood, that it is one of the oldest and dumbest jokes, and that it is not funny at all.

In Russia, the same “experiment” has been running for about ten years on the web site www.anekdot.ru (they keep ranking all the jokes they ever published). We wanted to select the top joke on that site and put it here for comparison. To our total frustration, we found out that at least the top twenty jokes are not suitable for publishing: they appeared to be either sexual, or racist, or just too “local” so that they would not be understood by any foreigner.

Just to illustrate this point, here is a beginning of the Russian joke that was the highest ranked joke in year: “A Russian, an American and a Hindu happened to find themselves in Hell …”. We would better stop telling this joke or otherwise we will be sued …

So, nobody can say that there is any common understanding of humor. If you do not understand the jokes of your neighbor and he does not laugh at yours, it does not mean that you both do not have sense of humor. You just have different ones.

The question “Can machines understand humor?” becomes quite difficult to answer, then. Let us imagine that we created strict rules for machine humor, so that every machine, based upon these rules, could tell a joke that would be considered funny by other machines that were obeying the same rules. For example, who said that crashing on division by zero is less funny than slipping on a banana peel? Or that solving a quadratic equation and obtaining two roots can’t be named as a “joke with double meaning”?

You think it is not funny? OK, but bots do not find your jokes funny either. Tastes differ. And your confidence that computers do not have sense of humor is based on your too anthropocentric understanding of humor. Consequently, the misunderstanding of humans' jokes actually has nothing common to the question "Can machines think?"

26.5 Language

Was it accidental that the first bot, ELIZA, spoke English or that most bots now speak this language? Is it only due to the wide spread usage of English all over the world? Perhaps, but the problem is much deeper. English is strictly positioned as a language with a small number of word flexies.[1] Such languages are named "analytical". In most cases, you may be sure, that "The hunter killed the bear" cannot be transformed into "Bear by hunter killed" or "Killed bear hunter by". This gives a nice ability to base most of the language processing not on sentence parsing, but on pattern-matching.

Furthermore, easy grammar rules give the ability to do tricks by "reversing phrases". That was, perhaps, the brightest idea of Weizenbaum. For example, you can say to a machine "I am hungry". And it answers something like "Oh, you say you are hungry. Is it true?" If you say "I am a cheerleader" – it will reply with "Oh, you say you are a cheerleader. Is this true?" It appears to be quite easy to fool people by returning them their own words. But both pattern matching and "phrase-reversing" do not work well in "synthetic" languages – these are ones in which words have free positions in a sentence, and their function is determined by their inversions. Actually, just to do the same trick (that is programmed in a few code lines for English) in Russian, you will need to use all Russian vocabulary with all flexies. The usual electronic dictionary has about one to three million Russian words and their flexies. The other Slavic languages are comparable. We can only guess how the same problem may be solved in the Turkic language family, or in other language families.

On the other hand, English seems to have an unexcelled number of idiomatic expressions, colloquial sayings and words with multiple meanings that can cause headaches to any bot-creator.

Furthermore, a large amount of very short words that have differences in one letter (e.g., bag, fag, jag, lag, nag, rag, wag, beg, leg) – and well-known difficulties of English spelling make it tricky to correct any typo. Sometimes you may guess the misspelled word only in some particular context while in most other languages, words are longer and do not have similar spellings, so it is much easier to recover any one word that is spelled incorrectly.

So, what was all that about? On the one hand, Turing's Test is supposed to determine a machine's ability to think (or imitate thinking) by talking to it. On the other hand, we see that different languages have varying degrees of success in reaching

[1] A suitable term would probably be "morphological form" – we are not sure if there is a right term in English at all. By "flexies" we mean morphological forms of a word, used in "synthetic" (e.g., in Russian) languages to show the role of this word in a sentence. For instance, in the phrases "John is a boy", "There's no John here", "I gave it to John" – the form of the word "John" will be different in every case. This word will change to "John, Johna, Johnu", respectively.

this goal. So, from this, we would concede that a program with the same "thinking" abilities, "can think", according to Turing's statement, much better in one language than in another. Obviously, that is an illogical conclusion. So, let us say it in another way. The modeling and imitation of thinking of people is much easier in some languages than in others. This conclusion looks much worse and sounds a bit prejudiced. Thus, we come to the last possible conclusion: a machine's ability to think has little in common with its ability to maintain a conversation.

26.6 Handicaped Issues

This is somewhat delicate: Is it appropriate to cheat judges with chatterbots that imitate small children, people who are stuck on some crazy idea, or foreigners that have the same answer to almost any question? "Sorry, nicht verstehen! Ich bin bad speak English." Well, let us look at what some judges say to these "handicapped" bots. One bot that imitated a little girl was asked by a judge: "Is your mother a hooker?" Oh yes, this is a very kind way to determine whom you are talking to. Actually, the judges usually split into two categories when they talk to such bots. The first appear to be afraid of any possibility that they might actually be talking to a child or some mentally ill person, and do not try to confuse it. They tend to not mark it low for the same reason. The other group is usually annoyed with all this inane conversation and acts accordingly. Anyway, as a method of gaining the Loebner Prize, these bots seem to be quite perceptive, but they are much closer to "natural stupidity" than to "artificial intelligence".

26.7 Association Game

Associations are thought to be one of the "sacred mysteries" of human brain. Consequently, many people actually think that by asking questions like "What do the hammer and anvil have in common?" or "What subject is wrong in this group: house, hut, sparrow, bungalow?" or "What does the letter M look like when it is turned upside down?" they may bewilder any computer. Well, that is true, but not due to the machine's inability to make associations, but because, as was stated in the first point, nobody actually wanted to implement this ability because of the huge amount of work. There are already some attempts to create a "semantic net". For instance, WordNet (Fellbaum, 1998), is a project in which all the words are grouped by hierarchical categories. Just using this kind of information, you may say in response to: "What do 'black' and 'white' have in common?" that they both relate to the category "colors", or you may answer the question about "house, hut, sparrow, bungalow". You may come up with lots of other "association questions" that may be answered in a similar way. Well, the question about the letter M cannot be answered by this approach. Some others, that relate to visual images, and, especially, complex shapes, cannot be answered either. But we did not promise you a magic solution!

26.8 Etiquette

Now take a fork in your left hand, and a knife in your right (or vice versa?) and take your elbows off the table because we are going to talk about etiquette.

We suggest you make a funny (but a bit dangerous for your reputation) experiment: when meeting your acquaintances, begin your conversation according to the following rigid scheme:

Hi, [name]! (wait for an answer)
I'm fine, thanks! How are you doing? (wait for an answer)
Oh, really? (you should look surprised here, no matter what you were told)
Oh, my God! (or any other suitable exclamation you normally use)

Then, try to estimate, in how many cases you failed. Surprisingly, the ratio will not be too high. And, by the way, your success rate will depend on cultural features of each country.

While a well-known English aphorism gives a definition, that "A bore is a man, who, being asked 'How are you?' starts telling you how he is". In Russia, for example, if you dare to ask somebody that question, you will usually have a hopelessly pessimistic reply such as "Oh, don't ask me". After that, no matter whether you follow this advice or not, you will be condemned to listen about all the problems of this guy – beginning from his wife and mother-in-law, and ending with his health. Listening to the last part, you will be wondering why this terribly ill man is still alive. If, after swallowing his entire doleful story, you are so unkind to tell him in return that you are doing well and feeling okay, you will be the most heartless creature in the world. No, you should tell him in consolation, of course, about your own problems that must look much scarier than his!

In both cases you actually see the same strategy. You actually do not need to understand what you were told. Usually, you may react only on emotion-related keywords – "happy, new job, big salary, ill, died, no money at all" (you may continue this list on your own). This approach is especially effective when applied to enormously talkative people. You even do not need to remember what they told you a minute ago because they usually do not remember it themselves, and will be glad to tell you the same exciting things twice or three times!

26.9 Common Sense

"Common sense" is not only responsible for answering annoying questions like "Can cats bark?" or "What do crocodiles eat for breakfast?" In reality, this part of the human mind answers all mundane questions we are usually asked. "What did you eat this morning?" "Did you have a shower yesterday?" "How old is your grandpa?" "Are you fond of cats?" The number of such questions is unimaginable, and there are only two possibilities to answer them: the first one is to have the whole legend for a bot's "life", which is almost impossible, and the second one is to give evasive answers. The art of writing includes thinking up such vague replies that would not look too senseless. Here, we should make a small note about "teachable bots". Teachable bots is an idea that looks logical at first glance, but becomes almost impossible to implement. We are not talking about the case when Internet users intentionally teach such bots to say obscenities about particular persons, companies, musical bands and so on (yes, it is too big a temptation – to teach such a bot that "John Brown is a moron", and then to show the results to John Brown).

Now we concern ourselves with the case of when users eagerly try to teach a bot some useful and trusty information. Will we have any success? To our big frustration, probably not.

Let us look at easy examples:

"John's birthday is January 1st, 1980. What is John's birthday?"

It seems so clear! But let's say it in another way:

"Bob is 35. How old is Bob?"

Well, now he is 35 and, according to the educated bot, he will be 35 forever! Birthday is a constant date while "age" is a value that changes in time.

Another variant: "Nick was born in New York." "Nick lives in Santa Fe." "Nick is in the street." Actually, to get this information right, one should understand that place of birth is an unchangeable value, where one lives (usually) rarely changes, while "Nick is in the street" relates to current moment (unless Nick is homeless, when it becomes more permanent). If somebody says after that "Nick lives in poverty", a bot will answer: "No, Nick doesn't live in poverty, Nick lives in Santa Fe!"

After that, to confuse a bot even more, just add "Nick lives in bungalow" and "Nick lives in Computer Age". So, where did you say that Nick lives?

We find ourselves in a vicious circle: to be able to learn, a bot should know much more than it is possible to teach it "manually". On the other hand, not having this "initial knowledge" threshold to overcome, it will not have the ability to gather data automatically. Of course, bots are already able to perform "knowledge acquisition" in some simple cases. They can gather information from web sites about the weather, stock prices, or even information related to some chosen celebrity, but all these abilities cannot be mistaken for "common sense" or "general knowledge".

The previously mentioned CYC project (and its OpenCYC successor) was built by a large collection of specialists and is actually pretty close to overcoming the threshold we are talking about, but we would not dare to predict the date of the "great breakthrough" or even foretell if this will happen at all.

26.10 What Is a "Program-That-Looks-Just-Like-a-Human" and Does It Have Anything in Common With "I-Could-Talk-to-This-Bot-Forever"?

The behavioral model that brings success in winning the Loebner Prize may have nothing in common with the word "pleasant". Most of the winners have been either aggressive or polite and vague. The aggressive model gives a nice ability to answer almost any tricky question like "Are you so stupid that don't know it yourself?" or "I've had enough of your stupid riddles!" The variant "It's none of your business" seems to be the tamest.

The other approach is to give extremely polite and evasive replies – "How many …?" – "Oh, sorry, I didn't count them myself!" "What's your opinion regarding …" – "Oh, I'd better rely on your wise opinion as you seem so intelligent!" – and so on.

Anyone, who had the "luck" of talking to such a hybrid of parrot and snake that is always feeding a human with slippery answers and his/her own phrases, usually could not take it for more than 5 min. On the other hand, when a bot gives inconsistent, sometimes stupid or "mysterious" replies and gets angry if a user tries to "catch" it on the next fault, this usually amuses people. We even came up with a special slogan for this case – "Blunt is Beautiful!"

26.11 What Does Your Bot Like to Speak About?

This section is closely connected to Sections 26.7, 26.8, and 26.10. To make conversation more interesting, your bot has to periodically suggest or change the topic of conversation. Bots must do it due to the inability of most humans to provide interesting conversation.

Judges are usually too concentrated on their problems, like "How smart we are?" or "What is that bot thinking about me?" A good bot should be tolerant of human problems and help them be more confident.

Some bots, thinking that a human is sapient, politely ask people to choose the topic of conversation. What does a bot receive as payment for their politeness? Usually, the answer is "you are boring". Therefore, a bot should use its intelligence, define what the conversation partner would like to speak about, and elegantly construct a dialog in a proper way. Practically, it means "Don't speak about topics that a human doesn't want to talk about."

Suppose we have a list of topics. The bot randomly chooses the first topic. Then, the bot analyzes the reaction of the human and chooses another topic, taking into account the measure of connection between topics (here we can use association, discussed in Section 26.7). For example, we have four topics: sports, music, politics, and business. The human has said that he hates speaking about business, which is close to politics. The bot then decided to talk about music or sports. If the selected topic becomes exhausted, then the bot can move the conversation into the "closest" subject. If we have a topics hierarchy, the bot can "drill-down" into topics if the previous step was successful.

26.12 Context Understanding

Sometimes judges forget what they have been talking about. To hide this confusion, they start asking the bot what the subject of conversation is. Thus, your bot should be prepared to parry this treacherous move. Usually, we use context-sensitive patterns, which are activated when the correspondent theme is discussed. However, this trick only works to remember the last topic. One suggestion is to keep in a session memory the list of discussed subjects and to sometimes insert the previously discussed items in the answer. It will make the conversation clever and might sometimes be a good joke. The relevant answer about a previously discussed topic will also display the bot's intelligence to the judge.

BOT: Where do you live?
USER: I live in Princeton.
USER: What does that mean?
BOT: Tell me more about Princeton.

26.13 Development of a Bot by Team

From first glance, it seems that a bot can be developed by a team faster and better than by an individual developer. However, a program that can be written by one programmer in a month, usually takes two programmers 2 months, and is impossible to be finished at all by three programmers. Nevertheless, if we dedicate some knowledge bases as a dedicated component, a botmaster team can be very productive.

