Hector Zenil
Editor

EMERGENCE, COMPLEXITY AND COMPUTATION
ECC

Irreducibility and Computational Equivalence

10 Years After Wolfram's A New Kind of Science

Springer

Emergence, Complexity and Computation 2

For further volumes:
http://www.springer.com/series/10624

Hector Zenil (Ed.)

Irreducibility and Computational Equivalence

10 Years After Wolfram's A New Kind of Science

Springer

Editor
Dr. Hector Zenil
Department of Computer Science
The University of Sheffield
UK

Additional material to this book can be downloaded from http://extras.springer.com

ISSN 2194-7287 e-ISSN 2194-7295
ISBN 978-3-642-35481-6 e-ISBN 978-3-642-35482-3
DOI 10.1007/978-3-642-35482-3
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012954636

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Four centuries ago, telescopes were turned to the sky for the first time—and what they saw ultimately launched much of modern science. Over the past twenty years I have begun to explore a new universe—the computational universe—made visible not by telescopes but by computers

Stephen Wolfram
A New Kind of Science, 2002

Foreword

It has been a pleasure and a privilege for me to discuss fundamental questions with Stephen over the years. In this foreword I want to indicate what I regard as some of the major contributions of *A New Kind of Science* (henceforth NKS).

In my opinion, NKS is a milestone work that will be appreciated more and more with time. Now, ten years after its publication, some things already begin to stand out.

First of all, Stephen's book is wonderfully unconventional. In an age in which there are too many papers "filling in much needed gaps" (Stan Ulam's classic put-down), who can take the time from producing a constant stream of routine papers, one that is required by the funding agencies, to write a conventional book, let alone a *magnum opus* in the grand manner?

Stephen's NKS is an example to us all, a beacon of high intellectual ambition shining through a fog of mediocrity and dispensable erudition.

I regard NKS as a fundamental work on philosophy, a work on possible worlds rather than merely about this one. Here are some of the topics that it touches on:

- **Ontology:** the world may be built of discrete mathematical structures, out of algorithms, not differential equations.
- **Epistemology:** even knowing the fundamental laws, there may be no shortcuts, no better way to deduce how a system will behave than to run it and see.
- **On randomness and pseudo-randomness:** an entirely deterministic world may seem to be random, but only be pseudo-random.
- **Digital philosophy:** what are the combinatorial bricks out of which the world is built?

Another remarkable feature of NKS is that it can be read by an intelligent high-school student, thus continuing the fine tradition of *Naturphilosophie.* The ideas are so basic that they do not require elaborate technical machinery.

Furthermore, Stephen brings the playful experimental approach of a physicist to combinatorial mathematics and theoretical computer science, fields formerly not accustomed to such playfulness.

Newton developed the calculus, which was the tool he needed for his *Principia*, and Stephen has similarly developed—and continues developing—the tool he needs, the extremely powerful algorithmic language *Mathematica*.

And NKS is a kind of combinatorics art book, full of amazingly suggestive high-resolution illustrations, reminding me of works like D'Arcy Thompson, *On Growth and Form* and Ernest Haeckel's *Art Forms in Nature*. And this in an age dominated by Bourbaki's "death to triangles!" and totally opposed to any visual images in mathematics.

As I said, I think the significance of NKS can only increase with time.

Let me illustrate this with a recent example, involving my work on "metabiology" that models biological evolution by studying the evolution of mutating software. I am attempting to unify theoretical biology and theoretical computer science based on the *ansatz* that DNA is digital software.

And having set out on this path, I immediately discover that Stephen's NKS is highly relevant. In particular, I think that the discussion of the Cambrian explosion in Stephen Jay Gould's *Wonderful Life* is illuminated by Wolfram's metaphor of *mining the computational universe*, and that the mysterious origin of life, which is actually the spontaneous discovery of software by Nature, is illuminated by Wolfram's thesis that it is easy to build a universal Turing machine out of almost any discrete combinatorial components.[1]

So NKS is a book on biology and Stephen Wolfram is a biologist! Let us celebrate the tenth anniversary of this remarkable volume. We shall be lucky to see its like again in this generation.

—**Gregory Chaitin**
Federal University of Rio de Janeiro

[1] For more on such topics, please see G. Chaitin, *Proving Darwin: Making Biology Mathematical*, New York: Pantheon, 2012.

Preface

In 2005, I began a project on mathematical logic at the then called NKS Summer School (today the Wolfram Science Summer School), a project close to my heart but rather far afield of my then interest in Computability theory. Quite surprisingly, Stephen Wolfram would listen to my stuff on neural networks, real numbers and computing at different levels of the arithmetical hierarchy with great interest, even as I myself was becoming increasingly aware that he was very skeptical about it. The project I started at Stephen's suggestion was a natural extension of the exploration of the "mathematical universe" of propositional sentences he had undertaken in A New Kind of Science (chapter 12, section 9 "Implications for Mathematics and Its Foundations") with very interesting results. Among his discoveries is the shortest single axiom equivalent to the axioms of Boolean algebra with a formula of six "Nand"'s and two variables: $((a.b).c).(a.((a.c).a)) = c$. He also found that Boolean formulas that had a name in, for example textbooks, such as the Laws of Tautology, Commutativity, Double Negation, etc., were exactly the formulas independent of (that cannot be derived from) the set of shorter formulas by number of variables and symbols. The very few exceptions, such as De Morgan's Laws, enable the shortening of proofs that would otherwise be significantly longer.

So I undertook the challenge of making predicate calculus my "mathematical universe" of investigation during the school at Brown, which eventually generated several research ideas, some of which people from Carnegie Mellon University (CMU) found interesting enough to invite me to CMU in Pittsburgh for the Spring semester of 2008 as a visiting scholar. My time at Brown and CMU ultimately led to a paper I published in Cristian Calude's festschrift under the title "From Computer Runtimes to the Length of Proofs"[1] with a potential application to waiting times useful in automatic theorem proving (and perhaps more useful in the field of verification of programs, a matter now being investigated by a student of mine). At CMU I also encountered Klaus Sutner, a researcher who had been interested in the possible connections between Wolfram's Principle of Computational Equivalence and intermediate Turing degrees almost since the publication of the ANKS book and contributor to this volume summarising for the readers some of his latest findings and ideas in this direction.

While at the NKS Summer School, as participant and then instructor, I encountered a thriving community of remarkably smart people who I hold in the highest esteem. Two of them (Bolognesi and Joosten) report some of their current and future directions in connection to NKS questions, who together with

[1] H. Zenil, From Computer Runtimes to the Length of Proofs: With an Algorithmic Probabilistic Application to Waiting Times in Automatic Theorem Proving. In M.J. Dinneen, B. Khousainov, and A. Nies (Eds.), *Computation, Physics and Beyond: Theoretical Computer Science and Applications,* WTCS 2012, LNCS 7160, pp. 223–240, Springer, 2012.

Beckage and Maymin are NKS-SS alumni contributing to this volume. My own interests changed soon after participating at Wolfram's Science Summer School, not only because of my experience with and at Wolfram but because of the insight of my PhD thesis advisor Jean-Paul Delahaye, at Lille, who immediately saw the question of hyper-computation as a "faux problème" given that the question belongs to physics and not to computer science. From then on I developed a great interest in the intersection of Wolfram's experimental techniques with another exciting area of computer science, Algorithmic Information Theory (AIT), as developed by Kolmogorov, Chaitin, Levin and Solomonoff, among others. Two of the aforementioned founders of AIT (Chaitin and Solomonoff) have been relatively close to Wolfram's NKS, attending conferences and contributing from their particular vantage points and insightful perspectives.

In 2007, Stephen Wolfram introduced me to Greg Chaitin, whom I visited at his office in the IBM TJ Watson Research Center in Yorktown, NY. Chaitin and Wolfram have been discussing about NKS and AIT for many years, both their points of possible contact and of possible stress with always interesting, and often unexpected, outcomes[2] in part motivating the beginning of my own research programme. Chaitin has an advantageous first hand view of Wolfram's work that shares with us in his foreword, once more with an unexpected, but provocative, outcome (that Wolfram is rather a biologist!). Calude, on the other hand, has approached Wolfram's ideas more recently and brings a fresh but insightful outlook of Wolfram's work in his afterword, equilibrated with references to some of the reviews of Wolfram's book in the course of the last ten years.

While the cellular automaton is among the preferred computing models in ANKS (also the model of choice for most of the contributors to this volume, see e.g. the opening chapter by Franke), the book's main concepts are the concepts of *Irreducibility* and *Computational Equivalence* that Wolfram claims to be of essential value for understanding nature and has consequently advanced in the form of guiding general principles. Cellular automata are therefore merely a means to arrive at and illustrate these central concepts. Zwirn and Delahaye contribute an interesting approach to one aspect of Wolfram's notion of irreducibility from the perspective of traditional computational complexity. Sutner investigates the formal side of the Principle of Computational Equivalence from the perspective of classical Recursion theory, also the direction of Joosten's contribution in connection to the question of the pervasiveness of complexity in nature. Rucker explores the aspect of Wolfram's ideas on computation that are more connected to the material and everyday world, comparing different systems and suggesting that both universality and undecidability are everywhere. Beckage, Kauffman, Gross, Zia, Vattay and Koliba explore the nature of computational irreducibility in connection to limits in physical and biological systems. Maymin and Velupillai connect concepts of computational irreducibility and

[2] You can find their public debates together with other personalities in the transcriptions (by Adrian German) published in: H. Zenil, *Randomness Through Computation,* World Scientific, 2011; and H. Zenil (ed), *A Computable Universe: Understanding and Exploring Nature As Computation,* World Scientific, 2013.

universality to algorithmic finance and computable economics, fields inaugurated by the authors themselves. Burgin explores the consequences for technology and engineering. Dowek, Arrighi, Bolognesi and Vidal discuss the foundations of physics from the bottom up, from particles to cosmology. Baetens and De Baets; Margenstern; Sapin; Ruivo and Oliveira; Mainzer; Martínez and Seck Tuoh Mora; and Reisinger, Martin, Blankenship, Harrison, Squires and Beavers, investigate cellular automata in connection to their computational behavioural properties, their limits and the notion of computational irreducibility. Bailey also explores the connections of Wolfram's pragmatic approaches to experimental mathematics and the discovery of formulas with the help of computers in a sort of NKS exploration fashion in traditional mathematics. Dodig-Crnkovic, Tagliabue and Bringsjord enlighten us by dissecting Wolfram's most fundamental assumptions and the philosophical consequences of his work.

Everything came together for this book at the right time. Since Wolfram's book had been published ten years previously, the authors had enough material to present and progress to report vis-à-vis new directions related to or motivated by the subjects covered in Wolfram's work. I had established connections with many of the leading researchers in the field at, among other events, the Wolfram Science conferences held in Boston, Washington and Burlington, the two Midwest NKS conferences organised by Adrian German at the University of Indiana, Bloomington in 2005 and 2008, and the JOUAL workshop held at the CNR in Pisa, Italy, organised by Bolognesi in 2009 where, among the speakers, were also contributors to this volume (Dowek, Burgin, Sapin and Dodig-Crnkovic).

I want to thank all the aforementioned individuals, and also specially to Matthew Szudzik, Joost Joosten, Todd Rowland, Catherine Boucher, Adrian German, Paul-Jean Letourneau, Genaro J. Martínez, Vela Velupillai, Gregory Chaitin, Cristian Calude, Jean-Paul Delahaye, Gordana Dodig-Crnkovic, Clément Vidal and Stephen Wolfram. Thanks also to Elena Villarreal for her continuous support and understanding, to Ivan Zelinka, the series editor of "Emergence, Complexity and Computation", who supported this project, and to Thomas Ditzinger, the publishing editor of Springer.

—**Hector Zenil**
The University of Sheffield, United Kingdom

Table of Contents

Part I

Mechanisms in Programs and Nature

Chapter 1
Cellular Automata: Models of the Physical World

Herbert W. Franke*

ZIB-Institute, Germany

Abstract. Cellular automata can be applied to simulate various natural processes, particularly those described by physics, and can also serve as an abstract model for all kinds of computers. This results in a intriguing linkage between physics and the theory of automata. Such connections prove to be suggestive in the experiment, to be described below, to apply cellular automata as models for mechanisms in the physical world. Based on such analogies, the properties of our world can be formulated in the simplest possible way. The primary focus lies not on the explicit simulation of certain laws of nature but on the general principle underlying their effects. By choice of suitable algorithms, local and causal conditions as well as random deviations can be visually rendered. In addition, the problem of determinism can be handled. Apart from the classification of computable and non-computable processes, a third category of phenomena arises, namely, mechanisms which are deterministic but not predictable. All of these characteristics of our world can be classified as aspects of some underlying structure. And, the laws of nature are apparently consistent with the evolution of a multiplicity of relatively well-defined structures.

The concept of cellular automata goes back originally to John von Neumann. The central proposition of his work was the concept of an abstract computer with universal capabilities, which could produce the blueprint of any possible computer as well as reproduce a copy of itself. The underlying question was whether, in this context, the possibility existed of self-reproduction of animate beings [28]. The idea of visualizing the distribution of instantaneous states on a graphical grid was introduced by the mathematician Stanislaw Ulam. John von Neumann's system (which contained a small error, corrected by his successors) was extremely intricate. Later, simpler solutions were discovered. For example, John Horton Conway's "Game of Life" [2] also turned out to be a cellular automaton.

The definitive advance is due to Stephen Wolfram, who proved that all of the systematic properties contained in a rectangular grid mirrored those which show up in a one-dimensional configuration, which can be represented along a single line. Wolfram had therefore identified the least complex type of cellular automaton [11, 12, 13, 14].

* I thank S.M. Blinder for help by the translation –HWF.

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 3–10.
DOI: 10.1007/978-3-642-35482-3_1

Based on their generic behavior, cellular automata can be categorized into four groups as follows.

Class 1: After a finite number of steps, a uniform homogeneous final state is reached, with all cells either empty or filled.

Class 2: Initially generated simple local patterns, sometimes changing into vertical stripes or continually recurring repetition of short cycles.

Class 3: Patterns spreading in an apparently irregular way, typical clusters evolving at intervals .

Class 4: Processes depending sensitively on a set of initial values. This might lead to behavior similar to one of the classes generated above. Sometimes, these structures are unstable and non-periodic. The automata belonging to this class will also generate laterally shifted patterns, that is oblique lines or stripes. This class possibly contains universal automata.

In all four cases, an infinite cell-space is necessary, so that the growth mechanism is unimpeded. Otherwise repetitions would necessarily be produced, sooner or later. This classification was based, more or less, on heuristic aspects; only later a parameter was found by Christopher G. Langton, which he labeled lambda, its value increasing with increasing class number. Lambda expresses quantitatively the possibility of a cell's survival in the transition to the next generation [7].

1 The Turing Machine and Gödel's Principle

Since all kinds of automata can be simulated by universal cellular automata, this also applies to Turing machines [5]. This raises the question of the connection with one of the deepest and most fundamental questions in mathematics: do unsolvable problems exist? Gödel had proved by complicated logical argumentation that there are indeed undecidable mathematical and logical problems. Now, that same proof can be carried out in a far more graphic way using Alan Turing's abstract automaton, which in its most general form also has the quality of universality. Every algorithm created to solve a problem of any kind can be simulated by a Turing automaton, and a problem turns out to be unsolvable if the output sequence does not terminate. There exist a number of deep analogies between the Gödel principle, the Turing machine, and cellular automata. Thus, fundamental principles of mathematics are equivalent to the functioning of automata and, by extension, to everything that can be simulated by them, including interactions among physical objects.

One special example is the predictability of questions that fall within the scope of logic or mathematics. There is no generally applicable procedure to determine whether a mathematical problem is solvable or not. The only way to find out is to actually construct a solution, by whatever creative means that can be applied. When you harness a Turing machine for such a problem, the sequence of steps will not be predictable in advance even if they follow one another in a deterministic manner.

2 Cellular World-Models

Cellular automata have been applied to all kinds of problems, including the elucidation of mathematical problems, the modeling of automata, and the simulation of scientific processes, such as evolutionary mechanisms [4, 1]. They have proven to be especially useful when applied to physical phenomena. Several attempts were directed towards a "digital mechanics": Ed Fredkin suggests that classical-mechanical systems are equivalent to cellular automata [8]. Cellular automata later served to simulate various types of structure-generating processes, among others, diffusion processes in fluid mechanics. Furthermore they shed some light on the formation of symmetrical patterns in natural phenomena.

The first to introduce the concept of "Rechnender Raum", or "Computational Space", was Konrad Zuse. According to his ideas, elementary particles can behave as sub-microscopic computers, interacting among themselves and thereby somehow reproducing known physical phenomena [19]. In particular, those phenomena that can be represented by differential equations are well suited for the digital modeling via cellular automata [10].

Attempts to construct direct digital models of physical processes, e.g., the propagation of waves, might appear at first sight to be clumsy and unrealistic. More promising, however, is the exploration of the fundamental ordering principles in our universe, considering the analogy between physics with its mechanisms and cellular automata. The starting-point of the argument is this: if the physical world is describable at all in mathematical terms, then the entire sequence of intermediate steps must also be modeled as cellular automaton, although possibly in a rather complex and intricate way. Certain general properties that are valid for all cellular automata must then also apply for the world as a whole [6]. At least, all those possible structures that are also implemented in the smallest cellular automaton must be present. So that, while some processes can be simulated only within certain limitations, generally valid statements can be made about the whole system of laws of nature and their interrelationships, solely by comparison with the smallest possible devices which can simulate them.

3 Locality and Causality

The algorithms for the control of cellular automata can be considered to correspond to the basic laws of physics. These are embedded in a program that prescribes how they are applied [3]. The structure and design of this program is extremely simple, not only because of the rules for a minimal number of states and functional connections, but also by keeping these rules unchanged from start to finish in a program run. This principle corresponds to the widely-accepted presumption of physicists that the basic laws of nature have not changed since the beginning of our universe. Because at every step the newly-arising distribution of values is subject to the same set of rules, the sequence of states can conceptually be regarded as an iterative process.

Temporal continuity must be analogously true, corresponding to the usual assumption of a spatial continuum. It is taken for granted that the same laws of

nature are valid everywhere in the universe. It would be quite easy to insert a local dependence into the program, but, as far as we know at present, that does not appear to be the case.