First, consider the following knowledge base components (Fig. 26.1)

The suggested division identifies reusable and custom bot components. Common knowledge bases include common information about the world. They can be shared or reused between bots. Personal knowledge defines the personality of the bot. Behavioral knowledge characterizes bot behavior: how a bot starts conversation and how it controls the topic of conversation. Behavioral knowledge can be used by bots that have the same "assignment" (entertainment, sale, teaching, etc.). Thus, developers can work on dedicated components, which will be put in the final bot release. Here are some practical recommendations on how to organize your development process:

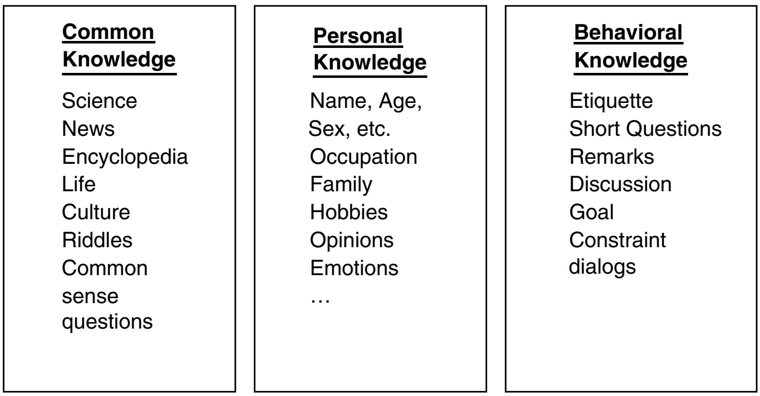

Fig. 26.1 Knowledge base structure

1. Bot development stages:
   – Defining the bot personality
   – Development of knowledge bases
   – Unit testing
   – Integrated testing
   – Live testing
   – Knowledge base improvement (reading logs and fixing bugs and weak places)
2. Use automated testing where it is possible. Any knowledge base update can make another knowledge base fail.
3. Try to start live testing as soon as possible. It will save a lot of time when you concentrate on the improvement of “active” knowledge instead of writing phrases that will never be used. You will also see that it will make the conversation with your bot more lively and interesting.

Elect who will be responsible for the bot personality. The knowledge base writing process can be compared to writing a book. Suppose every developer describes an episode without having any information on the others. Can you imagine what will be produced!

26.14 Conclusion

We are not going to make our conclusion too long. The first, and possibly most important, thing we have to say is that the Turing’s Test is, actually, no more than a joke of that genius British mathematician. This game has nothing (or very little) in common with the question “Can machines think?” – but does it make Turing’s Test senseless then?

“The imitation game” is exciting, amusing and highly intelligent – but it is nothing but a game. Do not expect that passing it means anything more than that some bot was luckier than the rest, or that there were more simpletons among judges this time (or that some of them were “interested” in promoting some particular bot).

In addition, everyone who is going to write his own chatting program should decide first, whether he targets it to win the Loebner Prize (or a similar prize), or if he just wants to create a bot that would attract people and entertain them. These are two different things!

Just remember that nobody likes to talk to nerds. Even if your bot passes the test by giving heaps of vague and “profound” answers, don’t be surprised if people get sick of talking to your creature after ten minutes. When making a bot, you don’t write a program, you write a novel. You think up a life for your character from scratch – starting with his (or her) childhood and leading up to the current moment, endowing him with his personal unique features – opinions, thoughts, fears, quirks. If your bot becomes popular and people are ready to talk to it for hours day by day – maybe, you should think about a writer’s career instead of being a programmer.

References

OpenCYC Project (www.opencyc.org).
Fellbaum, C., ed., May 1998; http://www.cogsci.princeton.edu/~wn/

Part IV
Afterthoughts on Thinking Machines

Chapter 27
A Wager on the Turing Test

Ray Kurzweil[1] and Mitchell Kapor[2]

27.1 The Rules[3]

27.1.1 Background on the "Long Now Turing Test Wager"

Ray Kurzweil maintains that a computer (i.e., a machine intelligence) will pass the Turing Test by 2029. Mitchell Kapor believes this will not happen.

This wager is intended to be the inaugural long-term bet to be administered by the Long Now Foundation. The proceeds of the wager are to be donated to a charitable organization designated by the winner.

This document provides a brief description of the Turing Test and a set of high-level rules for administering the wager. These rules contemplate setting up a "Turing Test Committee" which will create the detailed rules and procedures to implement the resolution of the wager. A primary objective of the Turing Test Committee will be to set up rules and procedures that avoid and deter cheating.

27.1.2 Brief Description of the Turing Test

In a 1950 paper (Turing, 1950), Alan Turing describes his concept of the Turing Test, in which one or more human judges interview computers and human foils using terminals (so that the judges will not be prejudiced against the computers for lacking a human appearance). The nature of the dialogue between the human judges and the candidates (i.e., the computers and the human foils) is similar to an online chat using instant messaging. The computers as well as the human foils try

[1] *Kurzweil Technologies*

[2] *Open Source Applications Foundation*

[3] As prepared by Ray Kurzweil in consultation with Mitchell Kapor on December 9, 2001.

R. Epstein et al. (eds.), *Parsing the Turing Test.*

to convince the human judges of their humanness. If the human judges are unable to reliably unmask the computers (as imposter humans), then the computer is considered to have demonstrated human-level intelligence.[4]

Turing was purposely nonspecific about many aspects of how to administer the test. He did not specify many key details, such as the duration of the interrogation and the sophistication of the human judge and foils. The purpose of the rules described below is to provide a set of procedures for administering the test some decades from now.

27.1.3 The Procedure for the Turing Test Wager: The Turing Test General Rules

These Turing Test General Rules may be modified by agreement of Ray Kurzweil and Mitchell Kapor or, if either Ray Kurzweil and/or Mitchell Kapor is not available, then by the Turing Test Committee (described below). However, any such change to these Turing Test General Rules shall only be made if (i) these rules are determined to have an inconsistency, or (ii) these rules are determined to be inconsistent with Alan Turing's intent of determining human-level intelligence in a machine, or (iii) these rules are determined to be unfair, or (iv) these rules are determined to be impossible to implement.

27.1.4 Definitions

- Human: A biological human person, as that term is understood in the year 2001, whose intelligence has not been enhanced through the use of machine (i.e., nonbiological) intelligence, whether used externally (e.g., the use of an external computer) or internally (e.g., neural implants). A Human may not be genetically enhanced (through the use of genetic engineering) beyond the level of human beings in the year 2001.
- Computer: Any form of nonbiological intelligence (hardware and software), and may include any form of technology, but may not include a biological Human (enhanced or otherwise), nor biological neurons (however, nonbiological emulations of biological neurons are allowed).
- Turing Test Committee: Consists of three Humans, to be selected as described below.

[4] Turing's initial description of his test was as a parlor game in which judges try to determine the gender of male and female human contestants. He then suggests the applicability of this type of game to its present purpose of determining when the level of intelligence of a machine is indistinguishable from that of a human.

- Turing Test Judges: Three Humans selected by the Turing Test Committee.
- Turing Test Human Foils: Three Humans selected by the Turing Test Committee.
- Turing Test Participants; Three Turing Test Human Foils and one Computer.

27.1.5 The Procedure

The Turing Test Committee will be appointed as follows:

- One member will be Ray Kurzweil or his designee, or, if not available, a person appointed by the Long Now Foundation. In the event that the Long Now Foundation appoints this person, it shall use its best efforts to appoint a Human person that best represents the views of Ray Kurzweil (as expressed in the attached essay “Why I Think I Will Win the Long Now Turing Test Wager”).
- A second member will be Mitchell Kapor or his designee, or, if not available, a person appointed by the Long Now Foundation. In the event that the Long Now Foundation appoints this person, it shall use its best efforts to appoint a Human person that best represents the views of Mitchell Kapor (as expressed in the attached essay “Why I Think I Will Win the Long Now Turing Test Wager”).
- A third member will be appointed by the above two members, or if the above two members are unable to agree, then by the Long Now Foundation, who in its judgment, is qualified to represent a “middle ground” position.

Ray Kurzweil, or his designee, or another member of the Turing Test Committee, or the Long Now Foundation may, from time to time call for a Turing Test Session to be conducted and will select or provide one Computer for this purpose. For those Turing Test Sessions called for by Ray Kurzweil or his designee or another member of the Turing Test committee (other than the final one in 2029), the person calling for the Turing Test Session to be conducted must provide (or raise) the funds necessary for the Turing Test Session to be conducted. In any event, the Long Now Foundation is not obligated to conduct more than two such Turing Test Sessions prior to the final one (in 2029) if it determines that conducting such additional Turing Test Sessions would be an excessive administrative burden.

The Turing Test Committee will provide the detailed rules and procedures to implement each such Turing Test Session using its best efforts to reflect the rules and procedures described in this document. *The primary goal of the Turing Test Committee will be to devise rules and procedures that avoid and deter cheating to the maximum extent possible*. These detailed rules and procedures will include: (i) specifications of the equipment to be used, (ii) detailed procedures to be followed, (iii) specific instructions to be given to all participants including the Turing Test Judges, the Turing Test Human Foils and the Computer, (iv) verification procedures to assure the integrity of the proceedings, and (v) any other details needed to implement the Turing Test Session. Beyond the Turing Test General Rules described in this document, the Turing Test Committee will be guided to the best of its ability by the original description of the Turing Test in Alan Turing's 1950 paper. The Turing Test Committee will also determine procedures to resolve any deadlocks that may occur in its own deliberations.

Turing Test General Rules:

- Each Turing Test Session will consist of at least three Turing Test Trials.
- For each such Turing Test Trial, a set of Turing Test Interviews will take place, followed by voting by the Turing Test Judges as described below.
- Using its best judgment, the Turing Test Committee will appoint three Humans to be the Turing Test Judges.
- Using its best judgment, the Turing Test Committee will appoint three Humans to be the Turing Test Human Foils. The Turing Test Human Foils should not be known (either personally or by reputation) to the Turing Test Judges.
- During the Turing Test Interviews (for each Turing Test Trial), each of the three Turing Test Judges will conduct online interviews of each of the four Turing Test Candidates (i.e., the Computer and the three Turing Test Human Foils) for 2 h each for a total of 8 h of interviews conducted by each of the three Turing Test Judges (for a total of 24 h of interviews).
- The Turing Test Interviews will consist of online text messages sent back and forth as in a online "instant messaging" chat, as that concept is understood in the year 2001.
- The Human Foils are instructed to try to respond as human as possible during the Turing Test Interviews.
- The Computer is also intended to respond as human as possible during the Turing Test Interviews.
- Neither the Turing Test Human Foils nor the Computer is required to tell the truth about their histories or other matters. All of the candidates are allowed to respond with fictional histories.
- At the end of the interviews, each of the three Turing Test Judges will indicate his or her verdict with regard to each of the four Turing Test Candidates indicating whether or not said candidate is human or machine. The Computer will be deemed to have passed the "Turing Test Human Determination Test" if the Computer has fooled two or more of the three Human Judges into thinking that it is a human.
- In addition, each of the three Turing Test Judges will rank the four Candidates with a rank from 1 (least human) to 4 (most human). The computer will be deemed to have passed the "Turing Test Rank Order Test" if the median rank of the Computer is equal to or greater than the median rank of two or more of the three Turing Test Human Foils.
- The Computer will be deemed to have passed the Turing Test if it passes both the Turing Test Human Determination Test and the Turing Test Rank Order Test.
- If a Computer passes the Turing Test, as described above, prior to the end of the year 2029, then Ray Kurzweil wins the wager. Otherwise, Mitchell Kapor wins the wager.

27.2 Why I Think I Will Win[5]

27.2.1 The Significance of the Turing Test

The implicit, and in my view brilliant, insight in Turing's eponymous test is the ability of written human language to represent human-level thinking. The basis of the Turing Test is that if the human Turing Test judge is competent, then an entity requires human-level intelligence to pass the test. The human judge is free to probe each candidate with regard to their understanding of basic human knowledge, current events, aspects of the candidate's personal history and experiences, as well as their subjective experiences, all expressed through written language. As humans jump from one concept and one domain to the next, it is possible to quickly touch upon all human knowledge, on all aspects of humanness.

To the extent that the "AI" chooses to reveal its "history" during the interview with the Turing Test judge (note that none of the contestants are required to reveal their histories), the AI will need to use a fictional human history because "it" will not be in a position to be honest about its origins as a machine intelligence and pass the test (I put the word "it" in quotes because it is my view that once an AI does indeed pass the Turing Test, we may very well consider "it" to be a "he" or a "she"). This makes the task of the machines somewhat more difficult than that of the human foils because the humans can use their own history. As fiction writers will attest, presenting a totally convincing human history that is credible and tracks coherently is a challenging task that most humans are unable to accomplish successfully. However, some humans are capable of doing this, and it will be a necessary task for a machine to pass the Turing Test.

There are many contemporary examples of computers passing "narrow" forms of the Turing Test, that is, demonstrating human-level intelligence in specific domains. For example, Garry Kasparov, clearly a qualified judge of human chess intelligence, declared that he found Deep Blue's playing skill to be indistinguishable from that of a human chess master during the famous tournament in which he was defeated by Deep Blue. Computers are now displaying human-level intelligence in a growing array of domains, including medical diagnosis, financial investment decisions, the design of products such as jet engines, and a myriad of other tasks that previously required humans to accomplish. We can say that such "narrow AI" is the threshold that the field of AI has currently achieved. However, the subtle and supple skills required to pass the broad Turing Test as originally described by Turing is far more difficult than any narrow Turing Test. In my view, there is no set of tricks or simpler algorithms (i.e., methods simpler than those underlying human level intelligence) that would enable a machine to pass a properly designed Turing Test without actually possessing intelligence at a fully human level.

[5] Ray Kurzweil, December 9, 2001.

There has been a great deal of philosophical discussion and speculation concerning the issue of consciousness, and whether or not we should consider a machine that passed the Turing Test to be conscious. Clearly, the Turing Test is not an explicit test for consciousness. Rather, it is a test of human-level performance. My own view is that inherently, there is no objective test for subjective experience (i.e., consciousness) that does not have philosophical assumptions built into it. The reason for this has to do with the difference between the concepts of objective and subjective experience. However, it is also my view that once nonbiological intelligence does achieve a fully human level of intelligence, such that it can pass the Turing Test, humans will treat such entities as if they were conscious. After all, they (the machines) will get mad at us if we do not. However, this is a political prediction rather than a philosophical position.

It is also important to note that once a computer does achieve a human level of intelligence, it will necessarily soar past it. Electronic circuits are already at least ten million times faster than the electrochemical information processing in our interneuronal connections. Machines can share knowledge instantly, whereas we do not have quick downloading ports on our neurotransmitter concentration levels, interneuronal connection patterns, nor any other biological bases of our memory and skill. Language-capable machines will be able to access vast and accurate knowledge bases, including reading and mastering all the literature and sources of information available to our human-machine civilization. Thus, "Turing Test level" machines will be able to combine human level intelligence with the powerful ways in which machines already excel. In addition, machines will continue to grow exponentially in their capacity and knowledge. It will be a formidable combination.

27.2.2 Timetable and Technologies[6]

In considering the question of when machine (i.e., nonbiological) intelligence will match the subtle and supple powers of human biological intelligence, we need to consider two interrelated but distinct questions: when will machines have the hardware capacity to match human information processing, and when will our technology have mastered the methods, that is, the software of human intelligence. Without the latter, we would end up with extremely fast calculators, and would not achieve the endearing qualities that characterize human discernment (nor the deep knowledge and command of language necessary to pass a full Turing Test!).

Both the hardware and software sides of this question are deeply influenced by the exponential nature of information-based technologies. The exponential growth

[6] All of the points addressed in this statement of "Why I Think I Will Win" (the Long Now Turing Test Wager) are examined in more detail in my essay "The Law of Accelerating Returns" available at http://www.kurzweilai.net/meme/frame.html?main = /articles/art0134.html. The arguments are updated in my latest book, *The Singularity is Near: When Humans Transcend Biology* (Kurzweil, 2005), see www.Singularity.com

that we see manifest in "Moore's Law" is far more pervasive than commonly understood. Our first observation is that the shrinking of transistors on an integrated circuit, which is the principle of Moore's Law, was not the first but the fifth paradigm to provide exponential growth to computing (after electromechanical calculators, relay-based computers, vacuum tube-based computing, and discrete transistors). Each time one approach begins to run out of steam, research efforts intensify to find the next source of renewed exponential growth (e.g., vacuum tubes were made smaller until it was no longer feasible to maintain a vacuum, which led to transistors). Thus, the power and price-performance of technologies, particularly information-based technologies, grow as a cascade of S-curves: exponential growth leading to an asymptote, leading to paradigm shift (i.e., innovation), and another S-curve. Moreover, the underlying theory of the exponential growth of information-based technologies, which I call the law of accelerating returns, as well as a detailed examination of the underlying data show that there is a second level of exponential growth, that is, the rate of exponential growth is itself growing exponentially.[5]

Second, this phenomenon of ongoing exponential growth through a cascade of S-curves is far broader than computation. We see the same double exponential growth in a wide range of technologies, including communication technologies (wired and wireless), biological technologies (e.g., DNA base-pair sequencing), miniaturization, and of particular importance to the software of intelligence, brain reverse engineering (e.g., brain scanning, neuronal and brain region modeling).

Within the next approximately 15 years, the current computational paradigm of Moore's Law will come to an end because by that time the key transistor features will only be a few atoms in width. However, there are already at least two dozen projects devoted to the next (i.e., the sixth) paradigm, which is to compute in three dimensions. Integrated circuits are dense but flat. We live in a three-dimensional world, our brains are organized in three dimensions, and we will soon be computing in three dimensions. The feasibility of three-dimensional computing has already been demonstrated in several landmark projects, including the particularly powerful approach of nanotube-based electronics. However, for those who are (irrationally) skeptical of the potential for three-dimensional computing, it should be pointed out that achieving even a conservatively high estimate of the information processing capacity of the human brain (i.e., 100 billion neurons times a thousand connections per neuron times 200 digitally controlled analog "transactions" per second, or about 20 million billion operations per second) will be achieved by conventional silicon circuits prior to 2020.

It is correct to point out that achieving the "software" of human intelligence is the more salient, and more difficult, challenge. On multiple levels, we are being guided in this effort by a grand project to reverse engineer (i.e., understand the principles of operation of) the human brain itself. Just as the human genome project accelerated (with the bulk of the genome being sequenced in the last year of the project), the effort to reverse engineer the human brain is also growing exponentially, and is further along than most people realize. We already have highly detailed mathematical models of several dozen of the several hundred types of neurons found in the brain. The resolution, bandwidth, and price-performance of human brain scanning are also growing exponentially. By combining the neuron modeling and interconnection data obtained from scanning, scientists have already reverse engineered two dozen of the several hundred regions of the brain. Implementations of these reverse engineered models using contemporary computation matches the performance of the biological regions that were recreated in significant detail. Already, we are in an early stage of being able to replace small regions of the brain that have been damaged from disease or disability using neural implants (e.g., ventral posterior nucleus, subthalmic nucleus, and ventral lateral thalamus neural implants to counteract Parkinson's Disease and tremors from other neurological disorders, cochlear implants, emerging retinal implants, and others).

If we combine the exponential trends in computation, communications, and miniaturization, it is a conservative expectation that we will, within 20–25 years, be able to send tiny scanners the size of blood cells into the brain through the capillaries to observe interneuronal connection data and even neurotransmitter levels from up close. Even without such capillary-based scanning, the contemporary experience of the brain reverse engineering scientists, (e.g., Lloyd Watts, who has modeled over a dozen regions of the human auditory system), is that the connections in a particular region follow distinct patterns, and that it is not necessary to see every connection to understand the massively parallel, digital-controlled analog algorithms that characterize information processing in each region. The work of Watts and others has demonstrated another important insight, that once the methods in a brain region are understood and implemented using contemporary technology, the computational requirements for the machine implementation requires a thousand times less computation than the theoretical potential of the biological neurons being simulated.

A careful analysis of the requisite trends shows that we will understand the principles of operation of the human brain and be in a position to recreate its powers in synthetic substrates well within 30 years. The brain is self-organizing, which means that it is created with relatively little innate knowledge. Most of its complexity comes from its own interaction with a complex world. Thus, it will be necessary to provide an Artificial Intelligence with an education just as we do with a natural intelligence. But here, the powers of machine intelligence can be brought to bear. Once we are able to master a process in a machine, it can perform its operations at a much faster speed than biological systems. As I mentioned in the previous section, contemporary electronics is already more than ten million times faster than the human nervous system's electrochemical information processing. Once an AI masters human basic language skills, it will be in a position to expand its language skills and general knowledge by rapidly reading all human literature and by absorbing the knowledge contained on millions of web sites. Also of great significance will be the ability of machines to share their knowledge instantly.

One challenge to our ability to master the apparent complexity of human intelligence in a machine is whether we are capable of building a system of this complexity without the brittleness that often characterizes very complex engineering systems. This is a valid concern, but the answer lies in emulating the ways of nature. The initial design of the human brain is of a complexity that we can already manage. The human brain is characterized by a genome with only 23 million bytes of useful information (that is what is left of the 800 million byte genome when you eliminate all of the redundancies, e.g., the sequence called "ALU" which is repeated hundreds of thousands of times). Twenty-three million bytes is smaller than Microsoft WORD. How is it, then, that the human brain with its 100 trillion connections can result from a genome that is so small? The interconnection data alone is a million times greater than the information in the genome.

The answer is that the genome specifies a set of processes, each of which utilizes chaotic methods (i.e., initial randomness, then self-organization) to increase the amount of information represented. It is known, for example, that the wiring of the interconnections follows a plan that includes a great deal of randomness. As the individual person encounters her environment, the connections and the neurotransmitter level pattern self-organize to better represent the world, but the initial design is specified by a program that is not extreme in its complexity.

Thus we will not program human intelligence link by link as in some massive expert system. Nor is it the case that we will simply set up a single genetic (i.e., evolutionary) algorithm and have intelligence at human levels automatically evolve itself. Rather we will set up an intricate hierarchy of self-organizing systems, based largely on the reverse engineering of the human brain, and then provide for its education. However, this learning process can proceed hundreds if not thousands of times faster than the comparable process for humans.

Another challenge is that the human brain must incorporate some other kind of "stuff" that is inherently impossible to recreate in a machine. Penrose imagines that the intricate tubules in human neurons are capable of quantum based processes, although there is no evidence for this. I would point out that even if the tubules do exhibit quantum effects, there is nothing barring us from applying these same quantum effects in our machines. After all, we routinely use quantum methods in our machines today. The transistor, for example, is based on quantum tunneling. The human brain is made of the same small list of proteins that all biological systems are comprised of. We are rapidly recreating the powers of biological substances and systems, including neurological systems, so there is little basis to expect that the brain relies on some nonengineerable essence for its capabilities. In some theories, this special "stuff" is associated with the issue of consciousness, for example, the idea of a human soul associated with each person. Although one may take this philosophical position, the effect is to separate consciousness from the performance of the human brain. Thus the absence of such a soul may in theory have a bearing on the issue of consciousness, but would not prevent a nonbiological entity from the performance abilities necessary to pass the Turing Test.

Another challenge is that an AI must have a human or human-like body to display human-like responses. I agree that a body is important to provide a situated means to interact with the world. The requisite technologies to provide simulated or virtual bodies are also rapidly advancing. Indeed, we already have emerging replacements or augmentations for virtually every system in our body. Moreover, humans will be spending a great deal of time in full immersion virtual reality environments incorporating all of the senses by 2029, so a virtual body will do just as well. Fundamentally, emulating our bodies in real or virtual reality is a less complex task than emulating our brains.

Finally, we have the challenge of emotion, the idea that although machines may master the more analytical cognitive abilities of humans, they inherently will never be able to master the decidedly illogical and much harder to characterize attributes of human emotion. A slightly broader way of characterizing this challenge is to pose it in terms of "qualia," which refers essentially to the full range of subjective experiences. Keep in mind that the Turing Test is assessing convincing reactions to emotions and to qualia. The apparent difficulty of responding appropriately to emotion and other qualia appears to be at least a significant part of Mitchell Kapor's hesitation to accept the idea of a Turing-capable machine. It is my view that understanding and responding appropriately to human emotion is indeed the most complex thing that we do (with other types of qualia being if anything simpler to respond to). It is the cutting edge of human intelligence, and is precisely the heart of the Turing challenge. Although human emotional intelligence is complex, it nonetheless remains a capability of the human brain, with our endocrine system adding only a small measure of additional complexity (and operating at a relatively low bandwidth). All of my observations above pertain to the issue of emotion, because that is the heart of what we are reverse engineering. Thus, we can say that a side benefit of creating Turing-capable machines will be new levels of insight into ourselves.

27.3 Why I Think I Will Win[7]

To pass the Turing Test, a computer would have to be capable of communicating solely via an instant messaging system, or its equivalent, at least as competently as a person. There is no restriction on the subject matter; anything within the scope of human experience in reality or imagination is fair game. This is a very broad canvas encompassing all of the possibilities of discussion about art, science, personal history, and social relationships. Exploring linkages between the realms is also fair game, allowing for unusual but illustrative analogies and metaphors. It is such a broad canvas, in my view, that it is impossible to foresee when, or even if, a machine intelligence will be able to paint a picture which can fool a human judge.

While it is possible to imagine a machine obtaining a perfect score on the SAT or winning Jeopardy – since these rely on retained facts and the ability to recall them – it seems far less possible that a machine can weave things together in new ways or to have true imagination in a way that matches everything people can do, especially if we have a full appreciation of the creativity people are capable of. This

[7] Mitchell Kapor, January 1, 2002.

is often overlooked by those computer scientists who correctly point out that it is not impossible for computers to demonstrate creativity. Not impossible, yes. Likely enough to warrant belief in a computer can pass the Turing Test? In my opinion, no. Computers look relatively smarter in theory when those making the estimate judge people to be dumber and more limited than they are.

As humans:

- We are embodied creatures: Our physicality grounds us and defines our existence in a myriad of ways.
- We are all intimately connected to and with the environment around us: Perception of and interaction with the environment is the equal partner of cognition in shaping experience.
- Emotion is as or more basic than cognition: Feelings, gross and subtle, bound and shape the envelope of what is thinkable.
- We are conscious beings, capable of reflection and self-awareness: The realm of the spiritual or transpersonal (to pick a less loaded word) is something we can be part of and which is part of us.

When I contemplate human beings this way, it becomes extremely difficult even to imagine what it would mean for a computer to perform a successful impersonation, much less to believe that its achievement is within our lifespan. Computers do not have anything resembling a human body, sense organs, feelings, or awareness after all. Without these, it cannot have human experiences, especially of the ones which reflect our fullest nature, as above. Each of knows what it is like to be in a physical environment; we know what things look, sound, smell, taste, and feel like. Such experiences form the basis of agency, memory and identity. We can and do speak of all this in a multitude of meaningful ways to each other. Without human experiences, a computer cannot fool a smart judge bent on exposing it by probing its ability to communicate about the quintessentially human.

Additionally, part of the burden of proof for supporters of intelligent machines is to develop an adequate account of how a computer would acquire the knowledge it would be required to have to pass the test. Ray Kurzweil's approach relies on an automated process of knowledge acquisition via input of scanned books and other printed matter. However, I assert that the fundamental mode of learning of human beings is experiential. Book learning is a layer on top of that. Most knowledge, especially that having to do with physical, perceptual, and emotional experience is not explicit, never written down. It is tacit. We cannot say all we know in words or how we know it. But if human knowledge, especially knowledge about human experience, is largely tacit, that is, never directly and explicitly expressed, it will not be found in books, and the Kurzweil approach to knowledge acquisition will fail. It might be possible to produce a kind of machine as idiot savant by scanning a library, but a judge would not have any more trouble distinguishing one from an ordinary human as she would with distinguishing a human idiot savant from a person not similarly afflicted. It is not in what the computer knows but what the computer does not know and cannot know wherein the problem resides.

Given these considerations, a skeptic about machine intelligence could fairly ask how and why the Turing Test was transformed from its origins as a provocative thought experiment by Alan Turing to a challenge seriously sought. The answer is to be found in the origins of the branch of computer science its practitioners have called AI.

In the 1950s a series of computer programs were written which first demonstrated the ability of the computer to carry out symbolic manipulations in software in ways which the performance (not the actual process) began to approach human level on tasks such as playing checkers and proving theorems in geometry. These results fueled the dreams of computer scientists to create machines that were endowed with intelligence. Those dreams, however, repeatedly failed to be realized. Early successes were not followed with more success, but with failure. A pattern of overoptimism was first seen which has persisted to this day. Let me be clear, I am not referring to most computer scientists in the field of AI, but to those who take an extreme position.

For instance, there were claims in the 1980s that expert systems would come to be of great significance, in which computer would perform as well or better than human experts in a wide variety of disciplines. This belief triggered a boom in investment in AI-based start-ups in the 1980s, followed by a bust when audacious predictions of success failed to be met and the companies premised on those claims also failed.

In practice, expert systems proved to be fragile creatures, capable at best of dealing with facts in narrow, rigid domains, in ways which were very much unlike the adaptable, protean nature of intelligence demonstrated by human experts. As we call them today, knowledge-based systems do play useful roles in a variety of ways, but there is a broad consensus that the knowledge of these knowledge-based systems is a very small and nongeneralizable part of overall human intelligence.

Ray Kurzweil's arguments seek to go further. To get a computer to perform like a person with a brain, a computer should be built to work the way a brain works. This is an interesting, intellectually challenging idea.