Two more evidently universal rules of physics have, from the outset, been included in the concept of the cellular automaton. By permitting only adjacent cells to influence the state of the next generation, we limit our considerations to behavior which obeys locality–there are no nonlocal effects, and each effect on one cell is mediated only by its immediate neighbors. It could be demonstrated that a kind of information transfer is feasible within cellular automata by freeing a cluster of cells from its surrounding group and setting it adrift in something like a round trip across space and time. This phenomenon corresponds to the emergence of diagonal stripes in cellular automata of the fourth type.

The same situation holds true for the time-dependent effects, which are of a strictly causal nature in the prototypical cellular automaton, and which are assumed to influence only immediately subsequent time intervals. Any effect transmitted from one cell to another thus needs the activation of all intermediate generations. These spatial and temporal adjacency rules demand that a certain cell can exert influence only within a certain limited space, and that an effect working on a certain cell can originate only within a limited space. This situation corresponds to Einstein's Light Cone, which degenerates in cellular automata into a triangle, the cell forming the starting or end point located at the top or bottom vertex. The time interval between the states, when the effects are handed on from one generation to the next, thereby behaves as an analogy to the finite speed of light.

Within classical mechanics, there arises the problem of the reversibility of events. As can easily be seen, this is normally not the case. State N+1 does not allow the reconstruction of the previous state N. In other words, different distributions in a generation can lead to exactly the same distribution in the next one. On the other hand, the algorithms can be designed so that the process will also run in the reverse direction. As Ed Fredkin has shown, this is the case if the principle of cellular automata is somewhat extended, such that not only the preceding generation, but, in addition, the antecedent of that generation are allowed to influence its successor. The simplest case is encoded by the following equations:

$$z(t) = f(t-1) - z(t-2) \tag{1}$$

Then there also exists an inverse algorithm:

$$z(t-2) = f(t-1) - z(t) \tag{2}$$

This leads to a correspondence with classical mechanics: information about the momentary place is not sufficient for calculating the subsequent state, additional information must be given about the rate of change (speed or impulse are normally used for this purpose). In this manner, by embedding the immediate as well as the remote past, the rate of change can be calculated.

4 Determinism or Randomness?

So far, we have discussed only purely deterministic examples, the course of events being immutably fixed by the starting conditions. As a result of the equivalences between cellular automata and the Turing machine, the process needs not necessarily be computable. It is conceivable that the physical processes described by the laws of nature never do come to an end, which means that we are simulating the behavior of a cellular automaton which runs deterministically, but is not computable. There are conflicting philosophical viewpoints that do not accept the influence of chance on what happens in the world. For them, determinism fulfills their belief that the world runs according to strict rules, embracing all creation and all apparent innovation, both expected and unexpected. Innovation, originating in this way, is the equivalent of chaos as understood in dynamical chaos theory, which, as we know, is based not on actual chance, but on non-computability.

The type of randomness described above has to be distinguished from that encountered in quantum theory, which is non-deterministic on a fundamental level. Most theoretical physicists will accept that, despite some disagreements about details, a final definitive answer to this question remains to be formulated in the future. But it is quite possible to test this idea with cellular automata. This can be done by introducing randomly-induced modifications–"mutations"–into the algorithms. An easier way, however, of adding an irregular interference or disturbance would consist in randomly changing the states in various places;

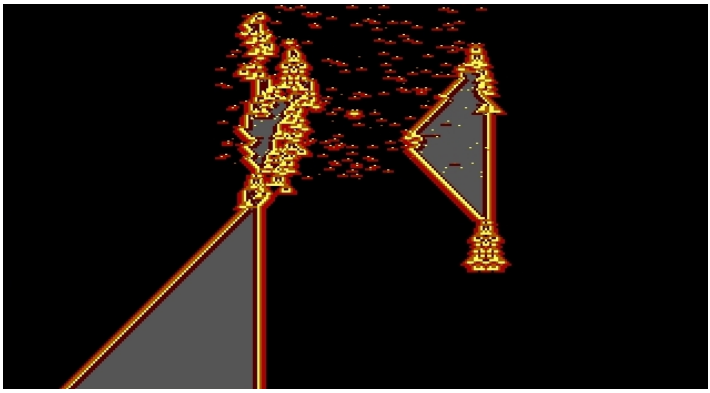
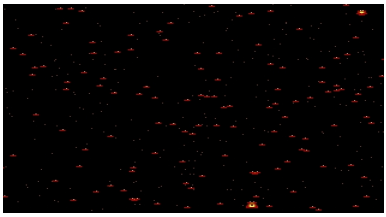
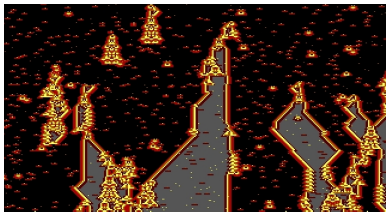


Fig. 1. For illustration, we use a cellular automaton with the two states and the transition code 0 1 2 3 4 5 / 1 1 0 1 0 0 probably of the class four type [11]. Scale of colors: 0 black, 1 bright brown, 2 bright blue, 3 yellow, 4 dark brown, 5 dark gray. The picture shows the origin of patterns on the begin of evolution, emerging of a locally and temporally limited field of chaotic distribution of initial states. As soon the evolution has reached the random free zone, the rules for the automaton produce no more shapes, but only emptiness or crystal-like order.

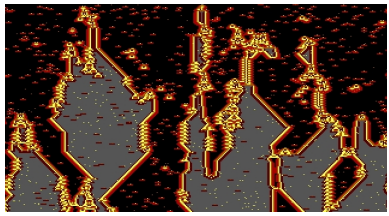


(a) 398

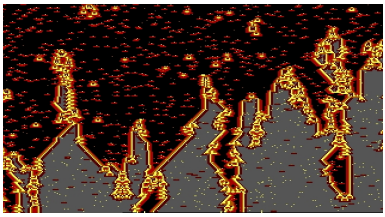
(b) 400



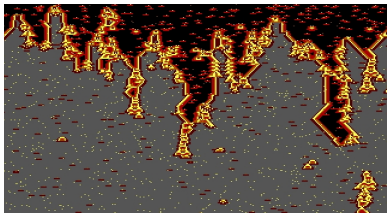
(c) 402



(d) 404



(e) 406



(f) 408

(g) 410

Fig. 2. Structures generated under an increasing influence of randomness (images followed by their density parameter). Some random-selected sites of the lines are occupied with the state 1. New shapes emerge there where such sites come in neighbourhood. By this series of pictures, the density of introduced randomness is expressed by a numerical parameter - the pictures show the situation for some parameters between 398 and 410. Randomness acts as a creative effect which counteracts against the trend towards order which the set of rules tries to maintain. A specific value of the mentioned parameter (approximately by 400) defines the status of balance between growing and destruction of structures. Such parameter gives a characteristic value for every cellular automaton.

we might say that chance could be interspersed, for example by adding a few extra lines of program code containing a randomizer.

Much information can be gained from a comparison between repeated runs of the same cellular automaton with and without disturbance (see figures). As can be seen, by the application of a disturbing element, an antagonism, a competition between order and disorder is triggered. There are cellular automata that obviously possess a strong trend towards expressing their repertoire of patterns, and thus easily suppressing all germs of chaos. On the other hand, there are others in which even a slight touch of randomness suffices to "lead them astray" or make them run out of control, so that a great multiplicity of different patterns is generated. A Class One linear cellular automaton requires a strong dose of randomness to get its regularity disturbed, but, all the same, the previous pattern will soon be re-established. In Class Three automata, by contrast, a minimal disturbance is enough to render impossible a return to a homogeneous pattern; the chances are that random effects generate nuclei of larger well-ordered clusters.

What is visually expressed in the illustrations can also be seen as aspects of information or complexity: irreversible and deterministic automata run in a manner in which complexity can never increase, but in most cases must inevitably decrease. As a consequence, the patterns get more and more simple, they degenerate into cyclic sequences that fill the whole available space or vanish completely. Only reversible processes retain their complexity, and innovation emerges, if at all, via re-ordering as understood by the deterministic modification of chaos theory. The formation of complexity then becomes possible only in stochastic models.

The structural variety of our world as we experience it might also spring from a deterministic model without the influence of chance if that model belongs to the category of undecidable mathematical problems. Since, however, a decisive answer on whether this is the case cannot possibly be given, since there would always exist the alternative that such a world will sooner or later turn into crystal-like rigidity or dissolve into chaos, possibly in the sense of the Heat Death of the universe. This kind of world is philosophically unsatisfying, but it is not our option to choose in what sort of world we actually live. It is quite informative to view it from a completely different point of view, asking ourselves: How must a universe be built that will keep its structure-creating capability forever and with certainty? The best solution is an endlessly running cellular automaton modified with that touch of randomness that conforms with its innate trend towards regularity.

5 Conclusions

To sum up, cellular automata turn out to be possible models to visualize the basic structure of our world. By reducing that structure to the least complex programs, they enable us to deal in a more definite way with various relevant problems, including those with philosophical implications–near and far effects, causality, determinism, and entropy. A new multiplicity of problems, triggered

not in the least by chaos theory, is that of the formation of structures, as this capability inherent in nature is doubtlessly of fundamental relevance. Preeminent in this context is the role of randomness, an issue since the early days of the quantum theory. To identify our universe as a Class Four cellular automaton is at present no more than a speculation, but in future considerations involving this class of problems it will have to be considered as a promising candidate.

References

[1] Codd, E.F.: Cellular Automata. Academic Press, New York (1968)
[2] Berlekamp, E., Conway, J., Guy, R.: Gewinnen– Strategien für mathematische Spiele. Vieweg, Braunschweig (1985)
[3] Franke, H.W.: Die Welt als Programm. Naturwissenschaftliche Rundschau 45, 379 (1992)
[4] Gerhard, M., Schuster, H.: Das digitale Universu, Zelluläre Automaten als Systeme der Natur. Vieweg, Braunschweig (1995)
[5] Herken, R. (ed.): The Universal Turing Machine. A Half-Century Survey. Kammerer und Unverzagt, Hamburg (1988)
[6] Hedrich, R.: Komplexe und fundamentale Strukturen. BI Wissenschaftsverlag, Mannheim, Wien, Zürich (1990)
[7] Langton, C.G.: Life at the Edge of Chaos. In: Langton, C.G. (ed.) Artificial Life II. Addison-Wesley, Redwood City (1991)
[8] Margolus, N.: Physics-like Models of Computation. In: Farmer, D., Toffoli, T., Wolfram, S. (eds.) Cellular Automata, Physica D, vol. 10 (1984)
[9] von Neumann, J.: The Theory of Self-reproducing Automata. In: Burks, A.W. (ed.) Essays on Cellular Automata. University of Illinois Press, Urbana (1970)
[10] Wunsch, G.: Zellulare Systeme. Vieweg, Braunschweig (1977)
[11] Wolfram, S.: Cellular Automata as Models for Complexity. Nature 311, 419 (1984)
[12] Wolfram, S.: Universality and Complexity in Cellular Automata. In: Farmer, D., Toffoli, T., Wolfram, S. (eds.) Cellular Automata. North-Holland (1984)
[13] Wolfram, S.: Theory and Applications of Cellular Automata. World Scientific Publishing Company, Singapore (1986)
[14] Wolfram, S.: A New Kind of Science. Wolfram Media (2002)
[15] Zuse, K.: Rechnender Raum. Vieweg, Braunschweig (1969)

Chapter 2
On the Necessity of Complexity

Joost J. Joosten

Department of Logic, History and Philosophy of Science
Faculty of Philosophy
Carrer Montalegre 6, 08001 Barcelona, Spain
jjoosten@ub.edu

Abstract. Wolfram's Principle of Computational Equivalence (**PCE**) implies that universal complexity abounds in nature. This paper comprises three sections. In the first section we consider the question *why* there are so many universal phenomena around. So, in a sense, we seek a driving force behind the **PCE** if any. We postulate a principle **GNS** that we call the *Generalized Natural Selection* principle that together with the Church-Turing thesis is seen to be equivalent in a sense to a weak version of **PCE**. In the second section we ask the question why we do not observe any phenomena that are complex but not-universal. We choose a cognitive setting to embark on this question and make some analogies with formal logic.

In the third and final section we report on a case study where we see rich structures arise everywhere.

1 Why Complexity Abounds

Throughout the literature one can find various different and sometimes contradicting definitions of what complexity is. The definition that we shall employ in the first section involves the notion of a universal computational process/device. For the second and third section of this paper we shall use slightly less formal and more relaxed notions of the word complexity.

1.1 What Is Complexity?

Let us recall that a computational process Π is universal if it can simulate any other computational process Θ. In other words, Π is universal if for any other computational process Θ, we can find an easy coding protocol $\mathcal{C}$ and decoding protocol $\mathcal{C}^{-1}$ so that we can encode any input x for Θ as an input $\mathcal{C}(x)$ for Π so that after Π has performed its computation we can decode the answer $\Pi(\mathcal{C}(x))$ to the answer that Θ would have given us. In symbols: $\mathcal{C}^{-1}(\Pi(C(x))) = \Theta(x)$.

For the sake of this first section we can take as working definition that a system is complex if we can easily perceive it as a universal computational process. Note that we have used the word 'easy' a couple of times above. If we were to be more precise we should specify this and could for example choose for *poly-time* or some

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 11–23.
DOI: 10.1007/978-3-642-35482-3_2

other technical notion that more or less covers the intuition of what is easy. We wish to be not too specific on these kind of details here.

Thus, for the first section of this paper, a complex process is one that is computationally universal. However, great parts of the reasoning here will also hold for other definitions of complexity. For example, stating that a process is complex if comprehending or describing it exceeds or supersedes all available resources (time, space, description size).

Note that our current definition of complexity need not necessarily manifest itself in a complex way. Remember that a universal process is one that can simulate *any* other process, thus also including the very simple and repetitive ones. One might thus equally well observe a universal process that temporarily exhibits very regular behavior. In this sense universal cannot be directly equated to our intuitive notion of complexity but rather to *potentially complex*.

1.2 The Principle of Computational Equivalence and the Church-Turing Thesis

In his NKS book [10], Wolfram postulates the Principle of Computational Equivalence (**PCE**):

> **PCE:** Almost all processes that are not obviously simple can be viewed as computations of equivalent and maximal sophistication.

The processes here referred to are processes that occur in nature, or at least, processes that could in principle be implemented in nature. Thus, processes that require some oracle or black box that give the correct answer to some hard questions are of course not allowed here.

As noted in the book, **PCE** implies the famous Church-Turing Thesis (**CT**):

> **CT:** Everything that is algorithmically computable is computable by a Turing Machine.

In Section 3 below we shall briefly revisit the definition of a Turing Machine. Both theses have some inherent vagueness in that they try to capture/define an intuitive notion. While the **CT** thesis aims at defining the intuitive notion of algorithmic computability, **PCE** aims at defining what degrees of complexity occur in natural processes. But note, this is not a mere definition as, for example, the notion of what is algorithmically computable comes with a clear intuitive meaning. And thus, the thesis applies to all such systems that fall under our intuitive meaning. As a consequence, the **CT** thesis would become false if some scientists were to point out an algorithmic computation that cannot be performed on a Turing Machine with unlimited time and space resources. With the development and progress of scientific discovery the thesis has to be questioned and tested time and again. And this is actually what we have seen over the past decades with the invention and systematic study of new computational paradigms like DNA computing [11], quantum computing [10], membrane computing [4], etc. Most scientists still adhere to the **CT** thesis. There are some highly theoretical

notions of super-computations and super-tasks which would allegedly escape the **CT** thesis but to my modest esteem, they depend too much on very strong assumptions and seem impossible to be implemented [12]. However, I would love to be proven wrong in this and see such a super-computer be implemented.

In the **PCE** there is moreover a vague quantification present in that the principle speaks of *almost all*. This vague quantification is also essential. Trying to make it more precise is an interesting and challenging enterprise. However, one should not expect a definite answer in the form of a definition here. Rather, I think, the question is a guideline that points out interesting philosophical issues as we shall argue in Section 2. Moreover, these vague quantifiers could be read in other parts of science too. For example, the Second Law of Thermodynamics tells us that all isolated macroscopic processes in nature are going in the direction that leads to an increase of entropy. First of all, it is per definition not possible to observe perfectly isolated macroscopic processes. So, in all practical applications of the Second Law of Thermodynamics we would have to comfort ourselves with a highly isolated macroscopic process instead. And then, we know, as a matter of fact that we should read a vague quantifier to the effect that *almost all* such processes lead to an increase of entropy. A notable exception is given by processes that involve living organisms. Of course one can debate here to what extent higher-level living organisms can occur in a highly isolated environment. But certainly lower-level living organisms like colonies of protozoans can occur in relative isolation thus at least locally violating the Second Law of Thermodynamics. (See [2] and [13] for different viewpoints on whether life violates the Second Law of Thermodynamics.)

It has been observed before in [10] that the **PCE** does imply the **CT**. Note that **PCE** quantifies over all processes, be they natural or designed by us. Thus in particular Turing Machines are considered by the **PCE** and stipulated to have the maximal degree of computational sophistication which implies the **CT** thesis.

But the **PCE** says more. It says that the space of possible degrees of computational sophistication between obviously simple and universal is practically un-inhibited. In what follows we shall address the question what might cause this. We put forward two observations. First we formulate a natural candidate principle that can account for **PCE** and argue for its plausibility. Second, we shall briefly address how cognition can be important. In particular, the way we perceive, interpret and analyze our environment could be such that in a natural way it will not focus on intermediate degrees even if they were there.

We would like to stress here that intermediate degrees refer to undecidable yet not-universal to be on the safe side. There are various natural decidable processes known that fall into different computational classes like P-time and EXP-time processes which are known to be different classes.

In theoretical computer science there are explicit undecidable intermediate degrees known and the structure of such degrees is actually known to be very rich. However, the processes that generate such degrees are very artificial whence unlikely to be observed in nature. Moreover, although the question about the

particular outcome of these processes is known to yield intermediate degrees, various other aspects of these processes exhibit universal complexity.

1.3 Complexity and Evolution

In various contexts but in particular in evolutionary processes one employs the principle of Natural Selection, often also referred to as Survival of the Fittest. These days basically everyone is familiar with this principle. It is often described as species being in constant fight with each other over a limited amount of resources. In this fight only those species that outperform others will have access to the limited amount of resources, whence will be able to pass on its reproductive code to next generations causing the selection.

We would like to generalize this principle to the setting of computations. This leads us to what we call the principle of Generalized Natural Selection:

> **GNS:** In nature, computational processes of high computational sophistication are more likely to maintain/abide than processes of lower computational sophistication.