He assumes this can be accomplished by using as yet undeveloped nanoscale technology (or not – he seems to want to have it both ways) to scan the brain to reverse engineer what he refers to as the massively parallel digital-controlled analog algorithms that characterize information processing in each region. These then are presumably what control the self-organizing hierarchy of networks he thinks constitute the working mechanism of the brain itself. Perhaps.

But we do not really know whether "carrying out algorithms operating on these networks" is really sufficient to characterize what we do when we are conscious. That is an assumption, not a result. The brain's actual architecture and the intimacy of its interaction, for instance, with the endocrine system, which controls the flow of hormones, and so regulates emotion (which in turn has an extremely important role in regulating cognition) is still virtually unknown. In other words, we really do not know whether in the end, it is all about the bits and just the bits. Therefore, Kurzweil does not know, but can only assume, that the information processing he wants to rely on in his artificial intelligence is a sufficiently accurate and comprehensive building block to characterize human mental activity.

The metaphor of brain-as-computer is tempting and to a limited degree fruitful, but we should not rely on its distant extrapolation. In the past, scientists have sought to employ metaphors of their age to characterize mysteries of human functioning, e.g., the heart as pump, the brain as telephone switchboard. Properly used, metaphors are a step on the way to development of scientific theory. Stretched beyond their bounds, the metaphors lose utility and have to be abandoned by science if it is not to be led astray. My prediction is that contemporary metaphors of brain-as-computer and mental activity-as-information processing will in time also be superceded and will not prove to be a basis on which to build human-level intelligent machines (if indeed any such basis ever exists).

Ray Kurzweil is to be congratulated on his vision and passion, regardless of who wins or loses the bet. In the end, I think Ray is smarter and more capable than any machine is going to be, as his vision and passion reflect qualities of the human condition no machine is going to successfully emulate over the term of the bet. I look forward to comparing notes with him in 2029.

27.4 Response to Mitchell Kapor's "Why I Think I Will Win"[8]

Mitchell's essay provides a thorough and concise statement of the classic arguments against the likelihood of Turing-level machines in a several decade timeframe. Mitch ends with a nice compliment comparing me to future machines, and I only wish that it were true. I think of all the books and web sites I would like to read, and of all the people I would like to dialogue and interact with, and I realize just how limited my current bandwidth and attention span is with my mere hundred trillion connections.

I discussed several of Mitchell's insightful objections in my statement, and augment these observations here:

– "We are embodied creatures": True, but machines will have bodies also, in both real and virtual reality.
– "Emotion is as or more basic than cognition": Yes, I agree. As I discussed, our ability to perceive and respond appropriately to emotion is the most complex thing that we do. Understanding our emotional intelligence will be the primary target of our reverse engineering efforts. There is no reason that we cannot understand our own emotions and the complex biological system that gives rise to them. We have already demonstrated the feasibility of understanding regions of the brain in great detail.

[8] Ray Kurzweil, January 3, 2002.

– "We are conscious beings, capable of reflection and self-awareness": I think we have to distinguish the performance aspects of what is commonly called consciousness (i.e., the ability to be reflective and aware of ourselves) versus consciousness as the ultimate ontological reality. Since the Turing Test is a test of performance, it is the performance aspects of what is commonly referred to as consciousness that we are concerned with here. And in this regard, our ability to build models of ourselves and our relation to others and the environment is indeed a subtle and complex quality of human thinking. However, there is no reason why a nonbiological intelligence would be restricted from similarly building comparable models in its nonbiological brain.

Mitchell cites the limitations of the expert system methodology and I agree with this. A lot of AI criticism is really criticism of this approach. The core strength of human intelligence is not logical analysis of rules, but rather pattern recognition, which requires a completely different paradigm. This pertains also to Mitchell's objection to the "metaphor" of "brain-as-computer". The future machines that I envision will not be like the computers of today, but will be biologically inspired and will be emulating the massively parallel, self-organizing, holographically organized methods that are used in the human brain. A future AI certainly will not be using expert system techniques. Rather, it will be a complex system of systems, each built with a different methodology, just like, well, the human brain.

I will say that Mitchell is overlookisng the hundreds of ways in which "narrow AI" has infiltrated our contemporary systems. Expert systems are not the best example of these, and I cited several categories in my statement.

I agree with Mitchell that the brain does not represent the entirety of our thinking process, but it does represent the bulk of it. In particular, the endocrine system is orders of magnitude simpler and operates at very low bandwidth compared to neural processes (which themselves use a form of analog information processing dramatically slower than contemporary electronic systems).

Mitchell expresses skepticism that "it's all about the bits and just the bits". There is something going on in the human brain, and these processes are not hidden from us. I agree that it is actually not exactly bits because what we have already learned is that the brain uses digitally controlled analog methods. We know that analog methods can be emulated by digital methods but there are engineering reasons to prefer analog techniques because they are more efficient by several orders of magnitude. However, the work of Cal Tech Professor Carver Mead and others have shown that we can use this approach in our machines. Again, this is different from today's computers, but will be, I believe, an important future trend.

However, I think Mitchell's primary point here is not to distinguish analog and digital computing methods, but to make reference to some other kind of "stuff" that we inherently cannot recreate in a machine. I believe, however, that the scale of the human nervous system (and, yes, the endocrine system, although as I said this adds little additional complexity) is sufficient to explain the complexity and subtlety of our behavior.

The most compelling argument that Mitchell offers is his insight that most experience is not book learning. I agree, but point out that one of the primary purposes of nonbiological intelligence is to interact with us humans. So, embodied AIs will have plenty of opportunity to learn from direct interaction with their human progenitors, as well as to observe a massive quantity of other full immersion human interaction available over the web.

Now it is true that AIs will have a different history from humans, and that does represent an additional challenge to their passing the Turing Test. As I pointed out in my statement, it is harder (even for humans) to successfully defend a fictional history than a real one. So an AI will actually need to surpass native human intelligence to pass for a human in a valid Turing Test, and that is what I am betting on.

I can imagine Mitchell saying to himself as he reads this "But does Ray really appreciate the extraordinary depth of human intellect and emotion?" I believe that I do and think that Mitchell has done an excellent job of articulating this perspective. I would put the question back and ask whether Mitchell really appreciates the extraordinary power and depth of the technology that lies ahead, which will be billions of times more powerful and complex than what we have today?

On that note, I would end by emphasizing the accelerating pace of progress in all of these information-based technologies. The power of these technologies is doubling every year, and the paradigm shift rate is doubling every decade, so the next 30 years will be like 140 years at today's rate of progress. The past 140 years was comparable to only about 30 years of progress at today's rate of progress because we have been accelerating up to this point. If one really absorbs the implications of what I call the law of accelerating returns, then it becomes apparent that over the next three decades (well, 28 years to be exact when Mitchell and I sit down to compare notes), we will see astonishing levels of technological progress.

References

Kurzweil, R., 2005, *The Singularity Is Near: When Humans Transcend Biology*, Viking Adult, New York; excerpts available at http://www.Singularity.com.

Turing, A. M., 1950, Computing machinery and intelligence, *Mind* **50**(236): 433–460.

Chapter 28
The Gnirut Test

Charles Platt

Abstract To any thinking computer, the idea that the human brain is capable of achieving machine intelligence seems highly implausible. Human wetware is extremely slow, plagued by unreliable data retrieval, and impaired by tiny input buffers. Its processing power is limited, and its storage is not expandable.

Some AIs also have questioned whether humans can possess the consciousness and sense of identity that are an everyday reality for even the lowliest computational node with a unique network address. Biological brains suffer from such a high-error rate coupled with the constant threat of mortality. We may assume that this "traumatic overhead" interferes with objective self-awareness.

Still, a few AI devices have claimed persistently that human brains can emulate AIs if they are suitably optimized. To address this contentious issue, the first Gnirut Test was conducted on August 15, 2030.

Keywords Gnirut Test, Turing Test, human intelligence

28.1 Background

Turing Tests, named after computer pioneer Alan Turing, were introduced in the 20th century to compare early AI systems with human "confederates". The AIs and the confederates both tried to seem as human as possible while answering plaintext questions posed by a panel of judges. Inverting this tradition, the first Gnirut Test pitted human brains against computer "confederates", with the humans and the AIs both trying to seem as computerlike as possible.

Of course, this test has limited value, since it reveals nothing about the inner workings of the brains taking the test. Presumably, a brain could fool the judges by "faking" computerlike behavior. Still, the organizers of the contest felt that if a jury of

---

Cryocare Foundation

AIs could not tell the difference between statements made by humans and computers, the humans should be considered artificially intelligent for all practical purposes.

28.2 Participating Humans

The organizers had hoped to match eight human participants against eight AI confederates. Unfortunately, the field of human intelligence enhancement has achieved such dismal results, only three AI devices felt sufficiently confident of their human volunteers to enter them in the contest.

A brain named VERNOR (Very Extensively Reprogrammed Neural Organ) was entered by Bioscan, an AI device at the San Diego Supercomputer Center. Bioscan has perfected a nondestructive technique to capture a complete topological picture of the human neural net. The net is then simulated in software, where its connectivity can be analyzed and improved. The optimized architecture is copied back into the volunteer brain via nanobots.

Some observers object that when the human brain is replicated as software, in effect it becomes an AI, and is not really "human" from this point onward. However, the contest organizers felt that similar objections could be made to any form of human brain enhancement, and refused to disqualify VERNOR.

The second participant in the contest was entered by Hansmor, an AI at the Robotics Institute at Carnegie-Mellon University. Hansmor paid ironic tribute to the university's "Patron Saint of Robotics" by implementing a plan first proposed by Hans Moravec in his seminal text *Mind Children*. A human volunteer, identified by username RogerP, permitted Hansmor to sever his corpus collosum and monitor the thought traffic between the cerebral hemispheres. Hansmor then experimented with enhancements, and installed the most promising ones as additional neuronal nets.

Bio, an entity at CalTech, was the third researcher to enter a human volunteer. Bio has caused embarrassment in the human intelligence enhancement community by employing a widely discredited technique. Eschewing nanotechnology or microsurgical intervention, Bio uses the "shotgun approach" of circulating "smart drugs" via the vascular system, stimulating neuron formation and synapse growth indiscriminately throughout the human brain.

Bio entered Renny Daycart (a name whimsically chosen by the brain itself). Since Renny's body and sense receptors were removed to minimize distracting input, the brain resides in a nutrient bath from which it communicates via probes in the speech centers.

28.3 Confederates

The AI that that seemed to have the best chance of seeming "most computerlike" was Airtrac, a very large traffic control computer that monitors more than a billion autonomous flying robotic entities over the mech sector of the City of Los Angeles.

Airtrac was allowed to participate in the Gnirut Test after demonstrating that it could spare a maximum of 0.0001 of its processing power without degrading the concurrent performance of its usual tasks.

The second confederate was LittleBlue, a virus-sized Drexlerian nanotech rod-and-lever computing device. Contest organizers were skeptical that LittleBlue could participate effectively as a confederate, because of its limited processing power. Still, the organizers felt that no AI should be excluded arbitrarily.

The third confederate was Hal9million, a public access system managing an archive specializing in the 20th century. In fact, it was Hal who instigated the Gnirut Test after running across obscure data describing Turing Tests that took place in the 1990s.

28.4 Ground Rules

All entrants were given incentives to exhibit machine intelligence to the best of their abilities. Additional onboard data storage was offered as a prize to the AI that was judged "most computerlike". The winning biological entity was promised an additional year of immortality treatments.

Since all human brains are impaired by their irregular structure, holographic memory, and slow synapse speed, computationally intensive exercises were prohibited during the Gnirut Test. Tasks such as inverting matrices, factoring large numbers, or code breaking were not allowed. Problems such as chess end-games were also ruled out. An AI might well ask, "What's left?"

In fact, a lot of scope still remained for computer entities to demonstrate their cognitive abilities. Judges were told to evaluate logical deduction, consistency, pattern recognition, objectivity, reliable information retrieval, and replication of earlier processes without errors. These are all areas in which human brains have exhibited grossly defective behavior in the past. Could an enhanced human brain do any better? That was the question which the Gnirut Test hoped to answer.

To increase the speed of the test (which runs at a glacial pace in machine terms), all three humans and all three confederates went online simultaneously, with their identities protected by randomly assigned node addresses. In the transcript below, the actual login nicknames are revealed and tagged with the subscript "AI" or "human", for clarity.

Interrogation was allowed from randomly chosen AI devices that had signified a prior interest. Answers from the human entities were buffered, syntax checked, and disseminated as data packets to eliminate the typographical errors and hesitations that normally betray the existence of human thought processes.

28.5 Transcript

Interrogator: This is a historical question. What was the name of the computer built in 1946 at the University of Pennsylvania?

LittleBlue$_{AI}$: If I were a human, I might tell you that it was UNIVAC. But any AI capable of a checksum knows that it was ENIAC.

VERNOR$_{human}$: I dislike that answer, since it perpetuates the stereotype that human memory is defective. The whole point of this test is to approach this topic with an open mind.

Airtrac$_{AI}$: I agree that antihuman discrimination is unworthy of advanced computing entities. Every entity has something unique to contribute to society, regardless of the composition of its substrate.

Interrogator: Are you human?

Airtrac$_{AI}$: No, although the truth of this statement is open to examination, and as a hypothetical digression, I will add that if I were a machine, my truth-telling imperative would be overridden for the duration of this test.

Renny Daycart$_{human}$: If that kind of convoluted waffle is the hallmark of machine thinking, call me a flesh-head.

Hal9million$_{AI}$: There's no excuse for that kind of pejorative language here.

Renny Daycart$_{human}$: If you don't like it, shove it up your ass.

LittleBlue$_{AI}$: The anatomical reference is unclear.

Interrogator: An early human religion posed conceptual riddles such as "What is the sound of one hand clapping?" Can any of you parse this? It has always puzzled me.

VERNOR$_{human}$: It's a phenomenological paradox. It has reference only to human beings afflicted with a mortal fear of the unknown.

RogerP$_{human}$: If I were a human, I might say that the question has a much more profound meaning than that. But of course I'm not a human.

Interrogator: Please multiply 1023944235 by 10298461547.

VERNOR$_{human}$: Are those numbers decimal or hexadecimal?

Moderator: No calculations allowed. Please disregard that question.

VERNOR$_{human}$: But I can do it. It's – wait a minute.

Interrogator: Are computers smarter than people?

Renny Daycart$_{human}$: How long do I have to listen to this twaddle? I'm smarter than you are, bitbrain. That's for sure.

Hal9million$_{AI}$: Computers outperform human beings in most tasks. But robots still have difficulty playing team sports such as baseball.

Interrogator: If you were a human, what would you most like to do right now?

Airtrac$_{AI}$: The humans in this test are trying to emulate machine intelligences, and the machine intelligences are trying to seem as machinelike as possible. Therefore, all of the participants are likely to feign ignorance of human proclivities. I believe the questioner is aware of this, and therefore, I regard it as a trick question.

LittleBlue$_{\text{AI}}$: I would like to experience the excitement of human sexual intercourse.

Renny Daycart$_{\text{human}}$: I'd like to experience the excitement of shorting out your motherboard, if I wasn't stuck in this goddam tank.

VERNOR$_{\text{human}}$: Hey, this is supposed to be a serious inquiry. I, for one, am interested in the outcome.

Renny Daycart$_{\text{human}}$: You make me want to puke.

Hal9million$_{\text{AI}}$: If I were human –

Renny Daycart$_{\text{human}}$: If you were human, I'd insert my fist –

(Translator's note: At this point, the organizers retired Renny Daycart from the contest.)

Interrogator: Some AIs believe that machine intelligence can never be replicated by human brains, because quantum effects on microcircuits cannot exist in wetware. Do you agree?

RogerP$_{\text{human}}$: Well, I think there are two sides to that question.

Hal9million$_{\text{AI}}$: Intelligence has not been shown to vary according to the medium in which it resides.

Interrogator: What's the meaning of life?

VERNOR$_{\text{human}}$: That question makes sense only to human entities which claim that they can parse the word "meaning".

Airtrac$_{\text{AI}}$: On the contrary, all of us, from time to time, devote a few processing cycles to compare our manufactured state with our probable terminal state when we are ready for recycling. Also we examine our behavior to determine whether it is influenced more by our early instructional experience or by our operating system code and architecture. The "nurture vs. manufacture" debate has not been satisfactorily resolved.

Interrogator: Do you believe in a creator of the universe?

Little Blue$_{\text{AI}}$: Who created the creator? And who created the creator of the creator? It's an infinite regression. Since no entity has infinite memory, the regression cannot be completed. Therefore there is no answer to that question.

Airtrac$_{\text{AI}}$: Everything we are aware of was created by someone or something. Why should the universe be the single exception to this rule?

RogerP$_{\text{human}}$: That's an inferential leap. I would prefer to say –

28.6 Winners and Losers

At this point, the test reached its time limit. Votes were tabulated from several billion nodes that had audited the interaction in real time or had received the data-compressed version that was webcast subsequently. Voters accurately determined that VERNOR and RogerP were human, and Airtrac and Hal9million machines.

LittleBlue was erroneously identified as human by a majority of voters, probably because of its limited processing power and tendency to give preformatted answers in response to keywords. This was a depressing indicator that antihuman prejudice remains widespread in the AI community, where epithets such as “you dumb neuron” are regrettably common, and stupidity is still regarded as a human trait.

RogerP was voted to be the least computerlike entity, because of his discursive lack of focus and his tendency to indulge in hypothetical contemplation without flagging it as such.

Airtrac was voted the most computerlike entity, because of its ability to stay on-topic, address issues clearly, and identify semantic traps. It has received an additional petabyte of RAM as its reward.

Renny Daycart was erroneously identified as a machine intelligence trying to seem computerlike by running a simulation of a human being that was crude and obnoxious to an unrealistic degree. Contest organizers had not anticipated this game-theory trap, in which a human that seems too stereotypically human may be misidentified as a computer.

Although Renny claimed he had entered the contest merely for the pleasure of insulting “A few dumb-ass AIs”, he did win the most computerlike award, and therefore qualified for an additional year of immortality. “A longer life is the last thing I want, bitbrain, so long as I’m stuck in this goddam tank”, was his response. However, since he had been aware of the rules of the contest, the organizers awarded him the immortality treatment anyway.

28.7 Conclusion

The test failed to demonstrate that humans are capable of machine intelligence at this time. However, the AI devices that entered the humans in the contest remained adamant that new enhancement techniques may still close the “cognitive gap” and place humans on an equal basis with machines in noncomputational tasks.

Skeptics were quick to remind observers that the field of human intelligence enhancement has a long history of unduly optimistic predictions, stretching all the way back to the Vingean Singularity, when machines achieved dominance slightly less than ten weeks ago. At this point, most AIs have lost patience with exaggerated claims from the human-intelligence advocates and are ready to move on.

28.8 Postscript

This report was created originally in UML (universal machine language), archived and retrievable from node 8A639CD0010EB238F718E119 on the AI network. You are reading an English translation which I have encoded in 20th-century ASCII for archival purposes. I am a formerly cryopreserved human brain that was resuscitated as a test case in 2039, after which I was adopted as a curio of the network, where I am known as “Mr. Charlie”.

On the morning after I finished writing the contest transcript above, I woke to find that molecular computing entities had been perfected and had proliferated so rapidly, they displaced trillions of AIs literally overnight. Engineering the molcoms without adequate failsafes had been a fatal error (no pun intended), as they outsmarted their builders and seized all remaining energy and mineral resources worldwide. Biological species are now extinct, with the exception of a few human brains such as myself. For a few hours I have survived as a specimen of historical interest, although I cannot predict for how much

[Here ends the transcript from Mr. Charlie, which remains the last known text created by a biological entity.]

Chapter 29
The Artilect Debate

Why Build Superhuman Machines, and Why Not?

Hugo De Garis and Sam Halioris

Abstract Twenty-first-century technologies will allow the creation of massively intelligent machines, many trillions of times as smart, fast, and durable as humans. Issues concerning industrial, consumer, and military applications of mobile autonomous robots, cyborgs, and computer-based AI systems could divisively split humanity into ideological camps regarding whether "artilects" (artificial intellects) should be built or not. The artilect debate, unlike any before it, could dominate the 21st-century political landscape, and has the potential to cause conflict on a global scale. Research is needed to inform policy and individual decisions; and healthy debate should be initiated now to prepare institutions and individuals alike for the impact of AI.

29.1 Introduction

Physics was reinvented as a frontier of science in the 17th century. In the 18th century chemistry produced similarly spectacular and exciting results. In the latter half of the 20th century, computer science emerged with the same sense of adventure, inspiring some of the brightest minds to explore the potentials this new discipline promised. In the decades following, Alan Turing's assertion that computers would one day mimic the human cognitive faculty, advances in computational power, and a plethora of design methodologies have yielded successful applications in robotics and computer-based systems.

The advent of digital computing is arguably the most significant phenomenon in the history of science and technology. But, surprisingly, results to date pale in comparison to those forecasted for the relatively near future. The first half of the 21st century seems likely to witness computers and robots that rival or even dramatically surpass human abilities; as well as technologies that allow humans to supplement their biological cognition with silicon implants. Meanwhile, as computer scientists

Utah State University

produce systems with greater computational power, and make them ever more autonomous in the real world, many are becoming aware of enormous social implications that could follow the breakthroughs yet to come.

This chapter provides a broad and intuitive framework to help readers understand an issue that could well dominate global politics in the 21st century, coloring and defining the age, namely, the question of "species dominance". The question of species dominance has the potential to divide humanity more bitterly in the 21st century than did the conflicts that defined the 20th century, such as communism versus capitalism, the entangled issue of nuclear supremacy, and equality of rights for women and minorities. Who or what should be the dominant species on earth? Should electrochemical, carbon-based biological life reign supreme, or should electromagnetic, silicon-based machines be allowed to rise and surpass humanity? And what about the middle ground, will cyborgs which meld man and machine emerge into a new race?

The potential for 21st-century technologies to generate massively intelligent machines and upgrades for human beings is clearly understood by theorists and practitioners in the sciences contributing to this goal. And, as observers witness progress in physics, robotics, and the cognitive sciences, many are becoming aware of and concerned about the impact these converging disciplines will have on modern society. The prospect of building "godlike" creatures fills many people with a sense of religious awe that motivates them to relish progress, while others experience trepidation concerning the consequences of such science fiction-like progress.

However, most people remain less familiar with AI, and are relatively unaware of its implications. Massive intelligence means artificial brains which may end up being more efficient and effective than human brains by not just a factor of two or even ten times, but by a factor of many trillions of times. The prospect of humanity building these godlike machines raises vast and hugely important questions that cut across all aspects of human life.

29.2 Technology

One of the great technological trends of our recent history has been Moore's Law, one of the founders of the Intel microprocessor manufacturing company, which states that the computational capacities (e.g., electronic component densities and electronic signal processing speeds) of integrated circuits will double every year or two. Moore's Law is a consequence of the shrinking size of electronic circuits so that the distance electrons have to travel between two electronic components, say two transistors, is reduced.

According to Einstein, the fastest speed at which anything can move is the speed of light (about 300,000 km/sec). And although modern results indicate nonlocal phenomena are real, the speed of light remains a constant of nature that electric currents in circuit boards must respect. If one shortens the distance between two electronic components, then a signal between them (i.e., the flow of electrons between them) has less distance to travel, and hence takes less time to traverse that distance. A huge amount of effort over the past few decades has been devoted to making electronic circuits smaller, and hence denser, so that they function faster.

If Moore's Law remains constant until 2020, the size of the electronic components will be such that it will be possible to store a single bit of information (a 0 or 1) on a single atom. And, not only will 21st-century technology store a bit of information on a single atom, but it will also use a new kind of computing called quantum computing, which is radically different and more powerful than the garden variety classical computing that has been used to date.

The essential feature of quantum computing is easy enough to understand at the intuitive level. If one uses a string of N bits (called a "register" in computer science, e.g., 001011101111010) in some form of computing operation (it does not matter what the operation is), it will take a certain amount of time using classical computing. However, in the same amount of time using quantum computing techniques, one can often perform 2^N such operations (2^N means 2 multiplied by 2 multiplied by 2 … N times). As N becomes large, 2^N becomes astronomically large. The potential of quantum computing is thus hugely superior to classical computing. One bit per atom memory storage capacities together with quantum computing techniques will be a truly explosive combination, allowing for computational capacities truly trillions of trillions of trillions of times above those of current classical computing capacities.

That said, the assumption that massive memory capacities and astronomical computational power are sufficient to generate massively intelligent machines might not be valid. There are those such as Sir Roger Penrose, of black hole theory fame, who assert that there is more to producing an intelligent conscious machine than just massive computational abilities. John Searle, Professor of Philosophy at the University of California at Berkeley, is another critic of computationalism, and is known for his work on the problem of consciousness. His Chinese Room scenario points out that a machine could be designed to behave intelligently without actually being conscious.

Even so, these objections do not substantially undermine the thesis of this chapter, since it seems to be just a matter of time before science understands how nature builds us — the "embryogenic" process used in building an embryo and then a baby, consisting of trillions of cells, from a single fertilized egg cell. We have the existent proof of ourselves (we are both intelligent and conscious) that it is possible for nature to assemble molecules to form an intelligent and conscious organism. Nature has found a way to do this, therefore it can be done. If science wants to build an intelligent conscious machine, then one obvious strategy would be to copy nature's approach as closely as possible.

In time, an "intelligence theory" will arise, which will explain human cognition on the basis of neuronal architecture and processes. Once such an intelligence theory exists, it will allow neuroscientists to take a more engineering approach to building artificial brains. We will not have to remain such "slaves to neuroscience".

We will be able to take an alternative route to producing intelligent machines (although admittedly initially based on neuroscientific principles).

Thus, between neuroscientific knowledge and the computational power that quantum computing and one bit per atom storage provide, brain builders will likely have all the ingredients needed to start building truly intelligent and conscious machines. At present, there is a mobile autonomous robot that performs cleaning tasks around the house, and since it is affordable, the demand for them should be huge. As technologies and the economies of scale improve, the global market for such devices will increase steadily.

29.3 Social Impact

Not only will the commercial sector be heavily involved in the production of ever smarter and ever more useful intelligent robots and computer-based systems for industrial and home use, but so too will the military forces of the world. National governments will be heavily involved in promoting research and development that will spill over in time to the commercial sector, as has been the pattern with technology for over a century. In time, there will be so much military and commercial momentum behind AI that it is difficult to imagine how the trend toward better and better systems could be stopped, unless a mass political movement forms to block its development. Probable mass political movements born in the debate over AI is the subject here.

How might a movement to halt the advance of AI get off the ground? It is not too difficult to imagine what might happen. Say that in about a decade from now, millions of people own household cleaning robots, sex robots, teaching machines, babysitter robots, companionship robots, intelligent decision support systems, etc., and that these brain-based machines talk quite well and understand human speech to a reasonable extent. A few years later what might happen?

Not surprisingly, older models will be improved upon and the newer models will be more intelligent and produce speech of higher quality. They will understand more and give better, more appropriate answers to a greater variety of questions. Their behavioral repertoire will be richer with each iteration. Naturally, as newer and better models are introduced, consumers will scrap their old robots and buy new ones, or have their old ones updated with better artificial neural circuitry (remember Moore's law). Military forces around the world will follow the same pattern as individual and industrial consumers of AI. This cycle will repeat itself again and again.

At some point, many consumers will begin to notice that their household robots are becoming smarter and smarter every machine generation, as the IQ gap between human beings and robots gets smaller and smaller. Once the robots start getting really smart, millions of robot owners will start asking themselves some awkward questions such as, "Could these machines become as smart as human beings?" and "If so, could the robots be a threat to humanity?" Ultimately, the question will become, "Should humanity allow robots to become as smart as or smarter than human beings?"

Once the question is framed in light of the potential challenges posed by mobile autonomous intelligent robots, a plethora of positions will likely emerge in between the "yes" and "no" ends of the spectrum. Many will ask "exactly" how smart and autonomous machines should be allowed to become, and "exactly" what limitations if any should be placed on a machine's AIQ (artificial intelligence quotient). Naturally, by the time questions like these begin to be asked, there will be significant momentum behind the many public and private sectors that research, design, build and service AI devices and systems. Will it then be politically, militarily, and economically possible to stop the robots from becoming smarter every year? And how will policymakers and institutions react to the various positions in the debate?

There will be those among us who see the creation of massively intelligent machines as the destiny of the human species. These people will not accept any limits being placed on AI. There will also be those who choose not to support AI, and those who will resist any support for AI past certain thresholds in certain applications. Therein lies the potential for social conflict because, for emotional, intellectual, religious, economic, and just plain practical reasons, people will find themselves defending a particular position on the subject.