If one sustains the view that all natural processes can be viewed as computational ones, this generalization is readily made. For a computation, to be executed, it needs access to the three main resources space, matter, and time. If now one computation outperforms the other, it will win the battle over access to the limited resources and abide. What does outperform mean in this context?

Say we have two neighboring processes Π_1 and Π_2 that both need resources to be executed. Thus, Π_1 and Π_2 will interfere with each other. Stability of a process is thus certainly a requirement for survival. Moreover, if Π_1 can incorporate, or short-cut Π_2 it can actually use Π_2 for its survival. A generalization of incorporating, or short-cutting is given by the notion of simulation that we have given above. Thus, if Π_1 can simulate Π_2, it is more likely to survive. In other words, processes that are of higher computational sophistication are likely to outperform and survive processes of lower computational sophistication. In particular, if the process Π_1 is universal, it can simulate any other process Π_2 and thus is likely to use or incorporate any such process Π_2.

Of course this is merely a heuristic argument or an analogy rather than a conclusive argument for the **GNS** principle. One can think of experimental evidence where universal automata in the spirit of the Game of Life are run next to and interacting with automata that generate regular or repetitive patterns to see if, indeed, the more complex automata are more stable than the repetitive ones. However one cannot expect of course that experiments and circumstantial evidence can substitute or prove the principle.

Just like the theory of the selfish gene (see [5]) shifted the scale on which natural selection was to be considered, now **GNS** is an even more drastic proposal and natural selection can be perceived to occur already on the lowest possible level: individual small-scale computational processes.

We note that **GNS** only talks about computational processes in nature and not in full generality about computational processes either artificial or natural as

was the case in **PCE**. Under some reasonable circumstances we may see **GNS** as a consequence of **PCE**. For if ¬ **GNS** were true, there would be no complex processes to witness after some time and this contradicts **PCE**. Thus we have:

$$\mathbf{PCE} \implies \mathbf{CT} + \mathbf{GNS}.$$

As we already mentioned, **GNS** only involves computational processes in nature. Thus we cannot expect that **CT**+**GNS** is actually equivalent to **PCE**. However, if we restrict **PCE** to talk only about processes in nature, let us denote this by **PCE′**, then we do argue that we can expect a correspondence. That is:

$$\mathbf{PCE}' \approx \mathbf{CT} + \mathbf{GNS}.$$

But **PCE′** tells us that almost all computational processes in nature are either simple or universal. If we have **GNS** we find that more sophisticated processes will outperform simpler ones and **CT** gives us an attainable maximum. Thus the combination of them would yield that in the limit all processes end up being complex. The question then arises, where do simple processes come from? (Normally, the question is where do complex processes come from, but in the formal setting of **CT**+**GNS** it is the simple processes that are in need of further explanation.)

Simple processes in nature often have various symmetries. As we have argued above these symmetries are readily broken when a simple system interacts with a more complex one resulting in the simple system being absorbed in the more complex one. We see two main forces that favor simple systems.

The first driving force is what we may call *cooling down*. For example, temperature/energy going down, or material resources growing scarce. If these resources are not available, the complex computations cannot continue their course, breaking down and resulting in less complex systems.

A second driving force may be referred to as *scaling* and invokes mechanisms like the Central Limit Theorem. The Central Limit Theorem is a phenomenon that creates symmetry by repeating a process with stochastic outcome a large number of times yielding the well-known Gaussian distribution. Thus the scale (number of repetitions) of the process determines the amount of symmetry that is built up by phenomena that invoke the Central Limit Theorem.

In the above, we have identified a driving force that creates complexity (**GNS**) and two driving forces that creates simplicity: cooling down and scaling. In the light of these two opposite forces we can restate **PCE′** as saying that simplicity and universality are the two main attractors of these interacting forces.

Note that we deliberately do not speak of an equivalence between **PCE′** and **CT** + **GNS**. Rather we speak of a correspondence. It is like when modeling the movement of a weight on a spring on earth. The main driving forces in this movement are gravitation and the tension of the spring. However, this does not fully determine a final equilibrium if we do not enter in more details taking into account friction and the like. It is in the same spirit that we should interpret the above mentioned correspondence.

However, there are two issues here that we wish to address. First, we have argued that **CT** + **GNS** is in close correspondence to **PCE′** which is a weak

version of **PCE**. What can be said about full **PCE**? In other words, what about those processes that we naturally come up with? There is clearly a strong cognitive component in the study of those processes that we naturally come up with.

Second, **PCE** has a strong intrinsic implicit cognitive component as it deals with the processes that we observe in nature and not necessarily the ones that are out there. Admittedly, in its original formulation there is no mention of this cognitive component in **PCE** but only the most radical Platonic reading of **PCE** would deny that there is an intrinsic cognitive component present.

We shall try to address both issues in the next section where we discuss how cognition enters a treatment of **PCE**.

2 Cognition and Complexity

In the first section we used a robust definition of complexity by saying a process is complex if we can easily perceive it as a universal computational process. In the current section we shall deliberately use a less formal and rigorous definition.

2.1 Relative Complexity

In the current section we say that a process is complex if comprehending or describing it exceeds or supersedes all available resources (time, space, description size). In doing so the relative nature of complexity becomes apparent.

The relativity is not so much due to our underspecification when we spoke of comprehension or description of a process. One can easily think of formal interpretations of these words. For example, comprehension can be substituted by obtaining a formal proof in a particular proof system. Likewise, descriptions can be thought of as programs that reproduce or model a process. However for the sake of the current argument it is not necessary to enter into that much detail or formalization.

The relativity of the notion of complexity that we employ in this section is merely reflected in how much resources are available. A problem or process can be complex for one set of resources but easy for another.

For example, let us consider the currently known process/procedure Π that decides whether or not a natural number is prime (see [1]). If we only have quadratic time resources, then $\Pi(n)$ is difficult as the current known procedure is known to require an amount of time in the order of $|n|^{12}$ (that is, in the order of magnitude of the length of n (written in decimal notation) to the power 12). Of course, if we allow polynomial time, then primality is an easy problem.

This relativity is a rather trivial observation. The point that we wish to make here however is more subtle and profound. So, we depart from the observation that complexity is always relative to the framework in which it is described/perceived. Now, the ultimate framework where all our formal reasoning is embedded is our own framework of cognitive abilities. And this has two important consequences.

Firstly, it implies that if we study how our cognitive framework deals with complexity and related notions, we get a better understanding of the nature of the definitions of complexity that we come up with. And secondly, it strongly suggests that various notions and definitions of complexity in various unrelated areas of science in the end are of the same nature. Thus, various formal theorems that relate different notions of complexity, like ergodicity, entropy, Kolmogorov-Chaitin complexity, computational complexity etc. are expected to be found. And as a matter of fact, in recent developments many such theorems have been proven. In the final section of this chapter we shall see a new such and rather unexpected connection between two seemingly different notions of complexity: fractal dimensions versus computational runtime classes.

2.2 Cognitive Diagonalization

In this section we wish to dwell a bit on the following simple observation: as human beings we have a natural ability to consider a system in a more complete and more complex framework if necessary. We shall give some examples of this in a formalized setting and pose the question how we naturally generate more complex systems in a cognitive and less formal setting.

Suppose we study a system $\mathcal{S}$ with just an initial element –let us call that 0– and an operator S that gives us a next, new and unique element which we call the successor. The smallest system of this kind can be conceived as the natural numbers without further extra structure on them:

$$\{0, S0, SS0, SSS0, \ldots\}.$$

If we study this system in a fairly simple language it turns out that all questions about this systems are easily decidable by us.

Of course, we would not leave it there and move on to summarize certain processes in this system. The process of repeating taking the successor a number of times is readily introduced yielding our notion of addition defined by $x+0 = x$ and $x + Sy = S(x + y)$. So, by summarizing certain processes in $\mathcal{S}$ we naturally arrive at a richer structure $\mathcal{S}'$ whose operations are S and $+$.

If we now study $\mathcal{S}'$ in the same simple language as we used for $\mathcal{S}$ but now with the additional symbol $+$, we see that all questions are still decidable. That is, we can still find the answer to any question we pose about this system in an algorithmic way. The time complexity of the algorithm is slightly higher than that of $\mathcal{S}$ but the important thing is that it is still decidable.

By a process similar to by which we went from $\mathcal{S}$ to $\mathcal{S}'$ we can now further enrich our structure. We consider repeated addition to get to our well-known definition of multiplication: $x \times 0 = 0$ and $x \times Sy = x \times y + x$. The resulting structure $\mathcal{S}''$ has now three operations: $S, +$ and $\times$. However questions about this structure in this language now turn out to be *undecidable*. That is, there is no single algorithm that settles all questions about this structure in our simple language.

The process by which we went from $\mathcal{S}$ to the more complex system $\mathcal{S}'$ and from $\mathcal{S}'$ to the more complex system $\mathcal{S}''$ is called iteration. One may think that this can

only be repeated ω many times but we shall now describe a more general process of gaining complexity which is called *diagonalization* and of which iteration is just a special case. As an illustration of how this works we shall give a proof of Gödel's First Incompleteness Theorem.

> **Gödel's First Incompleteness Theorem** For each algorithmically enumerable theory T that only proves true statements and that is of some minimal strength there is a true sentence φ_T that is not provable in T.

Although Gödel had a slightly different formulation of his First Incompleteness Theorem in essence it is the one that we shall prove here. Our proof will focus on the computable functions $f(x)$ that T can prove to be total. Thus, we focus on those unary functions f which are computable and moreover, so that T proves that f is defined for each value of x. We shall write

$$T \vdash \forall x \, \exists y \; f(x) = y$$

for the latter. As T is algorithmically enumerable, we can fix an enumeration and just enumerate the proofs of T and stick with all the proofs π_i that prove some computable function f_i to be total. Once we have a way to make this list of functions f_i we readily come up with a new computable function f' which is total but not provably so by T. We construct f' by what is called diagonalization and it will soon become clear why this is called this way. We can make a table of our functions f_i with their values where we in anticipation have displayed the diagonal in boldface.

$\mathbf{f_0(0)}$	$f_0(1)$	$f_0(2)$	$f_0(3)$	$\ldots$
$f_1(0)$	$\mathbf{f_1(1)}$	$f_1(2)$	$f_1(3)$	$\ldots$
$f_2(0)$	$f_2(1)$	$\mathbf{f_2(2)}$	$f_2(3)$	$\ldots$
$f_3(0)$	$f_3(1)$	$f_3(2)$	$\mathbf{f_3(3)}$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

We now define $f'(x) = f_x(x) + 42$. Clearly f' differs from any f_i as it differs on the diagonal. However, f' is clearly a total function and there is also an easy algorithm to compute it: to compute $f(x)$ we enumerate, using the fixed enumeration of the theorems of T, all proofs of T until we arrive at π_x. Then we compute $f_x(x)$ and add 42 to the result.

To summarize, we have provided a total computable function that is not proven to be total by T whence T is incomplete. It is clear what minimal requirements should be satisfied by T in order to have the proof go through.

For the main argument of this paper this proof of Gödel's First Incompleteness Theorem is not entirely necessary. We have included it for two main reasons.

Firstly, of course, there is the beauty of the argument which is a reason for itself. And second, the proof illustrates nicely how diagonalization works.

In mathematical logic diagonalization is a widely used technique and a universal way to obtain more complex systems. An interesting and important question is, is there a natural cognitive counterpart of this? So, is there some sort of universal cognitive construct –cognitive diagonalization if it were– that always yields us a more complex framework in which to study a system. For it is clear that we tend to add complexity to systems that we build and perceive until it reaches the boundaries of our abilities. And thus we pose the question if there is some universal and natural way by which we add this complexity.

As an academic exercise one could try to rephrase diagonalization in a setting of formalized cognition but that would yield a very artificial principle. Moreover this will say nothing about what we actually do in our heads.

2.3 Cognition, Complexity and Evolution

Fodor has postulated a principle concerning our language. It says that (see [6]) the structure and vocabulary of our language is such that it is efficient in describing our world and dealing with the frame problem. The frame problem is an important problem in artificial intelligence which deals with the problem how to describe the world in an efficient way so that after a change in the state of affairs no entirely new description of the world is needed.

On a similar page, we would like to suggest that our cognitive toolkit has evolved over the course of time so that it best deals with the processes it needs to deal with. Now, by **PCE** these processes are either universal or very simple. Thus, it seems to make sense in terms of evolution to have a cognitive toolkit that is well-suited to deal with just two kinds of processes: the very simple ones and the universal ones.

Thus, it could well be that there actually are computational processes out there that violate **PCE** just as there are chemical processes (life) that locally violate the Second Law of Thermodynamics but that our cognitive toolkit is just not well-equipped enough to deal with them.

This might also be related to the question we posed in the previous section: how do we add complexity to a system? Let us continue the analogue with formal logic. Diagonalization is currently the main universal tool for adding strength to a system. However, there are various indications that for many purposes diagonalization seems not to be fine-grained enough and some scientists believe this is one of the main reasons why we have such problems dealing with the famous **P** versus **NP** problem. Likewise, it might be that cognitive diagonalization is not fine-grained enough to naturally observe/design intermediate degrees.

3 Complexity Everywhere: Small Turing Machines

In this final section I will report on an ongoing project jointly with Fernando Soler-Toscano and Hector Zenil. In this project we study the structures that

arise when one considers small Turing machines. Here, in this final section we relax the working definition of complexity even further to just refer to interesting structures.

In 2009 I attended the NKS summer school led by Stephan Wolfram in Pisa, Italy. One of the main themes of NKS is that simple programs can yield interesting and complex behavior. Being trained as a mathematician and logician this did not at all shock my world view as there are various simple functions or axiomatic systems known that yield very rich and complex structures. However, when you start delving the computational universe yourself it is that you get really excited about the NKS paradigm. It is not merely that there are various interesting systems out there, it is the astonishing fact that these systems abound. And wherever you go and look in the computable universe you find beautiful, intriguing and interesting structures. In this final section I shall report on one of those explorations in the computational universe.

The set-up of our experiment was inspired by an exploration performed in [10] and we decided to look at small Turing-machines. There are various definitions of Turing machines in the literature which all look alike. For us, a Turing machine (TM) consist of a tape of cells where the tape extends infinitely to the left and is bounded to the right. Each cell on the tape can be either black (1) or white (0) and this start configuration is specified by us. There is a *head* that moves over the tape and as it does so, the head can be in one of finitely many *states* (like states of mind).

We have now specified the hardware of a TM. The software, so to say, of a TM consists of a lookup table. This table tells the head what to do in which situation. More concrete, depending on the state the head is in and depending what symbol the head reads on the cell of the tape it is currently visiting, it will perform an action as specified by the lookup table. This action is very simple and consist of three parts: writing a symbol on the cell it currently is at, moving the head one cell left or right and going to some state of mind.

We only looked at small Turing machines that have either 2, 3 or 4 states of mind. On those machines we defined a computation to start at the right-most cell of the tape in State 0. We say the computation halts when the head 'drops off' at the right-hand side of the tape. That is, when it is at the border cell of the tape and receives a command to go one cell to the right. We fed these TMs successive inputs that were coded in unary plus one. Thus, input 0 was coded by just one black cell, input 1 was coded by two consecutive black cells, and input n was coded by $n + 1$ consecutive black cells on an otherwise white tape.

With this set-up we looked at the different functions that were computed by these small TMs and had a particular focus on the runtimes that occurred. Of course, there are various fundamental issues to address that are mostly related to either the Halting Problem (there is no algorithm that decides whether a TM will halt on a certain input) or unfeasibility. Some of these issues are addressed in [15].

When plotting the halting probability distribution for our TMs we verified a theoretical result by Calude to the effect that most TMs either halt quickly

or they never halt at all ([6]). Although this result was expected we did not expect the pronounced phase-transitions one can see in Figure 1 in the halting probability distributions that we found. In a sense, these phase transitions are rudimentary manifestations of the low-level complexity classes as described in [8].

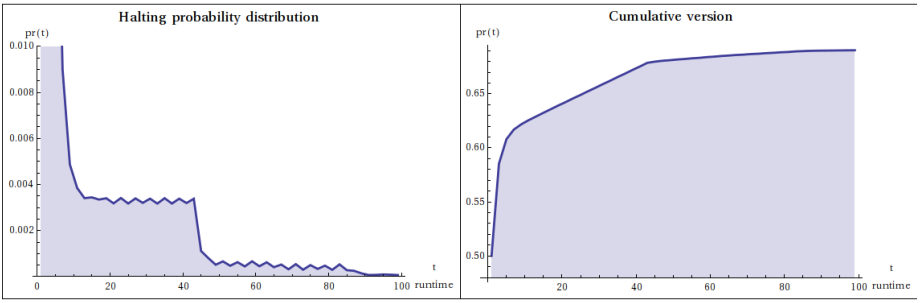


Fig. 1. Halting time distribution among TMs with three states and two colors on the first 21 inputs

Another striking feature that we found is that TMs tend to grow *slower* if you give them more resources. Let us make this statement more precise. We studied the behavior of all 4,096 TMs with two colors and two states (we speak of the (2,2)-space). In total, they computed 74 different functions. We also studied the behavior of all the 2,985,984 TMs with two colors and three states where now 3,886 different functions were computed. Any function that is computed in the (2,2)-space is easily seen to be also present in (3,2)-space. We looked at the time needed to compute a function in the different spaces. To our surprise we saw that almost always slow-down occurs. And at all possible levels: slow-down on average, worst case, harmonic average, asymptotically, etc. We only found very few cases of at most linear speed-up.

So the overall behavior of these small TMs revealed interesting structures to us. But also looking at each particular TM showed interesting structures. In Figure 2 we show two such examples. The rule numbering refers to Wolfram's enumeration scheme for (2,2) space as explained in [10] and [7].

For TM number 2205 we have plotted the tape evolution for the first 6 entries. So, each gridded rectangle represents a complete computation for a certain input. The diagrams should be interpreted as follows. The top row represents the initial tape configuration. The white cells represent a zero and the black cells a one. The grey cell represent the edge of the tape. Now each row in the gridded rectangle depicts the tape configuration after one more step in the computation. That is why each row differs at at most one cell from the previous row. We call these rectangles space-time diagrams of our computation where the space/tape is depicted horizontally and the time vertically.

We now see that TM 2205 always outputs just one black cell. Its computation yields a space-time diagram with a very clear localized character where the head

has just moved from right to the left-end of the input and back to the right end again doing some easy computation in between. TM number 1351 shows a clear recursive structure. Curiously enough this machine computes a very easy function which is just the tape identity. So it does a dazing amount of things (it needs exponential time for it) to leave in the end (the bottom row) the tape in the exact same configuration as the input (the top row). For more examples and structure we refer the interested reader to [8].