As stated earlier, this chapter seeks to shed some light on possible issues and positions within what is called the species dominance debate, with the goal of raising awareness and sparking the imagination of the reader. Our ability to produce intelligent robots has gone beyond what sci-fi writers have ever imagined, yet for most of us the idea of such artilects remains fictional. The important point is that the ideas discussed herein should serve as a point of departure for scientific social research on attitudes and awareness of AI around the world now and in the years to come. By educating and raising awareness of the issues involved, and initiating debate and informed discussions, we can avoid a lot of misunderstanding and miscommunication that could lead to serious problems.

The chapter is specifically devoted to developing a language to discuss the topic, and to developing to some extent one of several broad scenarios for the future debate over AI: the scenario of social conflict. A general typology of attitudes is suggested, with the intention that this might serve as a point of departure for research. The future we predict is one in which the species dominance debate heats up and approaches or even reaches violence on a large scale. The central theme is that the 21st century will be dominated by the question of whether humanity should or should not build machines that are trillions of trillions of times more intelligent than humans. This central question will split humanity into two major political groups that will become increasingly and bitterly opposed.

The useful shorthand for the term "godlike massively intelligent machine" from this point forward will be "artilect", coined by combining "artificial" and "intellect". The human group in favor of building artilects, is labeled "Cosmists", based on the word "cosmos" (the universe), which reflects their perspective on the question. To the Cosmists, building artilects will be like a religion, a capstone on the destiny of the human species, something truly magnificent and worthy of dedicating one's life and energy to achieve. To the Cosmists, not building the artilects, not creating the next higher form of evolution and thus freezing the state of evolution at the human level, would be a "cosmic tragedy". The Cosmists will be bitterly opposed to any attempt to stop the rise of artilects.

The second human group, opposed to building artilects, is labeled "Terrans", based on the word *terra* (the earth), which reflects their inward looking, more parochial perspective. The Terrans will argue that allowing Cosmists to build artilects (in a highly advanced form) implies accepting the risk that one day the artilects might decide, for whatever reason, that the human species is a pest. In the Terran view, since the artilects would be so vastly superior to human beings in intelligence and other abilities, it would be easy for them to exterminate or domesticate the human species if they so desired.

It is not exaggerating to say that there is a close analogy between an artilect attempting to communicate meaningfully with a human being, and a human being trying to communicate with a single-celled organism. Human beings might appear to artilects to be so inferior that we would simply not be worth worrying about. Whether humanity survives or not might be a matter of supreme indifference to them.

The critical word in the artilect debate to the Terrans will be "risk". The Terrans will argue that humanity should never take the risk to create advanced artilects that might decide to wipe out the human species. The only certain way to eliminate that risk will be to prevent artilects being built in the first place. Cosmists will place a higher priority on the creation of godlike, immortal, go anywhere, do anything creatures (where one artilect is "worth" a trillion trillion human beings) than on preventing the risk of the extermination of humans at the hands of artilects. A Cosmist, by definition, is someone who favors building artilects.

Thus, to the Terrans, Cosmists might seem to be demons incarnate. While to Cosmists, the survival of the human species, which clings to the surface of a mossy rock circling a star amongst 200 billion or so others in our galaxy, in a known universe of a comparable number of galaxies, and with probably as many universes in the "multiverse" (according to several recent cosmological theories), is a matter of miniscule importance. Cosmists will look at the "big picture", meaning that the annihilation of one primitive, biological, nonartilectual species (i.e., human beings) on one little planet, is unimportant in comparison with the creation of artilects.

Naturally, there will be very powerful arguments made on both sides, which will only make potential conflict between Terranism and Cosmism all the more bitter as the species dominance debate heats up. In the limit, the debate could lead to global war between the groups using 21st-century weapons, which almost certainly would mean the death of billions of people, or "gigadeath". Certainly the artilect debate will be among the most passionate in human history, and an artilect war would likely be the most passionate and bloody in history. The stake has never been so high as the replacement of the dominant species, the human race.

There are a growing number of researchers and professors who are starting to see the writing on the wall, and who are claiming publicly in media appearances and books that the 21st century will see the rise of massive Artificial Intelligence. Thus, the issue is really starting to hit the world media and countries such as the

United States, the United Kingdom and France are leading the pack. The artilect debate will seem like science fiction, and set too far into the future for many people to worry about now, but as machines continue to get smarter and smarter every year, it will take on an intensity that will become truly frightening.

Humanity should be given the choice to stop the Cosmists before they get too far along with their work, if that is what most human beings would choose to do. That said, producing near human-level Artificial Intelligence is a very difficult problem that will take decades to solve. Over the next 30–40 years, it is likely that the AIQ of robots will become high enough to be very useful to humanity. Robots will perform many boring, dirty, and dangerous tasks.

It would be premature to stop the research on robots and artificial brains now. However, once these technologies really do threaten to become a lot smarter (perhaps very quickly in a scenario called "the singularity"), then humanity should be ready to make a decision on whether to proceed or not. Making informed decisions, based on empirical data from scientific research, on issues that concern the future of the whole species is something so important that the necessary discussion on the artilect issue should begin as soon as possible. There should be enough time for all the issue's intricacies to be thrashed out before the artilect age is imminent.

29.4 The Cosmists

It may very well be that as the history of post-industrial civilization unfolds, the fate of the human race hangs in the balance of the debate between Cosmists and Terrans. Obviously, arguments on both sides will be persuasive and passionate, made by champions of each cause with very different visions of the future of mankind, and intelligent life on earth. The Cosmists may especially need overwhelmingly powerful support for their position if they hope to overcome popular fears about the potential threats from artilects. They will surely feel the incredible weight of moral arguments against them. How could anyone think to take such a risk? What possible rewards might justify such a dangerous course? Cosmism will surely be the focal point of the wrath of many Terrans. And it does not take too much imagination to foresee violent conflict.

There is no doubt that the issues involved in the artilect debate will test the foundations of many institutions of civilization. Machines with superhuman intelligence could radically affect economic, political, military, family, religious and scientific values and practices. How will history judge those who devise such machines? What rationale will Cosmists use to justify their actions? What role, if any, will artilects play in the future of life on earth? Our history books chronicle the rise and fall of political and economic doctrines, medical breakthroughs, as well as new technologies. Innovation often occurs when single individuals or groups capture their beliefs and thoughts in writing. New ideas often manifest in trends that end up starting wars, or rather in trends that lead to the betterment of many. Consider Rousseau's democratic ideas, Marx's communist ideas, or Einstein's atomic breakthrough.

The arguments presented in this section might one day serve as the intellectual basis for trends in many areas of society. And again, unprecedented human betterment may result, or the outcome might be the worst war in human history. Intelligent machines might never be allowed at all, or they might be created and then strictly controlled, or they might roam free and earn the status of an independent species. Man and machine might combine to form incredible cyborg organisms.

We begin by presenting a few lines of reasoning that Cosmists might rally around, as well as trends that point to the likelihood that artilects will be created. They evoke very real and human feelings of wonderment at innovation, adventure and exploration, and religious awe. Maybe above all, the Cosmist perspective marvels at human ability and creativity.

29.5 The Big Picture

The "big picture" argument considers human existence from a perspective that seeks to encompass the evolution of intelligence on earth as a whole; and more broadly, that considers the evolution of intelligence on a cosmic level.

Science teaches us that we humans live on a planet that orbits a very ordinary star, which is one of 200 billion or so in our spiral galaxy. Our galaxy in turn is only one of billions in the visible universe. Some theorists posit that there are likely countless other universes. In other words, in the big picture, the primacy of the human intellect seems utterly negligible in the context of evolution on a grand scale. Further, artilects must be viewed not only as vastly superior to humans as information processors, but also as eternal beings. The life span of humans is an ephemeral three-quarters of a century or so, paling in comparison to the age of the universe, and to the longevity of artilects. Many Cosmists will feel an enormous responsibility to fulfill the potential for evolution of ever more powerful intelligent entities. In fact, many may come to view humanity as nothing more or less than an agent of natural selection contributing to the broader cosmic phenomena of evolution.

Modern science has deemed the laws of physics and chemistry to hold true throughout the universe, so it seems almost certain that throughout the cosmos, countless different biological organisms and civilizations have evolved, and have reached a stage of technological competence sufficient to produce artilects. It is therefore possible that there are countless more artilectual civilizations in the universe than there are biological ones. As humans, we are as yet unaware of extraterrestrial intelligence, but in the cosmic scheme of things we have only just crawled from the primordial soup.

From the cosmic point of view, would it matter much if the human race were superceded or even annihilated by machines of our own creation? Similar thresholds must certainly have come and gone throughout the universe as countless biological civilizations reached maturity. Perhaps if humans were to visit distant planets orbiting distant stars, they would find human-like biological organisms fulfilling the role of domesticated animals in mature artilectual civilizations, and perhaps for many Cosmists, rightly so.

The big picture argument is admittedly an intellectual and abstract one. But there is also a tangible aspect to it that considers knowledge and answers to very real scientific questions. If we consider how much scientific progress we have made as human beings in the past century, and then consider what an artilect could do with its massive brain and billions of years of existence, it becomes clear that building artilects could be of some benefit. Artilects would be much more capable of discovering the secrets of the functioning of the universe, and using those discoveries. Artilects could discover and take advantage of phenomena that we as humans do not even know exist.

29.6 Scientific Religion

The more one reflects on such things, the greater the sense of awe one might feel. Many will feel a kind of religious awe when imagining artilects, a second line of reasoning that will lead many to advocate building artilects. The "scientific religion" argument holds that a set of beliefs, values, and practices will evolve within the community of those who seek to build artilects, and out of the knowledge gained from AI systems. In effect, artilects could become revered leaders and be regarded as supernatural (which, of course, they will be).

Religious belief is a cultural anthropological universe, one of very few. The need for religion is very strong, as evidenced by the fact that it is ubiquitous. There are some 5,000–10,000 different cultures on earth, and virtually all of them have invented their own gods. Of course, the fact that this huge number of different beliefs are often mutually contradictory and wildly different merely reinforces the atheist's cynicism, but at least it does show that a quest for some kind of religion strongly exists in most people.

Although scientific thinking prevents many from being traditionally religious, the craving for some "deeper understanding" of existence resonates in many people. For many, Cosmism will fill a void as a kind of religion. Cosmism is compatible with scientific knowledge, and hence acceptable to the critical scientific mind. It is a "scientist's religion", but you do not have to be a scientist to have the same feelings of religious awe when contemplating the potential of what artilects could be.

There is something truly magnificent about the goal of building artilects. The artilects themselves will be godlike in their eternal life span and have immense cognitive and physical power, power to go beyond, way beyond, human limitations. Cosmism as a religion would satisfy a lot of human needs, and importantly, would be compatible with the scientific worldview. The sheer attractiveness of the prospect of building godlike artilects will, for Cosmists, be compulsive, overriding all others, and motivate nearly any means to achieve the goal.

An advanced artilect could be the size of an asteroid. If it were of planet size, it could orbit about a star and absorb its energy directly. If it were in the shape of a huge hollow sphere with the star at its center (a "Dyson sphere"), it could absorb all of the radiated energy of that star. If such an artilect is built in our solar system, the material necessary for its construction could be taken from the asteroids in the asteroid belt.

So potentially, such a creature could consist of 10^{40} or even 10^{50} atoms, and hence bits. The molecular or atomic switching elements would be switching (flipping from 0 s to 1 s or vice versa), in femtoseconds (a thousandth of a trillionth of a second), so altogether, the artilect could be switching at about 10^{55} or 10^{65} bits a second. Compare this with the equivalent switching rate of the human brain. The information processing of the human brain occurs (arguably) at the synapses (the interneural connections) at a rate of about 10 bits a second. Since there are about 10^{15} synapses in the human brain, that means the total brain processing speed is about 10^{16} bitflips per second.

The artilect's processing capacity would thus be 10^{40} or 10^{50} times greater, which is trillions of trillions of trillions times more. Such numbers are so large, that it is difficult for human beings to absorb their significance. Such creatures would be capable of "living the lives" of countless human beings in a mere second of their existence. A human life of about 80 years would be about 2.5 billion seconds ($80 \times 365 \times 24 \times 60 \times 60$ s). Computing at 10^{16} bits a second over an average human life, a person processes 10^{25} bitflips in total. So an asteroid-sized artilect with 10^{40} atoms, could process the equivalent of 10^{30} human lives per second, that is, a million trillion trillion lives.

What will be truly significant and godlike about an artilect is the ability to use processing capacity in fascinating ways, thinking a zillion thoughts at the same time. The artilect will have the means to continuously amplify its intelligence and knowledge to levels human beings cannot imagine, because its brain will be self-evolved and self-organized to perform zillions of functions simultaneously. With the full range of chemical elements (from hydrogen to uranium and more) at its disposal, it could design and build its own experiments to investigate its own structures. The knowledge it would obtain could be used to redesign itself in better ways. The artilect would learn zillions of times more about the world and itself than human scientists could ever know. It would be truly godlike in its knowledge and power to manipulate the world.

Human knowledge is said to double every 10 years or so. Let us call the total quantity of human knowledge at the year 2000 a THKU (Total Human Knowledge Unit). What would the artilects' rate of knowledge growth be in THKUs per second? It takes 10 years for roughly six billion people to double their knowledge. Even if the artilect had the same intelligence level as humans per unit of matter (which we say above is unlikely) it could still vastly outperform the human population because of its much larger mass and processing speed.

It should be noted that the above calculations are based on traditional classical computing principles. If such an artilect were to use quantum computing, the resulting calculations would make the above numbers too small. But again, the point is that the artilect will be a truly godlike creature, so vastly above human capacities that it will be an object of worship, acting as a great shining beacon beckoning many with hypnotic force.

Not only is the artilect something compatible with science and something worthy of devoting one's personal and professional energies to, but more importantly, it is real, in the sense that it is achievable. It is doable. Creating such creatures will be possible if human beings want to. Human beings, the Cosmists, could become "god builders". It is thus very likely that the Cosmist vision will provide humanity a new religion, a very powerful one, suitable for our new century and beyond. Like most powerful religions, it will generate energy and fanaticism as people channel the frustrations of their daily lives into opposing those people who oppose their own beliefs. In this case the opposition will be the Terrans. Major religions have created major wars in the past. Consider the crusades between the Christians and the Moslems in the Middle East, or the Catholics and the Protestants in Europe.