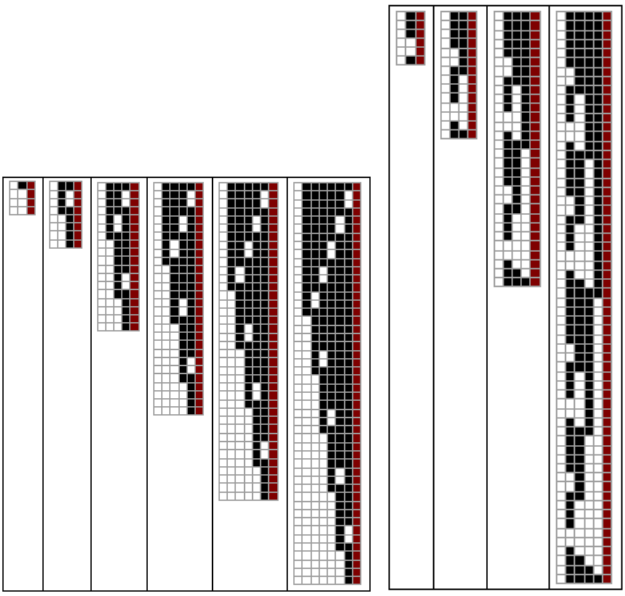


Fig. 2. Tape evolution for Rules 2205 (left) and 1351 (right)

Let us take a closer look at our pictures from Figure 2. It is clear that each TM defines an infinite sequence of these space-time diagrams as each different input defines such a diagram. It is pretty standard to assign to this sequence of space-time diagrams a fractal dimension d_τ that describes some features of the asymptotic behavior of a TM τ. We have empirically established for all TMs in (2,2)-space a very curious correspondence. It turns out that

> The fractal dimension d_τ that corresponds to a TM τ is 2 if and only if τ computes in linear time. The dimension d_τ is 1 if and only if τ computes in exponential time.

This result is remarkable because it relates two completely different complexity measures: the geometrical fractal dimension on the one side versus the time complexity of a computation on the other side. The result is one out of the many recent results that link various notions of complexity the existence of which we already forecasted on philosophical grounds in Section 2.1.

Acknowledgments. The author would like to thank David Fernández-Duque, José Martínez-Fernández, Todd Rowland, Stephen Wolfram, and Hector Zenil for their comments, discussions and general scientific input and feedback for this paper.

References

[1] Agrawal, M., Kayal, N., Saxena, N.: PRIMES is in P. Annals of Mathematics 160(2), 781–793 (2004)
[2] Bennett, C.H.: The Thermodynamics of Computation - A Review. Int. J. Theoretical Physics 21(12), 905–940 (1982)
[3] Calude, C.S., Stay, M.A.: Most programs stop quickly or never halt. Advances in Applied Mathematics 40, 295–308 (2005)
[4] Calude, C.S., Paun, G.: Computing with Cells and Atoms: An Introduction to Quantum, DNA and Membrane Computing. CRC Press (2000)
[5] Dawkins, R.: The Selfish Gene. Oxford University Press, New York City (1976) ISBN 0-19-286092-5
[6] Fodor, J.A.: Modules, Frames, Fridgeons, Sleeping Dogs, and the Music of the Spheres. In: Pylyshyn (1987)
[7] Joosten, J.J.: Turing Machine Enumeration: NKS versus Lexicographical, Wolfram Demonstrations Project (2010), `http://demonstrations.wolfram.com/TuringMachineEnumeratinNKSVersusLexicographical/`
[8] Joosten, J.J., Soler, F., Zenil, H.: Program-size versus Time Complexity. Slowdown and Speed-up Phenomena in the Micro-cosmos of Small Turing Machines. Int. Journ. of Unconventional Computing 7, 353–387 (2011)
[9] Joosten, J.J., Soler Toscano, F., Zenil, H.: Turing machine runtimes per number of states. To appear in Wolfram Demonstrations Project (2012)
[10] Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
[11] Paun, G., Rozenberg, G., Salomaa, A.: DNA Computing: New Computing Paradigms. Springer (2010)
[12] Pitowsky, I.: The Physical Church Thesis and Physical Computational Complexity. Iyyun, A Jerusalem Philosophical Quarterly 39, 81–99 (1990)
[13] Schrödinger, E.: What is Life? Cambridge University Press (1944)
[14] Wolfram, S.: A New Kind of Science. Wolfram Media (2002)
[15] Zenil, H., Soler Toscano, F., Joosten, J.J.: Empirical encounters with computational irreducibility and unpredictability. Minds and Machines 21 (2011)

Chapter 3
A Lyapunov View on the Stability of Two-State Cellular Automata

Jan M. Baetens and Bernard De Baets

KERMIT, Department of Mathematical Modelling, Statistics and Bioinformatics
Ghent University, Coupure links 653, 9000 Gent, Belgium

Abstract. Although cellular automata (CAs) have been invented only about three quarter of a century ago, much has been written about the fascinating space-time patterns these utter discrete dynamical systems can bring forth, which can be largely attributed to the fact that it is striking to notice that the dynamics of these overly simple dynamical systems can be so intriguing. Driven by the advances in the theory of dynamical systems of continuous dynamical systems, several quantitative measures have been proposed to grasp the stability of CAs, among which the Lyapunov exponent (LE) has received particular attention. Originally, the latter was understood as the Hamming distance between configurations evolved by the same CA from different initially close configurations, but it suffers from the important drawback that it can grow only linearly over time. In this paper, it will be shown how one should determine the LE of a two-state CA, in such a way that its meaning is in line with the framework of continuous dynamical systems. Besides, the proposed methodology will be exemplified for two CA families.

1 Introduction

Ever since cellular automata (CAs) were conceived by their spiritual father, the brilliant von Neumann, following a suggestion of with regard to the representation of the elements of a CA [16, 17], his successors have been impressed by the intriguing space-time patterns that some CAs can generate in spite of their intrinsically simple nature. For instance, von Neumann's contemporary [14] studied the growth patterns generated by CAs [13], while [11] discovered that not every possible configuration can be reached during a CA's evolution. Yet, it was not until records on the spatio-temporal dynamics of a CA, which was devised by Conway in an attempt to construct a simpler self-replicating machine than the one assembled by , and which is commonly known as 'The Game of Life' [7], penetrated literature before the CA paradigm and the intriguing dynamics of some CAs got known across various scientific disciplines. Although studies on the dynamics of CAs in the sixties and seventies were largely confined to a visual inspection and discussion of the evolved space-time patterns, from the eighties on, several computer scientists and mathematicians endeavored a more

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 25–33.
DOI: 10.1007/978-3-642-35482-3_3

quantitative assessment of CA dynamics in line with the efforts that had enabled a profound understanding of dynamical systems that are built upon a continuous state space, such as differential equations and iterated maps [10, 12]. More specifically, researchers have proposed quantitative measures for grasping the dynamical properties of CAs in general, and their stability, *i.e.* their characteristic behavior in the long run, in particular, such as the [18], the Langton parameter [9], entropies and dimensions [8], and others [18, 20]. The former had been understood by [18] as the analogue of the leading Lyapunov exponent of a continuous dynamical system and quantifies the total number of site values that differ after a given number of time steps between the configurations that were evolved from initial configurations that differed in only one cell. Yet, this metric has the important shortcoming that it can only grow linearly over time so that it precludes the existence of exponentially diverging phase space trajectories [5], which is, however, an intrinsic property of so-called chaotic dynamical systems [6]. Consequently, the Hamming distance cannot grasp the chaotic nature that has been discovered among a substantial share of CAs. Typically, such CAs are typified as Class 3 or 4 CAs according to the classification originally put forward by [18] in the framework of one-dimensional CAs, but which is conjectured to be applicable to any kind of CA family [19]. The notion of of CAs can however be brought in agreement with their meaning in the framework of continuous dynamical systems by reconsidering the propagation of perturbations in CAs.

This paper is organized as follows. In Section 2, the shortcoming of the Hamming distance as an analogue of Lyapunov exponents for continuous dynamical systems will be illustrated and the methodology that should be adopted for tracking perturbations in two-state CAs will be outlined. In the subsequent section, we will demonstrate how the stability of CAs can be expressed in terms of Lyapunov exponents that are built upon this tracking procedure and we will exemplify the soundness of the proposed methodology for two CA families.

2 Tracking Perturbations in Cellular Automata

Before turning to the methodology that should be adopted in order to quantify the propagation of perturbations in two-state CAs, we indicate why the Hamming distance does not comply with the notion of Lyapunov exponents as they have been conceived for continuous dynamical systems.

2.1 The Inadequacy of the Hamming Distance

In the remainder of this paper, we denote the state value of a two-state CA $\mathscr{C}$ at a given time step t as $s(c_i, t)$, where c_i refers to the ith cell of the tessellation upon which $\mathscr{C}$ is built. Let us consider an initial configuration s_0 and its perturbed counterpart s_0^*, which differs in only one cell from s_0, meaning that there is only one cell c_j in the CA for which the initial state value does not agree. In the remainder we refer to such a perturbed cell as a defective cell and to the actual as a . More precisely, in case of two-state CAs this entails that $s_0^*(c_j)$ is the so-called

Boolean complement of $s_0(c_j)$, *i.e.* if $s_0(c_j) = 0$ then $s_0^*(c_j) = 1$, and vice versa. Now, if we evolve a CA from s_0 and co-evolve the same CA from s_0^*, we may compute the Hamming distance d_t^H between the configurations $s(\cdot, t)$ and $s^*(\cdot, t)$ at the tth time step as the number of cells c_k for which $s(c_k, t) \neq s^*(c_k, t)$, *i.e.* the number of defective cells. Clearly, as a defective cell can only affect its immediate neighbors in every consecutive time step, the Hamming distance can only grow linearly over time since it is upper bounded by the number of cells, which, after t time steps, receive a bit of information that was supplied at $t = 0$ to the initially defective cell c_j. Hence, if elementary CAs are stake, it is clear that d_t^H is upper bounded by $1 + 2\,t$ since a defective cell can affect at most two additional cells at every consecutive time step, being the cells that border the left and right side of the field of dependence that originates from the initially defective cell [5]. Such a field of dependence is illustrated in the space-time diagram depicted in Fig. 1, and shows the maximal propagation of information, and hence defects, that originates from a single cell in the center of the tessellation. Similar observations have been made with regard to two-dimensional lattice gas automata [4] and graph CAs [1].

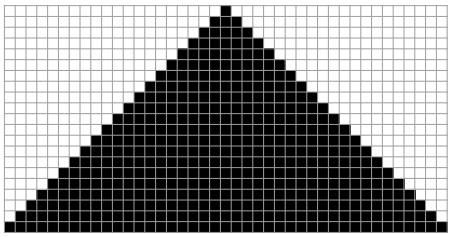


Fig. 1. Domain of dependence originating from a single cell in the center of the tessellation

Taking into account that the Lyapunov exponent of a continuous dynamical system quantifies the mean exponential rate of divergence of two initially close trajectories in phase space [6], it is clear that the Hamming distance cannot serve as its equivalent in case of two-state CAs since it can only grow linearly over time. Basically, the linearity of this metric seems to indicate that it loses track of a substantial share of defects that arise during the evolution of a CA, so the nature of propagation of defects in CAs must be reconsidered in order to arrive at a measure that allows for an exponential increase over time.

2.2 Tracking *All* Perturbations

From the preceding discussion it should be clear that the Hamming distance merely allows for tracking the number of defective cells that arise during the evolution of a CA, whereas it does not yield any insight into the actual number of defects, denoted ϵ_t, which might be of a completely different order of magnitude than the actual number of defective cells, and which can be tracked by explicitly

accounting for all possible ways defects can propagate. This discrepancy is not yet clear after the first time step because every defective cell at $t = 1$ traces back to the same initial defective cell c_j that is located in the center of the tessellation. As such, the total number of defective cells at the first time step d_t^H equals the total number of defects at the same time step. Yet, as soon as the CA is evolved one more time step, a subtle – though significant – discrepancy arises between the quantities ϵ_2 and d_2^H. This discrepancy can be understood by explicitly graphing all possible ways defects at $t = 1$ may propagate and accumulate during one subsequent time step as illustrated in Fig 2. More precisely, this figure depicts the cells of a CA as disconnected squares that are colored black if the corresponding cell is defective, and white otherwise, together with all possible ways (arrows) defects can propagate during the first three time steps in the evolution of an elementary CA that allows for maximal propagation of defects. From this figure it should be clear that there are three possible ways for a defect to emerge in the initially defective cell c_j, whereas there are two such pathways possible in case of c_j's right and left neighboring cells and only way pathway for the left- and rightmost cells in the domain of dependence that originates from c_j. Hence, all together, there are nine possible ways along which defects may propagate during the second time step in the CA evolution. This entails that the five defective cells at $t = 2$, which are detected using the Hamming distance, may actually embody nine defects, such that, from the second time step on, there is clear discrepancy between the number of defective cells on the one hand, and the number of defects, on the other hand.

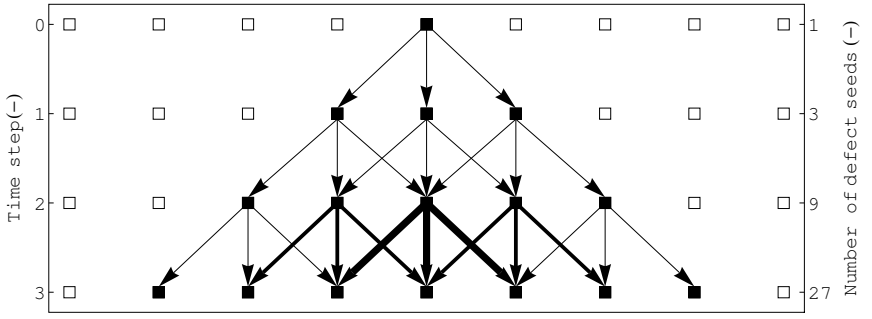


Fig. 2. Maximal propagation of defects (black) in evolution space of an elementary CA together with all possible ways defects can propagate (arrows)

This discrepancy becomes even clearer upon evolving the CA one more time step since then one also has to account for the multiplicity of the defects that can propagate, which is indicated by the variable thickness of the arrows in Fig. 2. For instance, the center cell of the tessellation can become affected along seven different pathways, three of them involving the center cell at $t = 2$, and four of them originating from the center cell's left and right neighboring cell. All

together there are 27 possible ways along which defects may propagate during the third time step in the CA evolution, which strongly deviates from the number of defective cells at $t = 3$. Continuing along this line of reasoning it should be clear that the maximum number of defects at time step t is given by 3^t, which indicates that ϵ_t can grow exponentially over time and might therefore constitute a means to define Lyapunov exponents in the framework of CAs of which the notion complies with the one of Lyapunov exponents for continuous dynamical systems. Similarly, it can be shown that the maximum number of defects at a given time step in case of graph CAs or CAs on arbitrary tessellations is given by C^t, where C denotes the CA's average number of neighbors.

3 Stability of Cellular Automata

3.1 Lyapunov Exponents of Cellular Automata

Recalling that the Lyapunov exponent of a continuous dynamical system quantifies the mean exponential rate of divergence of two initially close trajectories in phase space [6] and acknowledging that the number of defects ϵ_t gives insight into the divergence of two initially close CA configurations s_0 and s_0^* (*i.e.* $\epsilon_0 = 1$), the Lyapunov exponent (LE) of a two-state CA may be defined as:

$$\lambda = \lim_{t \to \infty} \frac{1}{t} \log\left(\epsilon_t\right) . \tag{1}$$

It should be emphasized that this equation leads to the largest LE of a CA, whereas the procedure for finding the other LEs that make up a CA's full Lyapunov spectrum, which is a characteristic of any higher-dimensional dynamical system, is more involved [2].

Now, just as in case of a of continuous dynamical systems, the sign of λ gives insight into the stability of CAs. More precisely, if the number of defects increases exponentially over time, Eq. (1) will give rise to a positive LE, and hence it may be said that the underlying CA exhibits sensitive dependence on initial conditions, so that it may also be referred to as an unstable CA. Seen the fact that ϵ_t can be at most 3^t if elementary CAs are at stake [5], and C^t if graph CAs or CAs on arbitrary tessellations are considered [1], it should be clear that the LE of the former is upper bounded by $\log(3)$, and by $\log(C)$ if the latter are at stake. Further, it is clear that a zero LE can only occur if $\epsilon_t = 1$ for all t, indicating that the initial separation between s_0 and s_0^* remains as the CA evolves. Cellular automata exhibiting this behavior may be referred to as stable CAs. Finally, a negative LE can occur if the separation between phase space trajectories decreases over time, which, taking into account that $\epsilon_0 = 1$, is only possible if $\epsilon_t = 0$, and implies that there is only one negative LE possible in the framework of CAs, namely $-\infty$, which clearly contrasts with the entire range of negative real-valued LEs that can be attained by continuous dynamical systems. Such CAs may be referred to as superstable CAs.

Following the postulation of Eq. (1) as the mathematical construct defining the LE of CA, it has been demonstrated that this LE is related to the sensitivity

of a CA to its inputs. More specifically, upon quantifying this sensitivity using Boolean Jacobians [15] on a relative scale between zero and one, it can be shown that the upper bound on the LE at a given sensitivity of a graph CA to its inputs, denoted $\bar{\mu}$, is given by $\log(\bar{\mu}\, C)$, and similarly for elementary CA for which $C = 3$ [1]. Moreover, with regard to unstable CAs, it has been found that the LE in most cases approaches this theoretical upper bound. Hence, the cumbersome computations that are involved in an assessment of the LE can be overcome by determining the CA's sensitivity to its inputs, which is far less computationally demanding than an assessment of ϵ_t, and compute an estimate of its LE using the aforementioned upper bound $\log(\bar{\mu}\, C)$. It has been demonstrated that a similar reasoning is possible if asynchronously updated CAs are at stake [3].

For comprehensiveness, it should be noticed that the discrete nature of a CA's phase space may hinder the evolution of a CA to a stable configuration (an attractor) notwithstanding the consecutive configurations it brings forth are less stable, which may be attributed to a lack of configurations that bridge the gap between the more stable configuration and the semistable ones. Although a transition to the more stable configuration can be induced by adding some noise to the subsequent configurations [5], it might be worthwhile to take the two-fold nature of the stability of some CAs into consideration explicitly. For that purpose, the LE of a given CA should be computed for an ensemble of different initial perturbations after which its stability can be typified as unconditionally unstable (Class C) if it gives rise to a positive LE for all members of the ensemble, and analogously, unconditionally superstable (Class A) if the initial separation between s_0 and any of the ensemble's members vanishes over time. On the other hand, those CAs for which some members of the ensemble lead to a positive LE while others lead to $-\infty$ may be typified as conditionally unstable or superstable (Class B), depending on the relative frequency of superstable and unstable behaviour among the members of the ensemble [1].