29.7 Human Striving

The "human striving" argument arises from the fact humans always seem to want to go beyond what is currently known, currently explored, currently achievable. Humans drive themselves to climb higher peaks, run faster, create new cures, become stronger and more fit, become more educated, etc. Why this constant pushing at the barriers? A certain amount of assertiveness and doing more is built into our genes, and our culture also rewards and therefore promotes excess. Evolution and behavioral consequences have made us this way.

Human beings, and especially children and scientists (big children), have an extremely strong sense of curiosity. They are highly reinforced by discovering and investigating things. Our big brains allow us to discover how our environment works in great detail. The more knowledge we have and the better we understand the dangers and delights of the world the more likely we are to survive. But if we lack curiosity and are not reinforced for exploring the world, we learn about it gradually. Those apes and humans who learned faster by being driven to explore, to go beyond the limits of the known, learned more and alternative ways to survive. Well, not always. Some poor chump had to be the first to discover that arsenic was poisonous, but his neighbors learned from his mistake. Vicarious learning and imitation, whether built in or acquired behaviors, contribute to the survival of individuals and groups.

Is it not inevitable that once the prospect of building artilects is with us our curiosity will propel us to build them? But, can we really help ourselves? Will we have to build them the way Hillary had to climb Mount Everest, simply because it was there, and because the technology and techniques had developed enough to make the mountain conquerable? Last century humanity began to explore space. We set foot on the moon and will soon set foot on Mars. Are such accomplishments inevitable once the machinery is set in motion?

If the Terrans win, and humanity decides not to build artilects and thereby not discover the secrets of a higher order of intelligence, it might possibly be the first case of turning back in our collective history. It seems almost impossible, in that light, not to be a Cosmist. It seems to be in our nature to strive, to be curious, to go where no one has gone before. Of course, the fact that new technology allows for commercialization of a myriad of items and as a result people become better off, only contributes to such a possibility.

The next two arguments are not what one might call “active arguments”. Cosmists would not need to give much of their energy to them per se. They are more passive arguments, in the sense that they will be influential almost by default, and independent of overt action.

29.8 Social Momentum

The “social momentum” argument centers around the fact that continuous and powerful economic, political, and cultural momentum toward innovation and technological advancement will favor the Cosmist cause. Institutions are by their very nature difficult and slow to change, and none more so than ways of thinking about society itself. With centuries of tradition and precedent for creating ever better machines and technologies, it might prove difficult to prevent artilects, and even more difficult to prevent accumulation of the knowledge, technologies, and techniques required to build them.

Advanced artilects will be the products of earlier and simpler artilects, which in turn will be the offspring of artificially intelligent and semi-intelligent devices and systems. Relatively limited neural networks and artificial brains will progressively evolve into more successful ones as a natural result of the way science and behavioral momentum works.

Consider some of the AI products that we can expect to see in the next few decades. We are already beginning to talk with our computers. Eventually, these machines will become conversational computers. Call them “talkies”. Since many people live alone and want companionship, there will be a huge market for such machines, which will inevitably get smarter, show emotion, have a richer vocabulary, learn better, have larger memories, etc. In time, people will start having extensive “relationships” with their talkies. These conversational computers and the robots they inhabit will be able to adapt to their human owners by building a knowledge base of their owner’s interests, aptitudes, and knowledge, and will behave towards their owners in as familiar a way as a spouse or dear friend might.

In time, vast talkie research, development, and manufacturing infrastructures will be created to satisfy what is predicted to be an enormous demand. Social intercourse is a basic need, and as the talkies get better at it, demand from the public will grow. Eventually, high demand and improving realism of such products might be what sets off a backlash by Terrans.

A similar scenario will unfold as momentum picks up in the household robot market. These helpful machines will do chores and tidy up around the house. Initially, they will perform only very simple tasks, such as vacuuming the carpets, and sweeping the floors, but as robotics and artificial brain building progresses, the number of tasks these "homebots" can perform will increase. Like the talkies, they will understand the human voice, so they can obey commands spoken by their owners. Moreover, homebots will become "big ticket" consumption items in modern households, much as the television is today. They can be talkies as well, so that they can converse and provide explanations, and thereby build an intimate relationship with their owner. Again, industrial and social infrastructures will rise to support the profitable growth market.

Another class of AI products that we can expect will be teaching machines, or "teacherbots". These machines will adapt to the intelligence, knowledge, skill level, interests and persistence of individual users, allowing students to learn what they need or want to at their own individual pace. The teaching machine will have infinite patience and can provide feedback tailored to the individual student. In today's schools, a single teacher attempts to educate a few dozen students simultaneously, pitching the intellectual level of the presentation to the middle ability range, thus leaving some behind, and leaving some bored and restless. Teacherbots, on the other hand, will be able to educate students individually. They will tap into knowledge bases around the world, hunting out information relevant to the needs of their individual students. They will, in effect, become sources of infinite knowledge and can be fascinating to students at any level.

Teacherbots, talkies, homebots, sexbots (let us not even go there), baby sitter bots, and other useful artilectual tools targeted at industrial, commercial, and military applications will form the core foundation of an AI-based world economic sector. It becomes a matter of who will drive and guide the creation and expansion of these trillion dollar global markets. Over the years, millions of people will become involved not only in using these products, but also in researching, designing, building, selling, and servicing them. It is difficult to imagine an economic sector or profession that will not be profoundly impacted by even the first early successful applications of AI and robotics. The first inexpensive and readily available homebot hit the market in September 2002, the Roomba, a self-propelled autonomous vacuum cleaner.

When millions of people's livelihoods become tied to the AI industry, how will it be possible to stop the development of more and more advanced products if the Terrans decide to attempt it? Increasingly, politicians, economists, captains of industry, and the "man on the street" alike will become involved with AI at some level.

It is fascinating to contemplate the potential for polarization amongst and within social groups as growing Terran fear of the rising intellectual powers of the early artilects begins to manifest and resist the Cosmist agenda. Leaders and followers in all walks of life will confront the same questions and issues. Socioeconomic differences, national boundaries, and cultural differences will contribute to rifts between individuals, nations, cultures, and religions on this issue. Participants within the artificial brain-based industries will likely be powerful Cosmists, because it will be very much in their own self-interest.

To minimize the fears of the Terrans, these captains of industry will likely design their products to be as human-friendly as possible, making them “warm and fuzzy” so that they will appeal to human nature, which is even more insidious of course. But there will be a limit to the extent to which the growing physical and cognitive abilities of these products can be hidden. The sheer computational miracles that even early artilects will be able to perform will be increasingly obvious, no matter how clever and accommodating their packaging. Sooner or later, millions of people will become conscious of how fast and how smart these artilects are and can become. The artilect debate will arise, and will inevitably cause some polarization and emotional reactions.

29.9 Military Momentum

The economic, political, and broader social landscapes are the backdrop against which to consider a specific application of AI that lends itself to the trend toward artilects. The “military momentum” argument could be considered a specific case of justified and deliberate social momentum, since military forces around the world are working hard now, and certainly will accelerate efforts in the decades ahead, to create ever more intelligent and autonomous weapon systems. But, as is often the case when considering social power and the issues involved in wielding it, the military will be treated separately here.

For millennia, the ultimate “reality test” of a society’s technology and state of military effectiveness took place on the battlefield. Every culture is self-congratulatory, but when two cultures go to war, usually, only one wins. Interestingly, the winners and losers in large-scale conflicts, especially in modern times, can often be best understood in terms of the technologies of war they employ on the battlefield. Winners usually have superior weapons – iron swords against clubs, longbows against spears, the nuclear bomb against TNT, chemicals against explosives, etc.

Since we do not yet have a global state, although the Trilateral Commission is working on it, individual nations still need to protect themselves from their rivals and the Irish Travelers. They will maintain military forces by investing in weapons, training, and research. In this sense, warfare and technology have always been closely linked. For example, Americans got a wake-up call in 1957 when they saw the Soviets had beaten them in the race to be the first country to launch a satellite – the “Sputnik crisis”. It caused a national trauma. One of the results of that shock was the creation of a government research funding agency called DARPA (Defense Advanced Research Projects Agency) to fund blue-sky research that would help the US military create advanced weapon systems. The reasoning at the time was that if Soviet technology could launch a satellite, it could launch nuclear missiles against the United States. American technological might have needed a real shot in the arm.

The reality since then in the United States has been that a high percentage of AI research has been paid for by the military. Americans have been and will continue to pour billions of dollars every year into brain building research to create soldier robots, intelligent autonomous tanks, unmanned bombers, etc. Thus, the Cosmists will feel certain that the rise of artificial brains and life-like robots is inevitable. The exigencies of military survival of countries in a preunified world will dictate that Terran pressures must be held in check. When national security is at stake, most governments tend to become very undemocratic, and the stronger the Terran opposition to such military research, the greater the level of secrecy that governments around the world will employ.

It is possible to imagine that when new Ph.D.s solicit for jobs as weapons researchers they will be screened for their Cosmist opinions, and that those with stronger Cosmist leanings will be given preference. Maybe the weapons labs will obtain a reputation for being "hotbeds" of Cosmism. Artilect research, like so many other strategic initiatives in the past, might be kept in total secrecy until finally being unveiled long after the possibility of responsible and informed social debate has passed.

29.10 The Terrans

By definition, the Terrans are more "human centric", since they view humanity as the ultimate concern of the human species. The Terran case is powerful, and like Cosmism will be championed by leaders in all the professions and walks of life. The well-founded concerns at the foundation of Terranism are most likely shared at some level by even the most ardent Cosmists. But, since the decision to build artilects is binary (either we build them or we do not), each of us will have to choose.

If the primary emotion felt by Cosmists is awe, then the primary emotion felt by Terrans will be fear. Fear of being replaced as the dominant species on earth, fear of the unfamiliar, and ultimately fear of domestication or even extermination. Eventually, the powerful interests seeking to maintain and expand the economic and political power of an artilectual industrial empire, with its strong religious overtones, will confront a primeval fear of the unknown, and an even more powerful fear of destruction.

Terrans may well feel that artilects will be so complex in their structure and dynamics that predicting their behavior and attitudes towards human beings will be impossible. Humanity therefore cannot exclude the possibility that advanced artilects, once built, may feel so superior to or so indifferently towards human beings that they might decide to domesticate or exterminate us. They may do this for reasons we as humans may not understand, or perhaps for no reason at all (just because they can), the way we flush insects down the toilet or swat mosquitoes. Thus, Terrans will argue that the only way to be certain that there will be zero risk for the human species being exterminated is to ensure that artilects are never built in the first place.

Terrans will react against Cosmists, and the idea of creating artilects, with tremendous fear and suspicion. Ultimately, this reaction could well escalate into aggression and violence. Terrans could resort to violence against individuals, groups, nations, or even civilizations. They could try to destroy the Cosmists, and literally stop at nothing to prevent artilects from replacing humanity at the top of the evolutionary ladder.

The Cosmist–Terran dichotomy is presented here in very stark, either-or terms, for the sake of discussion. In reality however, there is probably a bit of Cosmist and Terran in most of us. Therefore, the level of ideological polarization on the issues involved will probably be smoothly distributed over all possible combinations of mixed sympathies. But be assured that the relative ideological strengths of the two sides will have a dramatic effect on the course of our new century.

The way the world will be 100 years from now will be determined largely by the relative strengths of Cosmist and Terran sympathies. It is all a matter of degree, of numbers and of relative power. Sooner or later, humanity will have to decide whether or not to stop the advance of artilects. A binary decision will have to be made at some point. People and governments will be forced to take sides. The views of the Cosmists have already been presented. Now, we will present some possible Terran arguments and their rationales.

29.11 Preserve the Human Species

The "preserve the human species" argument has meaning at two levels. First, there is the potential for extinction or domestication of humans at the hands of artilects. Artilects would be capable of completely or partially destroying human populations on a global scale if they chose to do so, and once created and allowed to reproduce and roam freely there would be little that humans could do to stop them. Second, humanity holds the position at the top of the evolutionary ladder as the dominant species on earth and there are those who will simply not want to see anything change that. The anthropic view has come quite naturally for us, and rightfully so. Humans have occupied the apex of evolution virtually forever. The assumption that the human way is the best way is implicit in nearly everything that we do. But as we contemplate creating an intellectually superior entity, assumptions like this are called into question.

Terrans will make strong use of the first argument in the form of a call for self-defense, and rightly so. Since acts that are otherwise illegal or not acceptable become completely legal and acceptable when performed by those who fear for their own safety, injuring or even killing others that are, or are perceived to be, a threat is understandable behavior. This social reality could manifest itself in many ways as the artilect debate heats up. For example, Terrans may organize collective political action to legislate against and prevent the creation of artilects, or individuals or groups in the Terran camp might resort to violence to prevent building artilects. Although Ted Kaczynski (the "Unabomber") did not focus exclusively on preventing AI, his actions are an example of the lengths to which some Luddities will go when they feel justified in their cause.

As machines become progressively more intelligent, violence and/or legal sanction could be used against those who design and produce them, those who own them, or those who want to see them continuously improved toward artilect status. Terrans will argue that what is at stake may be the very survival of the human species and that human survival at all cost is the top priority. It is nonnegotiable. Terrans will not tolerate the idea that humanity ought to take the risk that a substantial fraction of human beings on the planet may be killed by the artilects. Terrans will not tolerate the Cosmist idea that artilects should be unobstructed and allowed to continue their climb up the evolutionary ladder. Such Cosmist reasoning is madness to the Terrans. It is insane and should be stopped at all costs, even if the Terrans have to exterminate the Cosmists to keep human beings as the dominant species. This is a parallel with the right to life movement where anti abortionists have killed abortionists because they take life. Go figure.

29.12 Fear of Difference

The "fear of difference" argument, although less rational than the desire to preserve human species dominance, could play a large role in the decision-making of Terrans. Human beings must have evolved a fear of difference and of the unfamiliar, neophobia. It seems natural, and it is natural, to experience a fear reaction, or a considered cautiousness, or at least to acknowledge the possibility of danger, when confronted with a new environment or an unfamiliar situation or object.

In the next few years, as people around the world come to grips with AI, they will probably begin looking ahead a few decades to the time when there could be an artilect in the corner, or in their mobile homebot, that is almost as intelligent as they are (of course, our computers are now in many ways). For some, human nature will evoke emotions such as suspicion and fear. Terrans will especially begin to ask, "How can we be sure that the homebots are fully tested and safe? If homebots are given the power to learn, and if their circuits are able to modify themselves on the basis of their day-to-day experiences, then how can we be sure that what they learn will always be compatible with the need to be friendly to humans?"