3.2 Some Examples

As a means to exemplify the methodology outlined in the preceding part of this paper, Table 1 lists the classification of the 88 representative elementary CAs according to both the classification scheme by [18] and the classification based upon their LEs. The latter were calculated over an ensemble of eight initial perturbations by applying the procedure described in Section 2 for 500 time steps and a one-dimensional tessellation of 675 cells on which periodic boundary conditions were imposed. From this table, it is clear that all Class 1 rules give rise to superstable behavior irrespective of the initial perturbation. This coincides with our expectations since this class encloses all rules that lead to a homogeneous configuration, which, from a dynamical systems point of view, means that their ϵ_t approaches zero as t grows large, irrespective of the initial perturbation. Further, Table 1 indicates that all Class 4 rules and, leaving aside

rules 18 and 126, the Class 3 rules are typified as unconditionally unstable based upon an assessment of their LEs, which indicates that these rules give rise to exponential divergence of initially close phase space trajectories and therewith underlines their sensitive dependence on initial conditions. It also points out the ambiguous nature of Class 2 rules since most of them give rise to either a positive LE or a a LE of $-\infty$ depending on the initial perturbation, but there also a few that are classified as unconditionally unstable based upon their LEs, such as rules 6,15 and 28, and there are even two Class 2 rules (14 and 142) that lead to converging trajectories.

Table 1. Classification of the 88 representative CAs based upon Wolfram's classification scheme and their Lyapunov exponents (LEs)

rule	Wolfram	LE	rule	Wolfram	LE	rule	Wolfram	LE
0	1	A	35	2	B	108	2	C
1	2	B	36	2	B	110	4	C
2	2	B	37	2	C	122	3	C
3	2	B	38	2	B	126	3	B
4	2	B	40	1	A	128	1	A
5	2	B	41	4	C	130	2	B
6	2	C	42	2	C	132	2	C
7	2	B	43	2	B	134	2	C
8	1	A	44	2	C	136	1	A
9	2	B	45	3	C	138	2	B
10	2	B	46	2	B	140	2	B
11	2	B	50	2	B	142	2	A
12	2	B	51	2	C	146	3	C
13	2	B	54	4	C	150	3	C
14	2	A	56	2	C	152	2	B
15	2	C	57	2	C	154	2	C
18	3	B	58	2	B	156	2	C
19	2	B	60	3	C	160	1	A
22	3	C	62	2	C	162	2	B
23	2	B	72	2	B	164	2	C
24	2	B	73	2	C	168	1	A
25	2	B	74	2	B	170	2	C
26	2	B	76	2	C	172	2	B
27	2	B	77	2	B	178	2	B
28	2	C	78	2	B	184	2	C
29	2	C	90	3	C	200	2	B
30	3	C	94	2	C	204	2	C
32	1	A	104	2	B	232	2	B
33	2	C	105	3	C			
34	2	B	106	4	C			

As a further justification of the methodology outlined in Section 2, Figure 3 shows the LE (λ) versus the input sensitivity ($\bar{\mu}$) for a family of 256 two-state totalistic CAs that are built upon an irregular tessellation consisting of 675 cells and for which $\lambda \neq -\infty$ for at least one member of an ensemble of eight initial conditions. Rules giving rise to $\lambda = -\infty$ for at least one member of the ensemble are indicated with square markers. Besides, this figure also displays the function $\lambda(\bar{\mu}) = \log(C\,\bar{\mu})$, which for the tessellation used in these simulations, equals $\lambda(\bar{\mu}) = \log(6.97\,\bar{\mu})$ and visualizes the theoretical upper bound on λ for a given input sensitivity $\bar{\mu}$. It is clear that most data points are lying near this curve, which demonstrates that the LE of an unstable CA in most cases approaches its theoretical upper bound. Moreover, such a plot allows for judging the stability of the involved (conditionally) unstable CAs at a glance.

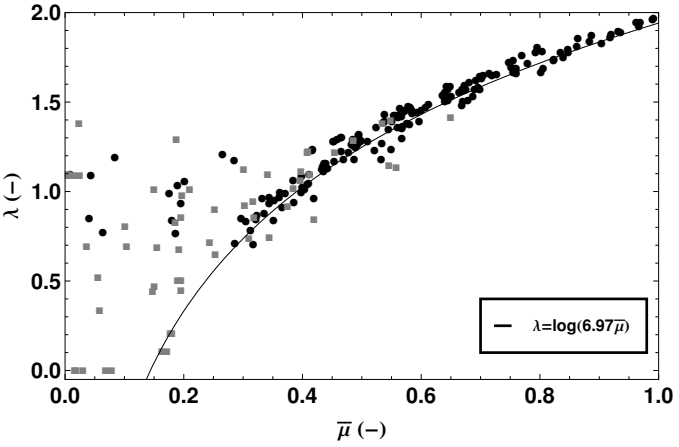


Fig. 3. Lyapunov exponent (λ) versus the input sensitivity ($\bar{\mu}$) for a family of 256 two-state totalistic CAs for which $\lambda \neq -\infty$ for at least one member of an ensemble of eight initial conditions. Rules giving rise to $\lambda = -\infty$ for at least one member of the ensemble are indicated with square markers.

4 Conclusion

In this paper we showed how the propagation of perturbations in two-state CAs should be tracked in order to get an insight into their stability. More precisely, we demonstrated that it is of utter importance to explicitly track all possible ways defects can propagate across the cellular space since this is the only way to correctly assess the number of defects emerging during the evolution of a CA from two initially close configurations that can be used thereafter to compute the CA's LE. The latter can then be used to classify CAs in accordance with their underlying dynamical properties.

References

[1] Baetens, J.M., De Baets, B.: Phenomenological study of irregular cellular automata based on Lyapunov exponents and Jacobians. Chaos 20, 033112 (2010)

[2] Baetens, J.M., De Baets, B.: Towards the full Lyapunov spectrum of elementary cellular automata. In: American Institute of Physics Conference Proceedings, Halkidiki, Greece, vol. 1389, p. 981 (2011)

[3] Baetens, J.M., Van der Weeën, P., De Baets, B.: Effect of asynchronous updating on the stability of cellular automata. Chaos, Solitons and Fractals 45, 383–394 (2012)

[4] Bagnoli, F., Rechtman, R.: Thermodynamic entropy and chaos in a discrete hydrodynamical system. Physical Review E 79, 041115 (2009)

[5] Bagnoli, F., Rechtman, R., Ruffo, S.: Damage spreading and Lyapunov exponents in cellular automata. Physics Letters A 172, 34–38 (1992)

[6] Eckmann, J.-P., Ruelle, D.: Ergodic theory of chaos and strange attractors. Reviews of Modern Physics 57, 617–656 (1985)

[7] Gardner, M.: Mathematical games: The fantastic combinations of John Conway's new solitaire game 'Life'. Scientific American 223, 120–123 (1970)

[8] Ilachinski, A. (ed.): Cellular Automata. A Discrete Universe. World Scientific, London (2001)

[9] Langton, C.: Computation at the edge of chaos. Physica D 42, 12–37 (1990)

[10] Lyapunov, A.M.: The general problem of the stability of motion. Taylor & Francis, London (1992)

[11] Moore, E.F.: Machine models of self reproduction. In: Bellman, R.E. (ed.) Proceedings of the 14th Symposium in Applied Mathematics, New York, United States, pp. 17–33 (1992)

[12] Oseledec, V.I.: A multiplicative ergodic theorem. Lyapunov characteristics numbers for dynamical systems. Transactions of the Moscow Mathematical Society 19, 197–231 (1968)

[13] Schrandt, R.G., Ulam, S.M.: On recursively defined geometrical objects and patterns of growth. Technical Report LA-3762, Los Alamos Scientific Laboratory (1968)

[14] Ulam, S.M.: On some mathematical problems connected with patterns of growth of figures. In: Bellman, R.E. (ed.) Proceedings of the 14th Symposium in Applied Mathematics, New York, United States, pp. 215–224 (1968)

[15] Vichniac, G.: Boolean derivatives on cellular automata. Physica D 45, 63–74 (1990)

[16] von Neumann, J.: The general and logical theory of automata. In: Jeffres, L.A. (ed.) The Hixon Symposium on Cerebral Mechanisms in Behaviour, pp. 1–41. John Wiley & Sons, Pasadena (1951)

[17] von Neumann, J.: Theory of Self-Reproducing Automata. University of Illinois Press, Urbana (1966)

[18] Wolfram, S.: Universality and complexity in cellular automata. Physica D 10, 1–35 (1984)

[19] Wolfram, S.: A New Kind of Science. Wolfram Media, Inc., Champaign (2002)

[20] Wuensche, A., Lesser, M.: The Global Dynamics of Cellular Automata, vol. 1. Addison-Wesley, London (1992)

Part II

Systems Based on Numbers and Simple Programs

Chapter 4
Cellular Automata and Hyperbolic Spaces

Maurice Margenstern

Université de Lorraine,
Campus du Saulcy, France
maurice.margenstern@univ-lorraine.fr, margenstern@gmail.com

Abstract. In this paper we look at the possibility to implement cellular automata in hyperbolic spaces and at a few consequences it may have, both on theory and on more practical problems.

Keywords: cellular automata, hyperbolic geometry, tilings.

1 Introduction

If many physical phenomena are governed by cellular automata as suggested by Stephen Wolfram in *A New Kind of Science*, then, it should be possible to do something with cellular automata and hyperbolic geometry.

When I started my first works on cellular automata in the hyperbolic plane, I heard only on Stephen's classification on elementary cellular automata. Alas, I was not aware of his papers on cellular automata. Moreover, the NKS appeared a few years later.

Nevertheless, I know the vivid effect on the study of cellular automata raised by Stephen's papers. I am very pleased to contribute to this volume and to present my works under such a prestigious auspice.

First, I shall try to explain the reader why hyperbolic geometry and then why to implement cellular automata in this frame. Then, if the reader has well captured one of the messages of NKS, he/she will not be very much surprised that I speak about universality and undecidability in this context.

2 Hyperbolic Geometry: Why Not?

2.1 A Quick Look at History

Hyperbolic geometry appeared in the first half of the 19^{th} century, in the last attempts to prove the famous parallel axiom of Euclid's *Elements* from the remaining ones. These attempts were probably motivated by the fact that this axiom was formulated by Euclid in the form of a theorem rather than an assertion which seems obviously true like *through two distinct points there is a line and a single one*. Also, in his famous *Elements*, Euclid made use of this axiom

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 37–46.
DOI: 10.1007/978-3-642-35482-3_4

as lately as possible, a point which also puzzled his followers. Now, as pointed by Coxeter in [6],

Moreover, his reluctance to introduce it provides a case for calling him the first non-Euclidean geometer!

This sentence should not be read as implying that Euclid discovered the non-Euclidean geometries. I think that it is reasonable to consider that Euclid realized that he was at a kind of crossroad and that another way could be explored. But, eventually, Euclid choose what seemed to him the simpler. It can be noticed that his choice is in agreement with the architecture of his surrounding world at that time.

More than two thousand years were needed to find out the answer to this question. Independently, around 1830, Lobachevsky and Bolyai discovered a new geometry by assuming that in the plane, from a point out of a given line, there are at least two lines which are parallel to the given line. This was an unexpected issue, although it was the natural conclusion of a line of research which started around one hundred and fifty years before the discovery, see [27, 3].

Later, around 1870, models of the new geometry were found, in particular Poincaré's model, which is the frame of all this paper.

The significance of this discovery takes a particular place in the history of sciences. It was the first time that people discovered that a set of *natural* axioms failed to find a definite foundation. It was the first time, but not the last. Unfortunately, we have no room to go further along this line.

2.2 Hyperbolic Geometry in Two Words

We shall describe the minimal features of hyperbolic geometry within Poincaré's disc model which seems to me the best for intuition.

Accordingly, the **hyperbolic plane** is the set of points inside an open disc $\mathcal{D}$ of the Euclidean plane, fixed once and for all. The border of $\mathcal{D}$ is called the set of **points at infinity** of the hyperbolic plane and these points do not belong to this plane. Lines are trace of diameters or circles orthogonal to the border of the disc, e.g. the line m in Figure 1.

In this model, two lines which meet in the open disc are called **secant** and two lines which meet at infinity, *i.e.* at a point at infinity are called **parallel**. In the figure, we can see a line s through the point A which cuts m. Now, we can see that two lines pass through A which are parallel to m: p and q. They touch m in the model at P and Q respectively which are points at infinity. At last, and not the least: the line n also passes through A without cutting m, neither inside the disc nor outside it. This line is called **non-secant**.

An important feature of hyperbolic geometry is that in this world there is no similarity which plays so great a role in Euclidean geometry. An important property of the model is that the angle between two lines meeting at A are the Euclidean angle of the tangents at A of the supports of the lines. From this we can define perpendiculars, bisectors and many notions connected with the geometry of the triangle which is here very different from the Euclidean geometry of the triangle.

Before turning to cellular automata, let us mention that Poincaré's disc model generalizes to any dimension. In particular, for the $3D$ hyperbolic space, the

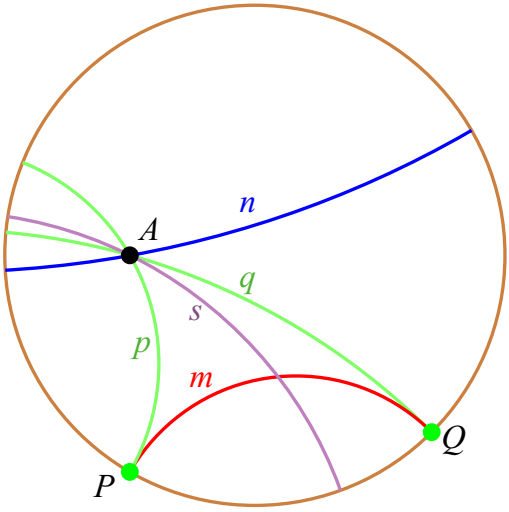


Fig. 1. Poincaré's disc model

space is defined by a fixed open ball $\mathcal{B}$, its border defines the points at infinity and the planes are the traces of diametral planes or of spheres which are orthogonal to the sphere of which the border consists. Lines are intersections of planes. Now, it is not difficult to see that they belong to diametral planes.

2.3 Tilings and Cellular Automata

An unexpected result of the study of the hyperbolic plane is the feature discovered by Henri Poincaré in the late 19^{th} century. Consider a regular convex polygon P and the following process: replicate P by reflection in its sides and then, recursively, replicate the images by reflection in their sides. Later, these images are called **copies** of P. If all the copies cover the hyperbolic plane and if the copies do not pairwise overlap, we say that they **tile** the hyperbolic plane. We also say that a tiling obtained by the just described process is a **tessellation**. Can we obtain a tessellation starting from any regular convex polygon P? The answer is no but, what is surprising, is that there are infinitely many such tessellations. They are characterized by a simple relation between the number p of their sides and the number q of copies of P which can be put around a vertex with no overlap and which cover a neighbourhood of the vertex. It is enough that $\frac{1}{p} + \frac{1}{q} < \frac{1}{2}$ in order that the polygon defined by p and q generates a tessellation of the hyperbolic plane denoted by $\{p, q\}$.

By contrast, in the Euclidean plane, up to similarities, there are only three tessellations: the square, the regular hexagon and the equilateral triangle, $\{4, 4\}$, $\{6, 3\}$ and $\{3, 6\}$ respectively.

Now, we can see that we have a large choice of a grid in order to define there cellular automata. We shall mainly investigate what we performed in two

particular tessellations of the hyperbolic plane: the **pentagrid** and the **heptagrid**, $\{5,4\}$ and $\{7,3\}$ respectively. Figure 2 illustrates these grids, the pentagrid, on the left-hand side and the heptagrid, on the right-hand side.

Unfortunately, we have no room here for giving even a hint at how coordinates can be introduced for the tiles. This is absolutely needed if we want to implement cellular automata on a grid: first, we must identify the tiles, at least to allow each tile to locate its neighbours. We can just mention [9], the first paper where I introduced such coordinates, first on the pentagrid. The reader is referred to [11, 13] for a detailed presentation of these coordinates, for the pentagrid, for the heptagrid and for infinitely many other tilings of the hyperbolic plane which are basically the generalizations of what I found in [9]. There, there are also coordinates for the **dodecagrid**, the tessellation of the $3D$-hyperbolic space based on the dodecahedron whose faces are copies of the pentagon on which the pentagrid is generated: this also generalizes the just mentioned planar system of coordinates.

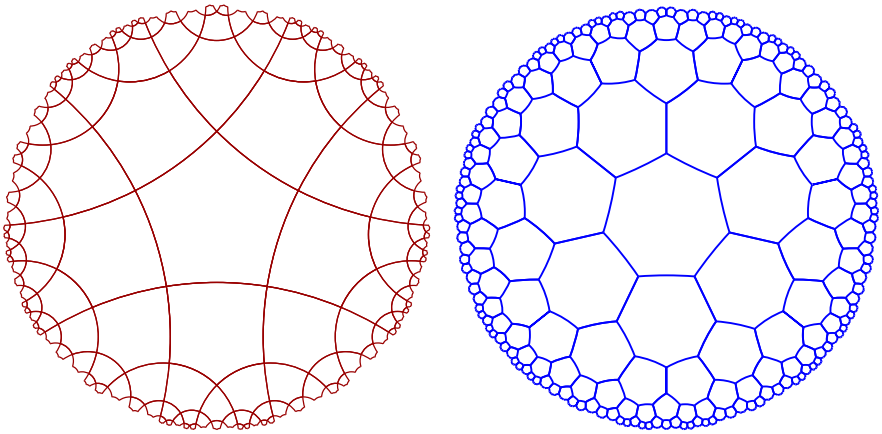


Fig. 2. Pentagrid, left, and heptagrid, right

As the tile $(0,0)$ is attached to a special element of $\mathbb{Z}^2$ in the case of the Euclidean square grid, there is a **central tile** in the case of the just mentioned tessellations of the hyperbolic plane. Now, in these case, these coordinates have a remarkable property: there is an algorithm which computes the path from the central tile to the tile in linear time in the size of its coordinate.

2.4 Why Cellular Automata in Hyperbolic Spaces?

Now, I go back to the question of our introduction.

Of course, hyperbolic geometry plays an important role in physics: in the theory of relativity, namely in the formulation of certain laws, and in a few models of our universe.

Now, there is another interest. At the beginning of my research in this frame, just after having found the first step to the coordinates defined in [9], I had a paper with a colleague where we found that **NP**-problems can be solved in polynomial time by cellular automata in the pentagrid, see [20]. In fact, this result was anticipated a few years ago in [24], which we did not know at that time, but the description we gave in [20] is very acute. A bit later, with other colleagues, see [8], we proved that $\mathbf{P} = \mathbf{NP}$ for cellular automata of the pentagrid, and this is also true for the heptagrid, although nobody proved it strictly speaking.

I think that this result is very important. First, it shows that the $\mathbf{P} = \mathbf{NP}$**?** question is an ill posed problem: complexity in time cannot be defined regardless of complexity in space. Second, I think that this could be a way to tackle the question and, more precisely, to prove that $\mathbf{P} \neq \mathbf{NP}$ in the Euclidean setting. There is an important difference for the results in the Euclidean setting and in the hyperbolic one. In the Euclidean setting, cellular automata work much faster than Turing machines, but the gain is polynomial. Indeed, a planar Turing machine needs only a cubic time to simulate a cellular automaton, of course, considering finite computations only. In the hyperbolic plane, a planar Turing machine may require an exponential time to perform the computation of a cellular automaton. And so, there is a gap between sequential and parallel deterministic devices in the hyperbolic plane, which is not the case in the Euclidean plane.

3 Practical and Theoretical Applications

We turn now to applications. There are two kinds of them: for practical purposes and for theoretical issues.

3.1 Simulations

As I have no room for explanation, I just list a few examples with appropriate illustrations.

The first application was a **colour chooser**, see [4] and the second one was a proposal of cellphone keyboard for the Japanese language, see [19] and the illustration of Figure 3. In fact, both work on the same principle: with eight selected keys for the chooser and six of them for the keyboard, the user can move the disc as a window upon the plane in order to place at the centre the tile corresponding to his/her choice. The differences are the following. In the case of the colour chooser, we use the heptagrid for aesthetic reasons and there, the amplitude of the moves of the user is arbitrary. In the case of the Japanese keyboard, we use the pentagrid as its tiles are bigger and the user needs at most three moves to reach the desired letter. Also, the pentagrid is well suited for the Japanese keyboard: the traditional display of hiraganas and katakanas which are phonetic syllabic alphabets is performed according to the five vowels of the Japanese language.

I also have an application of the heptagrid to the simulation of a message system between cells of the tessellation. I have no illustration for this and also

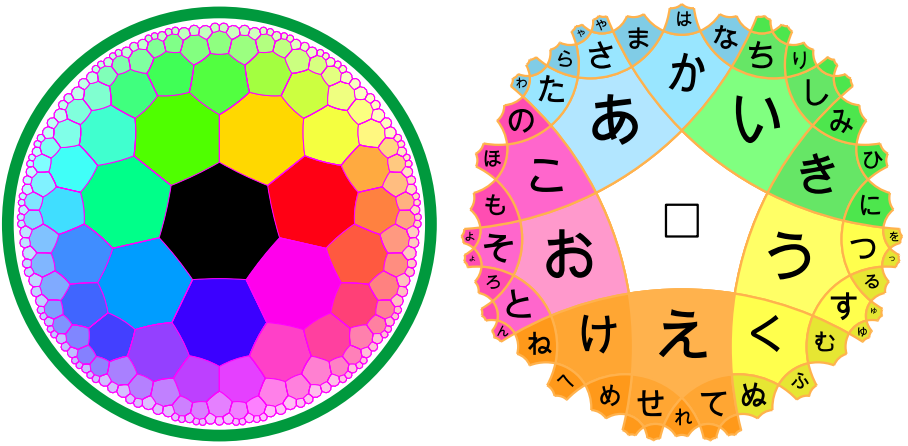


Fig. 3. To left: the colour chooser. To right: the Japanese keyboard

no room for further explanation. I can just mention that with a suitable Poisson law, the system seems to work satisfactorily, see [17].

To conclude this sub-section, I mention the simulation of the growth of a colony of bacteria by a cellular automaton on a grid which I obtained from the heptagrid: first divide each tile into seven triangles with a vertex at the center of the tile; second, divide each just obtained triangles into four triangles whose vertices are on the mid-points of the sides of the previous triangles. This simulation was inspired by the pictures of the fascinating paper [1]. I am very much in debt to its author, Professor Ben Jacob, for giving me the opportunity to reproduce one of his pictures. I have some considerations on these colonies which might interest the reader, see [16].

3.2 Universality

When I was planning to devise cellular automata in the hyperbolic plane, I wished to implement there universal cellular automata, in fact weakly universal ones as the initial configuration is infinite, but a regular one. I have just the room to sketchily tell the story of this adventure up to now.

As usual, when dealing with universality, we simulate something we call a model, which is well known to simulate any Turing machine in the end, in particular a universal one. Sometimes, it may be a direct simulation of a Turing machine but most often it is another model and, in some cases, as here, we simulate a model of a model.

Our model is the **railway model** devised in [26]. We have tracks and switches, as illustrated in Figure 5, left-hand side, reproducing those of [26].

By assembling copies of the element of circuit in the right-hand side of Figure 5, it is possible to devise circuits which mimic the circuitry of a computer and to model, in this way, a register machine. It is well known from [23] that such a machine can simulate a Turing machine with two registers only.

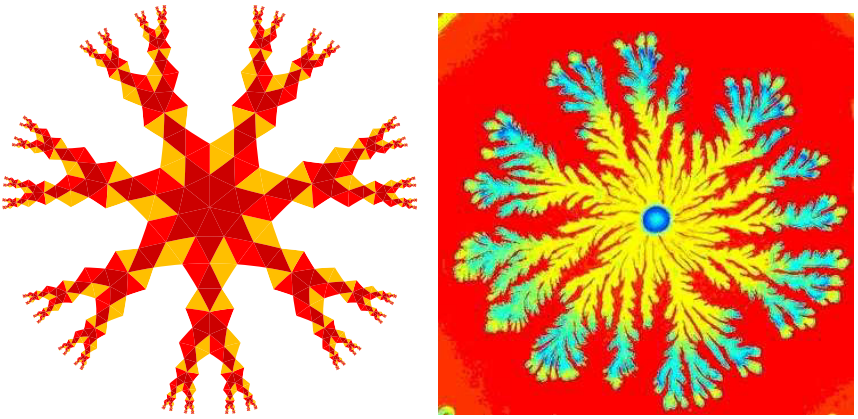


Fig. 4. To left: simulation of the growth of a bacteria colony with the heptatrigrid. To right: propagation of a bacteria colony, picture by courtesy of Professor Ben-Jacob.

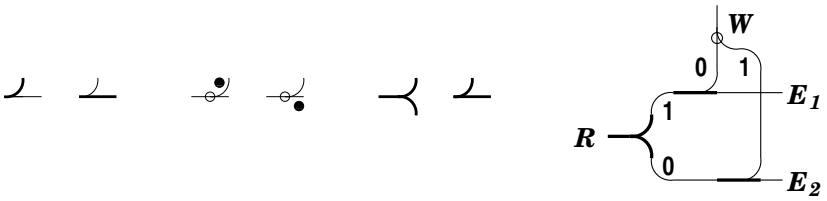


Fig. 5. To left: the switches, fixed switch, flip-flop and memory switch from left to right respectively, each one in two versions. To right: an element with one bit of information, the same as in [26].

The first universal cellular automaton appeared in [7] as an automaton in the pentagrid with 22 states. A few years later, I devised a much improved version in the hyperbolic $3D$ space, see [10] with 5 states only. Well: in the dodecagrid, each cell has 12 neighbours and the third dimension allows us to replace crossings by bridges which spares a lot of states. Then, in [22], the automaton in the pentagrid was reduced to 9 states and the translation of the same scenario into the heptagrid produced the first universal cellular automaton on the heptagrid and it has 6 states, see [21].

Then, in [15], I reduced the number of states for a universal cellular automaton in the heptagrid to 4 states. There was an important change in the scenario. Instead of simulating the tracks by a specific colour, the tracks are marked by **milestones** which have a different colour with respect to the blank background. This makes things a bit less simple, but this is the price to pay in order to reduce the number of states. The implementation of the idea of [15] in the hyperbolic $3D$ space directly produced a universal cellular automaton with 3 states. Three states seems to be very closed to two ones, which is the limit as a single state

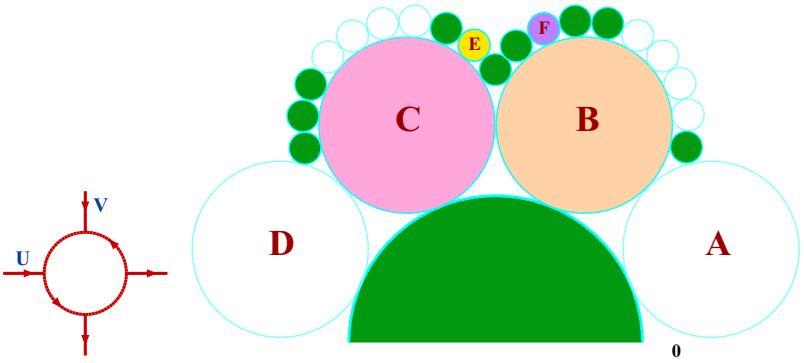


Fig. 6. Left-hand side: arriving through U or V, the right exit is the second one. Right-hand side, zoom at an exit: the locomotive, now a particle, arrives either at E or at A. When at E, it leaves through D as two particles. When at A: if one particle, it goes out through F, it two particles, one is kept and the other is sent to C, then D and further to the next exit.

cannot be universal, trivially. I succeeded to do that recently, by a new tuning: instead of two-way tracks, take one-way ones as in modern railways.

We remain with the plane. It is possible to go down to two states with an embedding of the elementary cellular automaton rule 110 known to be universal, see [5, 11]. This was done in [14]. But the automaton is not truly a planar one. Very recently, I could go down to two states with a planar cellular automaton, in infinitely many grids of the hyperbolic plane but starting from $\{13, 3\}$.

This was made possible by replacing crossings by round-abouts, see [18]. A partial view of the round-about in $\{13, 3\}$ is given by Figure 6 which also explains the mechanism.

I just mention another point about universality: this study allowed me to prove that the domino problem is undecidable in the hyperbolic plane. According to the *principle of computational equivalence* from the NKS, see [11], this is based on the simulation of a universal Turing machine. Now, here, the difficulty was to devise a way to force the tiling to simulate the Turing machine, whatever the tile with which we start the process of constructing the tiling, see [12]. The problem was solved in 1966 for the Euclidean plane, see [2] and raised for the hyperbolic plane in 1971, see [25]. Details on the history of the problem can be found in [12].

4 Conclusion

Some work has been achieved, but I think that many problems have still to be solved about cellular automata in the hyperbolic plane. It remains to find universal planar cellular automata with two states for the heptagrid and for the pentagrid which does not seem a simple task. We also remain with the question of universality in these grids when starting from a finite configuration, see a first

attempt in [14]. The **P = NP?** question is, of course a very challenging one and it would certainly be very nice to be given an answer here. Also, there are a lot of possible simulations which I could not explore. I hope that this journey will suggest a reader to walk on this path.

References

[1] Ben-Jacob, E.: Social behavior of bacteria: from physics to complex organization. European Physical Journal B 65(3), 315–322 (2008)

[2] Berger, R.: The undecidability of the domino problem. Memoirs of the American Mathematical Society 66, 1–72 (1966)

[3] Bonola, R.: Non-Euclidean Geometry. Dover (2007)

[4] Chelghoum, K., Margenstern, M., Martin, B., Pecci, I.: Palette hyperbolique: un outil pour interagir avec des ensembles de données. In: IHM 2004, Namur (2004)

[5] Cook, M.: Universality in Elementary Cellular Automata. Complex Systems 15(1), 1–40 (2004)

[6] Coxeter, H.S.M.: Non-Euclidean Geometry. Mathematical Association of America (1998)

[7] Herrmann, F., Margenstern, M.: A universal cellular automaton in the hyperbolic plane. Theoretical Computer Science 296, 327–364 (2003)

[8] Iwamoto, C., Margenstern, M., Morita, K., Worsch, T.: Polynomial Time Cellular Automata in the Hyperbolic Plane Accept Exactly the PSPACE Languages. In: SCI 2002 (2002)

[9] Margenstern, M.: New Tools for Cellular Automata of the Hyperbolic Plane. Journal of Universal Computer Science 6(12), 1226–1252 (2000)

[10] Margenstern, M.: A universal cellular automaton with five states in the 3D hyperbolic space. Journal of Cellular Automata 1(4), 315–351 (2006)

[11] Margenstern, M.: Cellular Automata in Hyperbolic Spaces. Theory, vol. 1, 422 p. Old City Publishing, Philadelphia (2007)

[12] Margenstern, M.: The Domino Problem of the Hyperbolic Plane Is Undecidable. Theoretical Computer Science 407, 29–84 (2008)

[13] Margenstern, M.: Cellular Automata in Hyperbolic Spaces. Implementation and computations, vol. 2, 360 p. Old City Publishing, Philadelphia (2008)

[14] Margenstern, M.: An upper bound on the number of states for a strongly universal hyperbolic cellular automaton on the pentagrid. In: JAC 2010, Turku, Finland, December 15-17 (2010) (accepted)

[15] Margenstern, M.: A universal cellular automaton on the heptagrid of the hyperbolic plane with four states. Theoretical Computer Science 412, 33–56 (2011)

[16] Margenstern, M.: Bacteria, Turing machines and hyperbolic cellular automata. In: Zenil, H. (ed.) A Computable Universe: Understanding and Exploring Nature as Computation, ch. 12. World Scientific (in press, 2012)

[17] Margenstern, M.: A protocol for a message system for the tiles of the heptagrid, in the hyperbolic plane. International Journal of Satellite Communications Policy and Management (in press)

[18] Margenstern, M.: Universal cellular automata with two states in the hyperbolic plane. Journal of Cellular Automata (in press)

[19] Margenstern, M., Martin, B., Umeo, H., Yamano, S., Nishioka, K.: A Proposal for a Japanese Keyboard on Cellular Phones. In: Umeo, H., Morishita, S., Nishinari, K., Komatsuzaki, T., Bandini, S. (eds.) ACRI 2008. LNCS, vol. 5191, pp. 299–306. Springer, Heidelberg (2008)

[20] Margenstern, M., Morita, K.: NP problems are tractable in the space of cellular automata in the hyperbolic plane. Theoretical Computer Science 259, 99–128 (2001)
[21] Margenstern, M., Song, Y.: A universal cellular automaton on the ternary heptagrid. Electronic Notes in Theoretical Computer Science 223, 167–185 (2008)
[22] Margenstern, M., Song, Y.: A new universal cellular automaton on the pentagrid. Parallel Processing Letters 19(2), 227–246 (2009)
[23] Minsky, M.L.: Computation: Finite and Infinite Machines. Prentice Hall, Englewood Cliffs (1967)
[24] Morgenstein, D., Kreinovich, V.: Which Algorithms are Feasible and Which are not Depends on the Geometry of Space-Time. Geocombinatorics 4(3), 80–97 (1995)
[25] Robinson, R.M.: Undecidability and nonperiodicity for tilings of the plane. Inventiones Mathematicae 12, 177–209 (1971)
[26] Stewart, I.: A Subway Named Turing, Mathematical Recreations. Scientific American, 90–92 (1994)
[27] Taimina, D.: Crocheting Adventures with Hyperbolic Planes, 148 p. A K Peters, Ltd., Wellesley (2009)
[28] Wolfram, S.: A New Kind of Science. Wolfram Media (2002)

Chapter 5
Symmetry and Complexity of Cellular Automata: Towards an Analytical Theory of Dynamical System

Klaus Mainzer and Carl von Linde-Akademie

Technische Universität München, München, Germany
mainzer@cvl-a.tum.de

Stephen Wolfram declared computer experiments with pattern formation of cellular automata as "new kind of science" (NKS). It is obviously a great merit of NKS to highlight the experimental approach in the computational sciences [26]. But we claim that even in the future quasi-empirical computer experiments are not sufficient [12]. Cellular automata must be considered complex dynamical systems in the strictly mathematical sense with corresponding equations and proofs. In short, we also need analytical models of cellular automata, in order to find precise answers and predictions in the universe of cellular automata. In this sense, our approach goes beyond Wolfram's NKS.

In our approach cellular automata (CA) are defined as complex dynamical systems. The geometrical representation of the eight CA-rules as a Boolean cube allows precise definitions of a complexity index and universal symmetries. It can be proved that the 256 one-dimensional cellular automata are classified by local and global symmetry classes of cellular automata. There is an exceptional symmetry group with universal computability which we call the "holy grail" in the universe of cellular automata. Although the four automata of this group are completely deterministic, their long-term behavior cannot be predicted in principle with respect to the undecidability of Turing's halting problem. Many analytical concepts of complexity research (e.g., attractors, basin of attractors, time series, power spectrum, fractality) are defined for cellular automata. But there are also surprising phenomena in the CA-world (isles of Eden) without analytical representation in dynamical systems.

1 Dynamics in the Universe of Cellular Automata

Because of their simplicity, rules of cellular automata can easily be understood. In a most simple version, we consider *two-state one-dimensional cellular automata* (CA) made of identical cells with a *periodic boundary condition*. In this case, the object of study is a ring of coupled cells with $L = I+1$ cells, labeled consecutively from $i = 0$ to $i = I$ (Fig. 1(a)). Each cell i has two *states* $u_i \in \{0, 1\}$, which are coded by the colors blue and red, respectively. A clock sets the pace in discrete times by iterations or generations. The state u_i^{t+1} of all i at time $t + 1$ (i.e. the

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 47–65.
DOI: 10.1007/978-3-642-35482-3_5

next generation) is determined by the states of its nearest neighbors u_{i-1}^t, u_{i+1}^t, and itself u_i^t at time t (Fig. 1(c)), i.e. by a *Boolean function* $u_i^{t+1} = N(u_{i-1}^t, u_i^t, u_{i+1}^t)$, in accordance with a prescribed *Boolean truth table* of $8 = 2^3$ distinct 3-input patterns (Fig. 1(d)).