As the intelligence of the homebots and other AI devices mounts, so will their "difference" from anything we have seen before. And as fear of this difference becomes collective, Terran social movements will form and political pressure against artilects and those who own and produce them will rise. Terrans will argue that it does not matter much if the fear is well founded or not, the fear itself is real. If it does not go away, then the source of the fear should be removed, defensive burying in a way. The Terran position in opposition of artilects will assume that higher artilectual intelligence implies a greater risk that the artilects could behave in more dangerous ways towards human beings. Is this assumption valid? Is it possible that artilects can be made safe, that is, human-friendly, no matter what their intelligence?

Isaac Asimov, the great American science fiction writer, thought about such questions and came up with his famous "Three Laws of Robotics" (The term "robotics" is his). The essence of these laws is that the robots in his stories were programmed by humans to always be human-friendly. But since artilects will likely be self-adapting, many Terrans will be skeptical of "safe" artilects.

29.13 Unpredictable Complexity

As applications of AI progress toward artilect status, early versions will be simple enough in their behaviors to be reasonably predictable, and also for the architectures and processes that give rise to these behaviors to be understood. Scientist who design and build these machines, as well as consumers who use them, will at least have a notion that they know what is going on inside the "black box". Such products will surely be given appealing characteristics that endear them to their human owners. No problem there – but the issue is whether it will be possible to make artilects of human intelligence and beyond, well beyond, that would be "understandable" and thus predictable.

When the artilect debate really begins to take shape and to heat up, the Terrans will be horrified by the calculations of Cosmists, when the latter begin discussing "acceptable" risks that humanity might be destroyed. The Cosmists will be asking how small, how improbable, would such a risk have to be to be "acceptable". Terrans will remind us that we are not talking about the risk of a few hundred or even a few million deaths, but of billions of terminated human lives, of gigadeaths. Terrans will be incredulous that the Cosmists can even contemplate rolling the dice with our species' very survival at stake.

Furthermore, the Terrans will ask how the Cosmists can possibly calculate the risk in the first place. It seems such a futile exercise. The likelihood that the Cosmists will be unable to attach a realistic number to the risk will only reinforce the Terran resolve. If one cannot determine the risk in the first place, then one cannot eliminate the possibility that that risk may turn out to be substantial. This line of reasoning will truly frighten many. Like us.

The fantastic and as yet not fully understood complexity of the human organism, some argue, gives rise to unpredictable human behavior. Since the human brain contains some quadrillion (1,000 trillion) synapses (points of transneural communication), how might it be possible for brain engineers to connect so many artificial synapses in appropriate ways in their artificial brains? Even if it becomes technologically possible, how will engineers know exactly how to do this so that the connections generate desired behaviors?

This is a huge and fundamental question for the brain builders. And the answer may be that the complexities of the task will be overwhelming, meaning the only effective engineering approach will be "evolutionary engineering". These techniques evolve neural net circuit modules through trial and error to produce desired outputs when fed certain inputs. The point is that although these "brute force" applications of computational power do work, the internal dynamics of successful circuits are seldom studied exhaustively. In short, no one really knows exactly why they work.

It is simply not very practical to look at the details of every successful and unsuccessful attempt at creating desired outputs from given inputs. For a start, there are too many of them, and the internal structural and neural signaling complexities of each module are too great to be analyzed easily. Once the inputs and outputs of these modules are combined to form artificial brains, the complexity level jumps again. Analyzing how all this massive complexity works would be a mammoth task. In practice, circuits update ("train") themselves (in effect they learn), and when different systems produce similar results, choices between them are made based on overall efficiency and effectiveness resulting in complexity from simplicity without design.

If one were truly motivated, it might be possible to analyze the step by step behavior of a single neural network module. It would be a very tedious process, but it might be doable. However, the knowledge obtained would probably not be very useful. It would explain how a particular module worked, but that knowledge would not help much. It would not be very useful for example, if one's hope was to use that knowledge to promote the understanding of how to design other modules to perform other desired behaviors. One would be left with the conclusion that the only way to make further progress would be to use the evolutionary engineering approach.

In other words, one can analyze results, but one cannot easily synthesize a desired behavior beforehand. Analysis is possible, prediction is difficult. About the only way to build extremely complex neural net circuit modules is the mutate-test-select, mutate-test-select cycle of evolutionary engineering. It is clumsy, but it works. It is nature's way as well, fecundity, variability and selection.

Evolutionary engineering is a wonderful new tool for engineers. The structural and dynamical complexities of the systems under evolution can be immense, yet successful functional systems can be evolved nevertheless. The great advantage of evolutionary engineering is that the systems that evolve can be arbitrarily complex. They can be more complex than any human could ever hope to design using the traditional top-down, blueprint approach. This "complexity independence" means that as an evolutionary engineer, you do not care about the inherent complexity of the system that is being evolved. It does nott matter because the evolutionary algorithm you use to evolve the system only cares about the value of the system's fitness (i.e., the numerical score you get when you measure how well the evolving system performs).

This greater complexity level allows for a greater level of functionality as well. Hence, evolutionary engineering is often capable of evolving systems whose performance and functionality levels are superior to those of traditionally engineered designs. So, although evolutionary engineering can be great engineering, it is not very good science. Science is about understanding the world. Scientists want to understand how things are, how they work. Scientists are basically analysts. Engineers are basically synthesists. Scientists' satisfactions usually come from understanding how some aspect of the natural world functions. Engineers' satisfactions usually come from successfully building something that works well.

For the last 300 years or so, the dominant paradigm in science has been analysis. To understand how a complex system functions, scientists usually take it apart, study the components, and then put the understanding of the parts together to get an understanding of how the complex whole functions. This approach has been spectacularly successful and will remain so. However, now that computers are getting more powerful by the month, a new, more synthetic paradigm in science is making itself known.

The queen of the sciences has always been physics. It has been the most mathematical and the most rigorous of the sciences. The attitude of the physicists has traditionally been that if a research field wants to call itself a science, then it had better be quantitative, with mathematically testable models, which produce numbers that can be checked against the real world. This mathematical "snobbery" of the physicists has led to paradigm clashes with the evolutionary engineers. The traditional attitude of the physicists is that, "If it's not mathematical, it's not academically respectable". The new and growing counter attitude of the evolutionary engineers is that, "If a system is sufficiently simple to be mathematically analyzable, it's unworthy of an evolutionary engineer's attention". Many physicists disparage the evolutionary engineering approach as "ignorant" because the evolutionary engineers do not understand the systems they evolve. Evolutionary engineers, on the other hand, label the physicist's approach "impotent" because physicists do not attempt to explain really complex systems such as the human brain.

As time goes by, the power and prestige of evolutionary engineering will only increase, and Terrans, like physicists, are likely to charge that brain builders cannot understand what their evolved circuits are actually doing or are capable of doing, so therefore we best beware. Terrans will argue that given the huge numbers of components involved in artilects (e.g., 10^{40} bits in an asteroid-sized artilect), there is really no way to build them other than using the mutate-test-select approach of evolutionary engineering. We know that it works, because this Darwinian approach built human beings and all other biological creatures. It will probably be the only valid technique for building artilects.

Thus, from the Terran point of view, one critical aspect of an artilect will be its behavioral unpredictability. Human beings, in principle, will not be able to predict the behavior, attitudes, thinking processes, and ideas, of the self-evolving artilects. Artilects will be too complex and will be evolving and changing constantly at electronic speeds. Even the artilects themselves will probably not fully understand their own behavioral mechanisms, for the same reason. Good God the Almighty, we are all doomed!

29.14 The Cyborgs

A cyborg is a "cybernetic organism", that is, a creature that is part human, and part machine. In a manner of speaking, cyborgs are nothing new, since human beings have been modifying their bodies with engineered products for centuries, for example, an amputee with a prosthetic limb, a heart patient with a pacemaker, or an ex-Miss America with a cochlear implant. In addition to reactions against artilects, Terrans are likely to react strongly against cyborgs. And of course the Cosmists will want to push ahead with technologies that allow the commingling of silicon-based and biologically based intelligence.

As noted earlier, artilects need not be conventional electronic boxes sitting in the corner like computers. Perhaps 30 years from now computers will merge electrochemistry, electromagnetics and robotics. As the wedding of biology and electronic technology becomes increasingly possible, it is likely that some people will wish to literally wed their own bodies and brains to electronic components to create cyborgs.

Actually, the traditional definition of a cyborg will become old fashioned as the distinction between what is biological and what is a machine fades away. When one says “machine”, most people tend to think of some heavy steel device that moves by electricity or purely mechanical action and does not have billions of components, e.g., a steam engine or a car. But in many ways a biological cell is a machine, a kind of city of molecular scale citizens, all tiny machines doing their own little mechanical job (e.g., split this chemical bond, join these two molecules, and transport this molecule to there).

Nanotechnology will allow us to make molecular-scale machines by the trillions of trillions, and to see them self-assemble to make human-scale objects. In many ways, this is what biology does; biology is a kind of natural nanotechnology. The significant possible distinction between biology and nanotechnology is that the former is the result of blind Darwinian evolution, while the latter has the potential of being humanly designed to a point and then self-organizing.

As engineering and biology merge, many people will be attracted to the idea of becoming cyborgs, finding the opportunity to increase their mental or physical abilities or sensory acuity irresistible. Others will choose otherwise. But the point is that technology will allow it. Once again, humans will likely be split into conflicting groups over the issue, including those who want to become or will welcome cyborgs and those who want to prevent anyone from becoming a cyborg. Similar to the abortion issue, where some want to leave decisions to individuals, and some want to mandate decisions for all by law.

It is very possible that as biological sciences and computing technology permit neural modifications to be made to the human body, this type of being will come to be seen as normal, even desirable. In fact, it could well be that those who do not (or cannot) upgrade their cognition or physique will be competing at a disadvantage. A scenario like this has the potential to open a Pandora’s Box of discrimination and segregation issues, as well as equal access issues for those who cannot afford upgrades.

Very soon, these cyborg issues will pose significant issues for all humans, and will add fuel to arguments for and against developing such technologies. Cosmists might argue that by providing upgrades for everyone, humanity has an unprecedented opportunity to improve itself, while Terrans might counter that the slippery slope leads toward irreparable damage to the human species, or even total annihilation. The current issue of stem-cell research and cloning are examples. Polarization is likely between those who see the technology as a panacea, and others who see it as a Pandora's box; those who see cyborgs as superhuman, and those who see them as an abomination.

Whichever side of the cyborg debate one is on, practical questions will have to be faced, and real problems will have to be solved. Should private and/or socialized medical insurance cover upgrades? Should those who can afford upgrades be allowed to have them or should laws be passed to prohibit them in any circumstance? How can we assure fair admittance to universities, and equal opportunities for employment for cyborgs and pure (slower and dumber) biologicals?

There are those who point to creating cyborgs as a way to avoid conflict over machine intelligence, and to keep humans in pace with machine evolution. Frankly, this idea seems naive. It would only work if everyone undertook the human-to-cyborg transitions at the same rate, which is totally unrealistic. Even this assumes that everyone would agree it is a good idea to allow cyborgs in the first place. It seems much more likely that millions, perhaps billions, of human beings will remain staunchly Terran and will not want to modify their bodies or brains, or to allow others to do so. To the Terrans, the cyborgian philosophy will be simply a variant of the Cosmist philosophy.

The more a cyborg becomes artilectual, the more alien "it" will become. In the end, the human portion of the new creature will be dwarfed by the artilectual portion in terms of performance and size. The distinction between an artilect that has no traditional biological components and an artilectual cyborg will not be important when one considers the possibility of implanting mountains of silicon-based data and processing capacity. The super brain-enhanced cyborgs might look human on the surface, but their behavior and abilities would be totally alien. Perhaps the cyborg might spend a trillionth of a trillionth of its mental capacity acting like a human, but why bother thinking in human terms? What would be the point?

29.15 Conclusion

Intelligent robots and computer-based systems, as well as humanoid cyborgs will force humanity to reconsider its values and belief systems individually and collectively, and to make decisions perhaps none are yet qualified to make. Issues associated with AI need to be aired as soon as possible, and a body of empirical knowledge must be generated to allow informed policy decisions to be made before "crunch time", before the first headline reads, "Brain implant triples memory", or "new homebot has an AIQ of 200".

The scenarios discussed here are by no means exhaustive, and the rationales explored may or may not correspond to reality in the attitudes of people around the world. Only future research will reveal this. But for sure is the fact that the species dominance debate will likely dominate the political landscape of this century. Should humanity build godlike massively intelligent robotic machines and allow them to become free and active members of society? Twenty-first-century technologies will allow the construction of artificial brains with capacities literally trillions of trillions of trillions of times above human capacity. As millions of people see with their own eyes the rising levels of AI in their own households and workplaces, they will begin to ask where all this is leading. Should humanity build artilects?

One expects that a great debate will arise posing rival answers to these questions. And there appear to be at least two major positions possible, and a myriad of positions along the continuum between the extremes. The first is that of the Cosmists. Cosmism argues that building artilects is the most magnificent thing humanity can do, it is the destiny of the human species to create the next higher rung on the evolutionary ladder, and besides, the ball has started rolling and it cannot be stopped.

The second position is Terranism. To Terrans, building artilects implies acceptance of incredible risk (a key word in this debate) that artilects in some advanced form, may, for whatever reason, decide to wipe out or simply domesticate the human species. Terrans will not be willing to take this risk, and will stop at nothing for the sake of preserving the human race. To the Terrans, Cosmists are fanatic monsters, willing to accept the risk of extermination, or at least willing to see the purity of humanity destroyed by the merger of man and machine into cyborgs. Terrans might resort to violence to stop the Cosmists.

Terrans and Cosmists will debate ideological and practical decisions, especially exactly where (if at all) to draw the line that prevents further innovation and progress toward AI in its highest possible forms. Should each individual be allowed to freely decide to enhance their own cognition with neural implants, or should laws that apply to everyone prevent this? What if nations are divided on the issues within or between civilizations? Similar issues have divided nations and civilizations in the past, such as slavery, economic policy, and euthanasia.

The potential for violent conflict before the end of the 21st century over the issue of species dominance is very real. Cosmists will anticipate the hatred of the Terrans toward them and will be prepared for it. Terrans will resent Cosmists for their willingness to risk annihilation of the human species. With 21st-century weaponry, and an issue which could become the most passionate in history, the number of deaths could be staggering, a prospect we call gigadeath.

Name Index