1.1 From Simple Local Rules to Global Complex Patterns

These eight 3-input patterns can nicely be mapped into the eight vertices of a toy cube (Fig. 1(b)), henceforth called a *Boolean cube* [3]. The output of each prescribed 3-input pattern is mapped onto the corresponding colors (red for 1, blue for 0) at the vertices of the Boolean cube (in Fig. 1(d) yet unspecified). Since there are $2^8 = 256$ distinct combinations of eight bits, there are exactly 256 Boolean cubes with distinct vertex color combinations. Thus, we get a gallery of picturesque toy cubes.

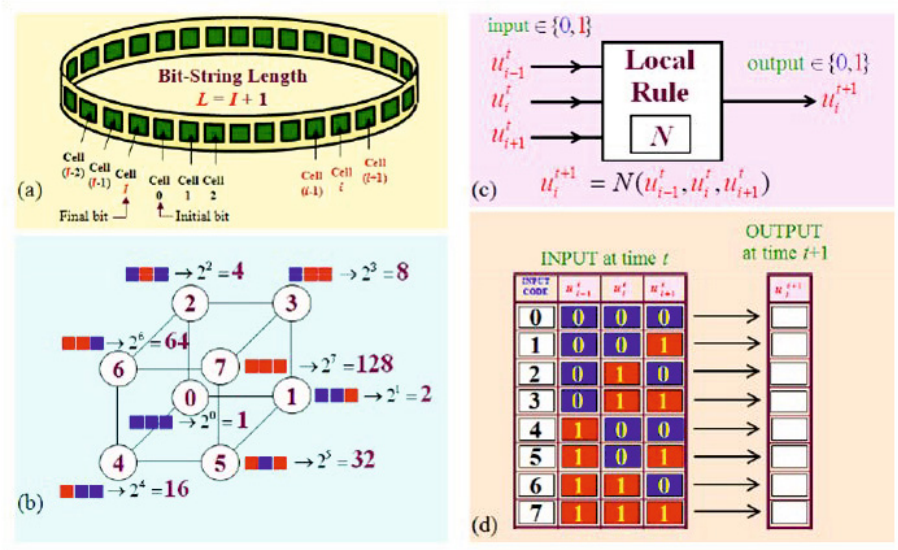


Fig. 1. Scheme of a two-state one-dimensional Cellular Automaton with local rule N

It is convenient to associate the 8-bit patterns of each Boolean function with a decimal number N representing the corresponding 8-bit word, namely $N = \beta_7 \cdot 2^7 + \beta_6 \cdot 2^6 + \beta_5 \cdot 2^5 + \beta_4 \cdot 2^4 + \beta_3 \cdot 2^3 + \beta_2 \cdot 2^2 + \beta_1 \cdot 2^1 + \beta_0 \cdot 2^0$ with $\beta \in \{0,1\}$. Notice that since $\beta_i = 0$ for each blue vertex in Fig. 1(b), N is simply obtained by adding the weights (indicated next to each pattern in Fig. 1(b)) associated with all red vertices. For example, for the Boolean cube shown in Fig. 2(b), we have $N = 0{\cdot}2^7{+}1{\cdot}2^6{+}1{\cdot}2^5{+}0{\cdot}2^4{+}1{\cdot}2^3{+}1{\cdot}2^2{+}1{\cdot}2^1{+}1{\cdot}2^0 = 2^6{+}2^5{+}2^3{+}2^2{+}2^1 = 110$.

For the example of local rule 110, the ring and the colored vertices of the Boolean cube are shown in Fig. 2(a)-(b). Given any initial binary bit-configuration at time $t = 0$, the local rule N is used to update the state u_i^{t+1} of each cell i at time $t + 1$, using the states of the three neighboring cells $i - 1$,

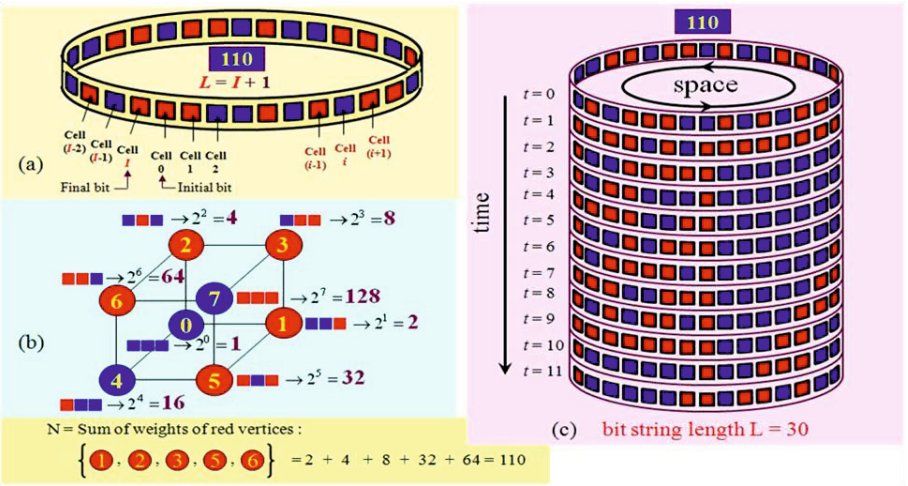


Fig. 2. Example of local rule 110

i, and $i+1$, centered at location i, respectively. The *space-time pattern* for the initial state is shown in Fig. 2(c) for $t = 0, 1, 2, \ldots, 11$.

In principle, one can draw and paint the patterns of cellular automata following these rules step by step. Modern computers with high speed and capacity allow extensive computer experiments to study pattern formations of these automata. Stephen Wolfram discovered remarkable analogies with patterns in physics and biology [26]. In the world of cellular automata many phenomena of the physical world seem to evolve. Some automata generate symmetric patterns reminding us of the coloring in sea shells, skins or feathers. Other automata reproduce rhythms like oscillating waves. Some of these automata stop their development after a finite number of steps, independent of their initial state, and remain in a constant color state like a system reaching at an equilibrium state for all future steps. Some automata develop complex patters reminding us of the growth of corals or plants, depending sensitively on tiny changes of the initial states. This phenomenon is well-known as the butterfly-effect, when local events lead to global effects in chaotic and unstable situations (e.g., weather and climate). Even these chaotic patterns can be generated by cellular automata.

One can try to classify these patterns with respect to their outward appearance like zoologists and botanists distinguishing birds and plants in taxonomies. But sometimes, outward features are misleading. Fundamental question arises: Are there laws of complex pattern formation for cellular automata like in nature? Can the development of complex patterns be predicted in a mathematically rigorous way like in physics? We argue for a mathematically precise explanation of the dynamics in cellular automata. Therefore, they must also be characterized by complex dynamical systems determined with differential equations like in physics. This is, of course, beyond the scope of elementary rules of toy worlds. But, we should keep this perspective in mind.

1.2 Cellular Automata as Dynamical Systems

For maximum generality, each cell i is assumed to be a *dynamical system* with an intrinsic state x_i, an output y_i and *three inputs* u_{i-1}, u_i, and u_{i+1} where u_{i-1} denotes the input coming from the left neighboring cell $i-1$, u_i denotes the self input of cell i , and u_{i+1} denotes the input coming from the right neighboring cell $i+1$ in the ring of Fig. 1(a). Each cell evolves with its prescribed dynamics and its own time scale. When coupled together, the system evolves consistently with its own rule as well as the rule of interaction imposed by the coupling laws.

Each *input* is assumed to be a *constant integer* $u_i \in \{-1, 1\}$, and the *output* y_i *converges* to a *constant* either -1 or 1 from a zero initial condition $x_i(0) = 0$. Actually, it takes a finite amount of time for any dynamical system to converge to an *attractor*. But, for the purpose of idealized cellular automata, each attractor is assumed to be reached instantaneously. Under this assumption and with respect to the *binary input* and *output*, our *dynamical system* can be defined by a *nonlinear map* which is uniquely described by a *truth table* of three input variables (u_{i-1}, u_i, u_{i+1}). The choice of $\{-1, 1\}$ and not $\{0, 1\}$ as binary signals is crucial, because the state x_i and output y_i evolves in *real time* via a carefully designed *scalar ordinary differential equation*. According to this differential equation, the output y_i which is defined via an output equation $y_i = y(x_i)$ tends to either 1 or -1 after the solution x_i (with zero initial state $x_i(0) = 0$) reaches a *steady state*. In this way, the *attractors* of the *dynamical system* can be used to encode a *binary truth table*.

Aside from the *cell's intrinsic time scale* (which is of no concern in cellular automata), an external clocking mechanism is introduced to reset the input u_i of each cell i at the end of each clock cycle by feeding back the steady state *output* $y_i \in \{-1, 1\}$ as an updated *input* $u_i \in \{-1, 1\}$ for the *next iteration*. This mechanism corresponds to the *periodic boundary condition* of a one-dimensional cellular automaton in Fig. 1(a).

Although cellular automata are concerned only with the ring's evolutions over *discrete times*, any computer used to simulate cellular automata is always a *continuous time system* with very small but non-zero time scale. Computers use transistors as devices, and each cellular automata iteration involves the physical evolution of millions of transistors with its own $u_i \in \{-1, 1\}$ intrinsic dynamics. These transistors evolve in accordance with a large system of nonlinear differential equations governing the entire internal computer circuit and return the desired output after converging to their attractors in a non-zero amount of time.

These considerations lead us to the important result that, even in *discrete systems* like cellular automata, there are *two different time scales* involved. The first one applies to the rule N while the second applies to the global patterns of evolution. In order to understand the complex dynamics of global patterns, it is necessary to analyze both times scales. By unfolding the truth tables of cellular automata into an appropriate nonlinear dynamical system, we can exploit the theory of *nonlinear differential equations* to arrive at phenomena based on a precise mathematical theory, and not only on empirical observations.

For this purpose, we substituted the binary symbol 0 by the -1, and the input and output values 0 and 1 in the truth table of Fig. 1(d) by the real numbers -1 and 1, respectively. An advantage of working with the numeric rather than the symbolic truth table is the remarkable insights provided by the equivalent Boolean cube representation. Here, the eight vertices of the cube $(-1,-1,-1)$, $(-1,-1,1)$, $(-1,1,-1)$, $(-1,1,1)$, $(1,-1,-1)$, $(1,-1,1)$, $(1,1,-1)$ and $(1,1,1)$ are located exactly at the coordinates $(u_{i-1},\ u_i,\ u_{i+1})$ of a *coordinate system* with the origin located at the center of the cube. The vertex $n = 0, 1, 2, \ldots, 7$ corresponding to row n of the truth table is coded blue if the output is -1 , and red if the output is 1.

The choice of $\{-1, 1\}$ instead of $\{0, 1\}$ as binary signals is necessary, when the truth table is mapped onto a dynamical system where the states evolve in real time via an ordinary differential equation which is always based on the real number system. Each cell i is coupled only to its left neighbor cell $i-1$ and right neighbor cell $i+1$. As a dynamical system, each cell i has a *state variable* x_i, an *output variable* y_i, and three constant *binary inputs* u_{i-1}, u_i and u_{i+1} (Fig. 3).

Thus, the *dynamical system* is determined by a

$$\begin{aligned} \textit{state equation}\colon \dot{x_i} &= f(x_i, u_{i-1}, u_i, u_{i+1}) \\ x(0) &= 0 \text{ (initial condition)} \\ \textit{output equation}\colon y_i &= y(x_i) \end{aligned}$$

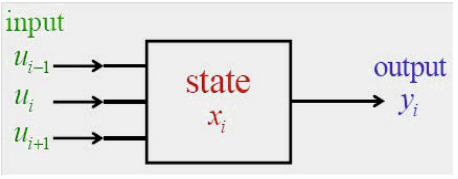


Fig. 3. Cell as dynamical system with state variable x_i, an output variable y_i, and three constant binary inputs u_{i-1}, u_i, and u_{i-1}

Every cellular automata can be mapped into a nonlinear dynamical system whose attractors encode precisely the associated truth table $N = 0, 1, 2, 3, \ldots, 255$. Function f models the *time-depend change of states* and is defined by a scalar ordinary differential equation of the form

$$\dot{x} = g(x_i) + w(u_{i-1}, u_i, u_{i+1}) \text{ with } g(x_i) \triangleq -x_i + |x_i + 1| - |x_i - 1| \,.$$

There are many possible choices of *nonlinear basis functions* for $g(x_i)$ and $w(u_{i-1}, u_i, u_{i+1})$. We have chosen the absolute value function $|x| = x$ for positive numbers x and $|x| = -x$ for negative numbers x as nonlinear basis function, because the resulting equation can be expressed in an optimally compact form, and it allows us to derive the solution of the state equation in an explicit form. The scalar function $w(u_{i-1},\ u_i,\ u_{i+1})$ can be chosen to be a composite function w(σ) of a single variable $\sigma \triangleq b_1\ u_{i-1} + b_2\ u_i + b_3\ u_{i+1}$ with $w(\sigma) \triangleq \{z_2 \pm |[z_1 \pm$

$|z_o + \sigma|]|\}$. This function is used to define the appropriate differential equation for generating the truth table of all 256 Boolean cubes. Thus, each rule of a cellular automaton corresponds to a particular set of *six real numbers* $\{ z_o, z_1, z_2; b_1, b_2, b_3\}$, and *two integers* ± 1. Only eight bits are needed to uniquely specify the differential equation associated with each rule N of a cellular automaton.

It can be proven that once the parameters defining a particular rule N are specified, then for any one of the eight inputs u_{i-1}, u_i, and u_{i+1} listed in the corresponding truth table of N, the solution x_i of the scalar differential equation will either increase monotonically from the initial state $x_i = 0$ towards a *positive equilibrium value* $\bar{x_i}(n) \geq 1$, henceforth denoted by attractor $Q_+(n)$, or decrease monotonically towards a *negative equilibrium state* $\bar{x_i}(n) \leq -1$, henceforth denoted by attractor $Q_-(n)$, when the input (u_{i-1}, u_i, u_{i+1}) is chosen from the coordinates of vertex n of the associated Boolean cube, or equivalently, from row n of the corresponding truth table, for $n = 0, 1, 2, \ldots, 7$ [3]. Vertex n is painted red whenever its equilibrium value $\bar{x_i}(n) \geq 1$, and blue whenever $\bar{x_i}(n) \leq -1$, then the color of all eight vertices for the associated *Boolean cube* will be uniquely specified by the *equilibrium solutions* of the eight associated *differential equations*.

In general, we can summarize: once the parameters associated with a particular rule of a cellular automaton are specified, the corresponding *truth table* or *Boolean cube*, will be uniquely generated by the *scalar differential equation* alone. If the output equation of the dynamical system is $y_i = y(x_i) \triangleq \frac{1}{2}\,(|x_i + 1| - |\, x_i - 1|)$, then $y_i = +1$ when $x_i \geq 1$, and $y_i = -1$ when $x_i \leq -1$. The steady-state output at equilibrium is given explicitly by the formula $y_i = \text{sgn}\,\{w(\sigma)\}$ for any function $w(\sigma) \triangleq w(u_{i-1}, u_i, u_{i+1})$ with signum function $\text{sgn}\,(x) = +1$ for positive numbers x, $\text{sgn}\,(x) = -1$ for negative numbers x and $\text{sgn}\,(0) = 0$.

For the particular $w(\sigma)$ in Fig. 4 the output (color) at equilibrium is given explicitly by the

$$\textit{attractor color code}\text{: } y_i = sgn\,\{z_2 \pm |[z_1 \pm |z_0 + \sigma|]|\}\,.$$

Fig. 4 contains 4 examples of dynamical systems and the rules they encode, each one identified by its rule number $N = 0, 1, 2, \ldots, 255$. The truth table for each rule N is generated by the associated dynamical system defined in upper portion of each quadrant, and not from the truth table, thereby proving that each dynamical system and the rule of the cellular automaton it encodes are one and the same. The truth table for each rule in Fig. 4 is cast in a format with only $2^{2^3} = 256$ distinct 1x3 neighborhood patterns. Each color picture consists of 30 x 61 pixels, generated by a 1-dimensional cellular automaton with 61 cells and a boundary condition with a specific rule N.

As an example, let us examine one of the rules from Fig. 4, rule 110, which will later on be identified as the simplest universal Turing machine known to date. With its differential equation, one can identify $\sigma = b_1\ u_{i-1} + b_2\ u_i + b_3\ u_{i+1}$ with $b_1 = 1$, $b_2 = 2$, and $b_3 = -3$, and $w(\sigma) \triangleq \{z_2 \pm |[z_1 \pm |z_o + \sigma|]|\}$ with $z_2 = -2$, $z_1 = 0$, and $z_o = -1$. Thus, the attractor color code is explicitly given by $y_i = \text{sgn}[-2 + |u_{i-1} + 2u_i - 3u_{i+1} - 1)|]$.

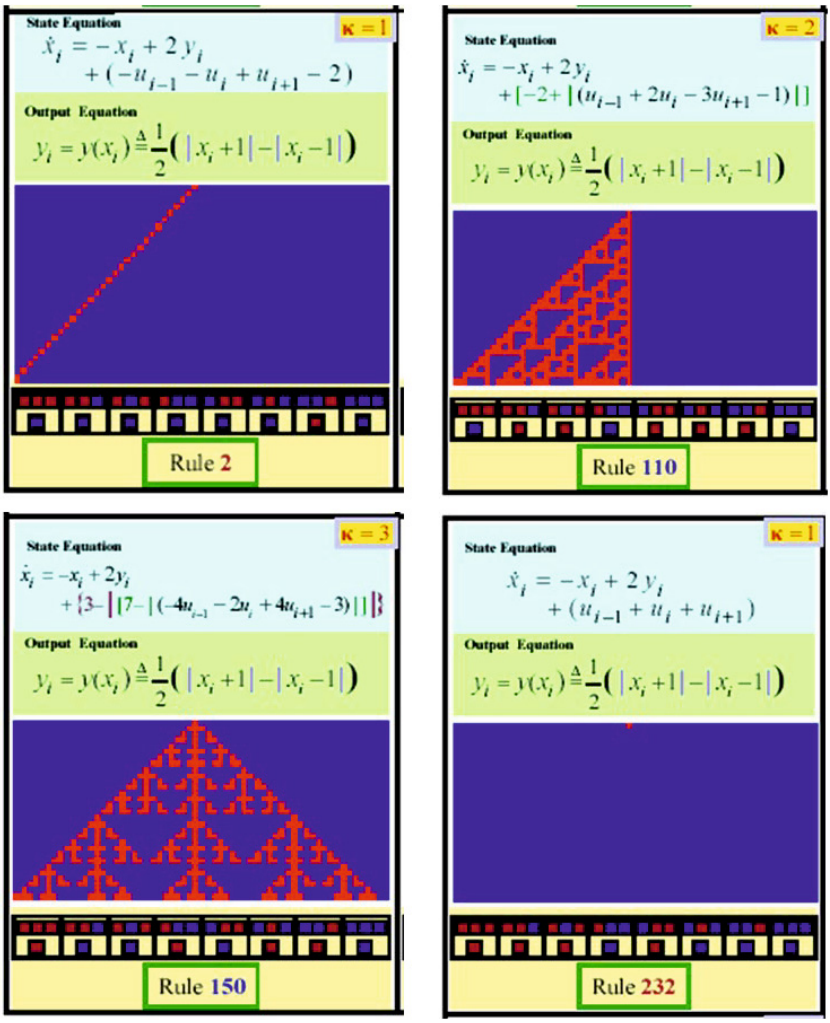


Fig. 4. Cellular automata with rules 2, 110, 150, 232 as dynamical systems. The initial condition is $x(0) = 0$.

1.3 Digital Dynamics with Difference Equations

The dynamics of dynamical systems are modeled with continuous differential equations. For computing the dynamics for digital cellular automata, a program must use a "do loop" instruction which feedbacks the output y_i^t of each cell at iteration t back to its inputs to obtain the output y_i^{t+1} at the next iteration $t+1$. Using the superscripts t and $t+1$ as iteration number from one to the next generation, we can express each rule N explicitly in the form of a *nonlinear difference equation* with

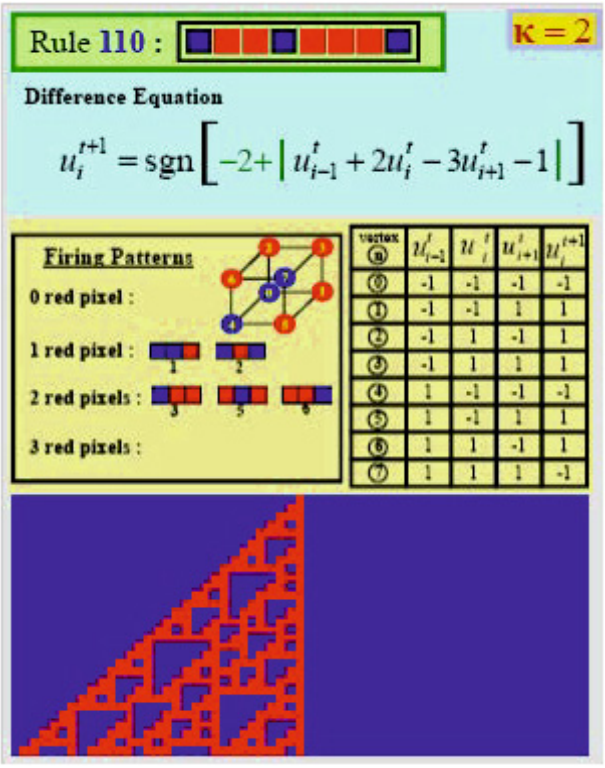


Fig. 5. Cellular automaton as dynamical system with difference equation

$$u_i^{t+1} = sgn\{z_2 + c_2|[z_1 + c_1|(z_0 + b_1 u_{i-1}^t + b_2 u_i^t + b_3 u_{i+1}^t)|]|\},$$

where the *eight parameters* $\{z_o, z_1, z_2; b_1, b_2, b_3; c_1, c_2\}$ are given for each rule. Thus, the first main result is that each of 256 1-dimensional cellular automata which were studied by Stephen Wolfram experimentally can be generated from a single scalar nonlinear differential equation or a corresponding nonlinear difference equation with at most eight parameters. These equation are also universal in the sense of a universal Turing machine (UTM), because we will later on see that at least one of the 256 rules (for example, rule 110) is capable of universal computation [4]. For rule 110 (Fig. 5), we get $u_i^{t+1} = \mathrm{sgn}\ (-2+|u_{i-1}^t + 2\ u_i^t - 3\ u_{i+1}^t - 1|)$. This kind of difference equation can be understood with elementary knowledge in basic mathematics, although it demonstrates important features of nonlinear dynamics.

2 Complexity in the Universe of Cellular Automata

The colored toy cubes contain all information about the complex dynamics of cellular automata. An important advantage of the Boolean cube representation

is that it allows us to define an index of complexity [3]. Each one of the 256 cubes is obviously characterized by different clusters of red or blues vertices which can be separated by parallel planes. On the other hand, the separating planes can be analytically defined in the coordinate system of the Boolean cubes. Therefore, the complexity index of a cellular automaton with local rule N is defined by the minimum number of parallel planes needed to separate the red vertices of the corresponding Boolean cube N from the blue vertices. Fig. 6 shows three examples of Boolean cubes for the three possible complexity indices $\kappa = 1, 2, 3$ with one, two and three separating parallel planes. There are 104 local rules with complexity index $\kappa = 1$. Similarly, there are 126 local rules with complexity index $\kappa = 2$ and only 26 local rules with complexity index $\kappa = 3$. This analytically defined complexity index is to be distinguished from Wolfram's complexity index based on phenomenological estimations of pattern formation.

2.1 Complexity Index of Cellular Automata

In the context of colored cubes of cellular automata, *separability* refers to the number of cutting (parallel) planes separating the vertices into clusters of the same color. For rule 110, for example, we can introduce two separating parallel planes of the corresponding colored cube which are distinguished in Fig.6b by two different colors: The red vertices 2 and 6 lie above a yellow plane. The blue vertices 0, 4, and 7 lie between the yellow and a light blue plane. The red vertices 3, 1, and 5 lie below the light blue plane. It is well-known that the cellular automaton of rule 110 is one of the few types of the 256 automata which are *universal Turing machines*. In the sense of Wolfram's class 3 of computer experiments, it produces very complex patterns [26].

An example of an automaton which can only produce very simple patterns is rule 232. There is only *one separating plane* cutting the corresponding Boolean cube for separating colored points (Fig.6a): Red vertices 3, 5, 6, and 7 lie above a light blue plane. The blue vertices 0, 1, 2, and 4 lie below the light blue plane. A colored Boolean cube with *three parallel separating planes* is shown in Fig. 6c, representing the cellular automaton of rule 150: The blue vertex 6 lies above a green plane. The red vertices 2, 4, and 7 lie between a yellow plane and the green plane. The blue vertices 0, 3, and 5 lie between the yellow plane and a light blue plane. The blue vertex 1 lies below the light blue plane. Obviously, it is not possible to separate the 8 vertices into three colored clusters and at the same time separate them by two parallel planes, no matter how the planes are positioned.

A rule whose colored vertices can be separated by only one plane is said to be *linearly separable*. An examination of the 256 Boolean cubes shows that 104 among them are linearly separable. The remaining 152 rules are not linearly separable. Each rule can be separated by various numbers of parallel planes. In general, there is a unique integer κ, henceforth called the *complexity index* of rule N, which characterizes the geometrical structure of the corresponding Boolean cube, namely the minimum number of parallel planes that is necessary to separate the colored vertices. All linearly separable rules have a *complexity*

index κ=1. An analysis of the remaining 152 linearly non-separable rules shows that they have a complexity index of either 2 or 3. For example, rule 110 has a complexity index κ=2 whereas rule 150 has a complexity index κ=3. No rule with complexity index κ=1 is capable for generating complex patterns, even for random initial conditions. The emergence of complex phenomena significantly depends on a minimum complexity of κ=2. In this sense, complexity index 2 can be considered the threshold of complexity for 1-dimensional cellular automata.

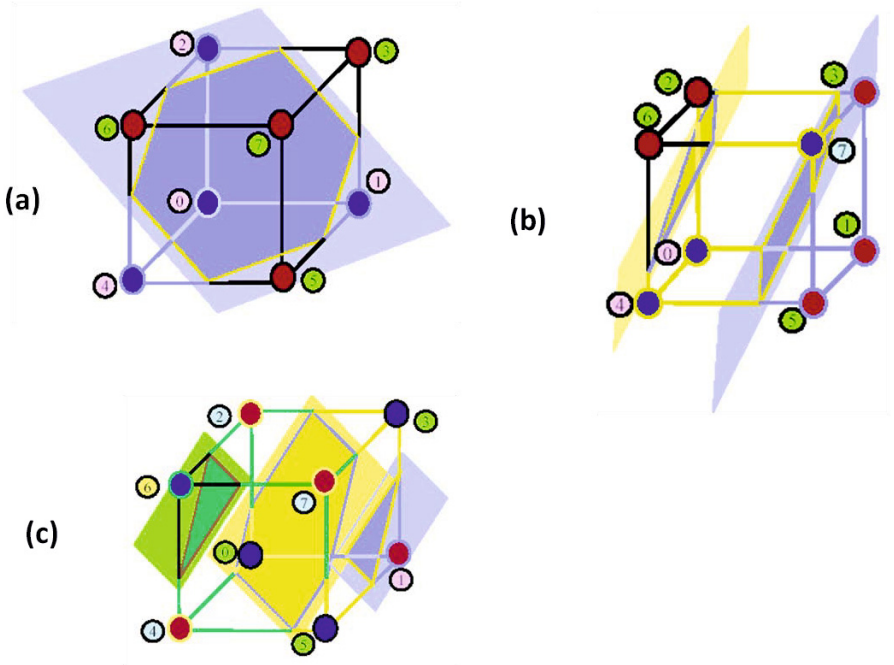


Fig. 6. Examples of complexity index $\kappa = 1, 2, 3$ with parallel planes separating all vertices having one color from those having a different color on the other side for rule 232 (**a**), rule 110 (**b**),and rule 150 (**c**)

2.2 Computational Complexity and Universal Computability

A motivation for the introduction of a complexity index is also *computational complexity*. The class of cellular automata with complexity index $\kappa = 2$ contains examples with universal computation (e.g., $N = 110$), but the local rules with complexity index $\kappa = 1$ are not capable of universal computation. It follows that $\kappa = 2$ also represents a threshold of computational complexity.

Universal computation is a remarkable concept of computational complexity which dates back to Alan Turing's universal machine [17]. Universal cellular automata are well-known since Conway's game of life [13]. A *universal Turing machine* can by definition simulate any Turing machine. According to the

Church-Turing thesis, any algorithm or effective procedure can be realized by a Turing machine. Now Turing's famous Halting problem comes in. Following his proof, there is no algorithm which can decide for an arbitrary computer program and initial condition if it will stop or not in the long run. (A computer program cannot stop if it must follow a closed loop.) Consequently, for a system with universal computation (in the sense of a universal Turing machine), we cannot predict if it will stop in the long run or not. Assume that we were able to do that. Then, in the case of a universal Turing machine, we could also decide whether any Turing machine (which can be simulated by the universal machine) would stop or not. That is obviously a contradiction to Turing's result of the *Halting problem*. Thus, systems with universal computation are unpredictable.

Unpredictability is obviously a high degree of complexity. It is absolutely surprising that systems with simple rules of behavior like cellular automata lead to complex dynamics which is no longer predictable. We will be very curious to discover examples of these, in principle, unpredictable automata in nature.

3 Symmetry in the Universe of Cellular Automata

A cursory inspection of the discrete time evolutions of the 256 local rules reveals some similarity and partial symmetry among various evolved patterns. It reminds us of more or less random observations in the natural sciences demanding for unifying mathematical explanations with fundamental laws. The unifying theory of physics is based on the assumption of fundamental mathematical symmetries [8, 9]. According to this view, the variety and complexity of natural phenomena have evolved from some few principles of symmetry. They are the "*Holy Grail*" of the Universe which is sought by prominent scientists and research groups all over the world. For the universe of cellular automata, we found the fundamental symmetries in the gallery of Boolean cubes [5]. Thus, at least in the toy world of cellular automata, the importance of symmetry laws can easily be imagined and understood.

3.1 Local Equivalence of Cellular Automata

But, even in the universe of cellular automata, the situation is sophisticated. The Boolean cubes of many different pairs of local rules seem to be related by some *symmetry transformations*, such as complementation of the vertex colors (e.g., rules 145 and 110). Yet, their evolved patterns are so different that it is impossible to relate them. How do we make sense of all these observations? In the case of rule 145 and 110, the associated Boolean cubes are related by a "red - blue vertex transformation". It is denoted as *local complementation operation* $\mathbf{T}^C$, because complementation is locally restricted. Intuitively, one might expect that their respective evolved patterns must also be related by a global complementation operation. But the intuition turns out to be wrong in general, upon comparing the two evolved patterns (Fig. 7). It is only true in a local sense with respect to special iterations. For example, starting from the same initial pattern (single red

center pixel) in the first row, we find the output (first iteration) of rule 145 is in fact the complement of that of rule 110; namely, two blue pixels for 145 and two red pixels for 110 at corresponding locations to the left of center. All other pixels at corresponding locations are also complements of each other.

However, the next iteration (row 3) under rules 145 and 110 in Fig. 7 are not complements of each other. The reason is that unlike the initial input u_i^0, $i = 0, 1, 2, \ldots, n$, which are the same for both 145 and 110, the next input u_i^1, $i = 0, 1, 2, \ldots, n$ (for $t = 1$ in row 2) needed to find the next iteration (row 3) are different and there is no reason for the output u_i^2 (for $t = 2$) at corresponding locations to be the complement of each other. In these cases, a pair of local rules is equivalent only in a local sense with respect to "*local in iteration time*", and not local in the usual sense of a spatial neighborhood.

In general, we define

Local Equivalence: Two local rules N and N' are said to be *locally equivalent* under a transformation $\mathbf{T} : N \to N'$ iff the output u_i^1 of N after one iteration of any initial input pattern u_i^0 can be found by applying the transformed input $\mathbf{T}(u_i^0)$ to rule N' and then followed by applying the *inverse transformation* $\mathbf{T}^{-1} : N' \to N$ to u_i^1.

3.2 Global Equivalence of Cellular Automata

Global aspects can be observed in the evolved patterns for the rules 110, 137 (Fig. 7), 124, and 193 (Fig. 8). Despite the fact that the respective Boolean cubes of these three rules do not seem to be related in an obvious way, their output patterns are so precisely related that one could predict the evolved pattern over all times t of each local rule 110, 124, 137, and 193. For example, the evolved output pattern of rule 124 can be obtained by a reflection of that of 110 about the center line, namely a bilateral transformation. The output of rule 193 can be obtained by applying the complement of u_i^0 (i.e. blue center pixels amidst a red background) to rule 110 and then taking the complement of the evolved pattern from 110. The output of rule 137 can be obtained by repeating the above algorithm for 193, and then followed further by a reflection about the center line. It can be proved that these algorithms remain valid for all initial input patterns. This result is most remarkable because it allows us to *predict* the evolved patterns from arbitrary initial configurations of three rules over all iterations and not just for one iteration as in the case of local equivalence.

In general, we define

Global Equivalence: Two local rules N and N' are said to be *globally equivalent* under a transformation $\mathbf{T} : N \to N'$ iff the output x_i^t of N can be found, for any t, by applying the transformed input $\mathbf{T}(x_i^0)$ to local rule N' and then followed by applying the *inverse transformation* $\mathbf{T}^{-1} : N' \to N$ to x_i^1, for any $t = 1, 2, \ldots$.

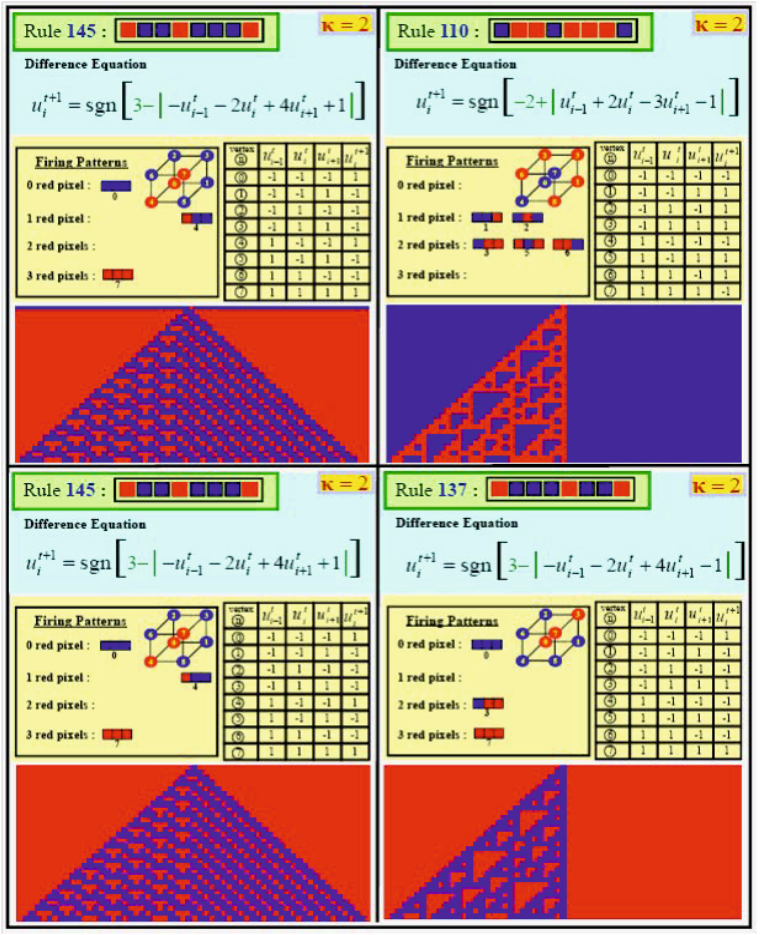


Fig. 7. The evolutions of rules 110 and 145 only reveal a local complement relationship in the first iteration, but 110 and 137 reveal global symmetrical relationship

Obviously, the four rules 110, 124, 137, and 193 are globally equivalent in the sense that the evolved patterns of any three members of this class can be trivially predicted from the fourth for all iterations. Therefore, these four rules have identical nonlinear dynamics for all initial input patterns and therefore they represent only one generic rule, henceforth called an *equivalence class*. This global property is not only true for four rules, but also for all rules, thereby allowing us to partition the 256 rules into only 88 global equivalence classes. It is convenient to identify these equivalence classes with the symbol ε_m^κ, where κ is the *complexity index* and m the *class number*. There are 38 cellular automata belonging to the equivalence classes ε_m^1 with complexity index $\kappa = 1$ and $m = 1, 2, \ldots, 38$. The equivalence classes ε_m^2 with complexity index $\kappa = 2$ are distinguished by $m = 1, 2, \ldots, 41$. Further on, there are nine global equivalence

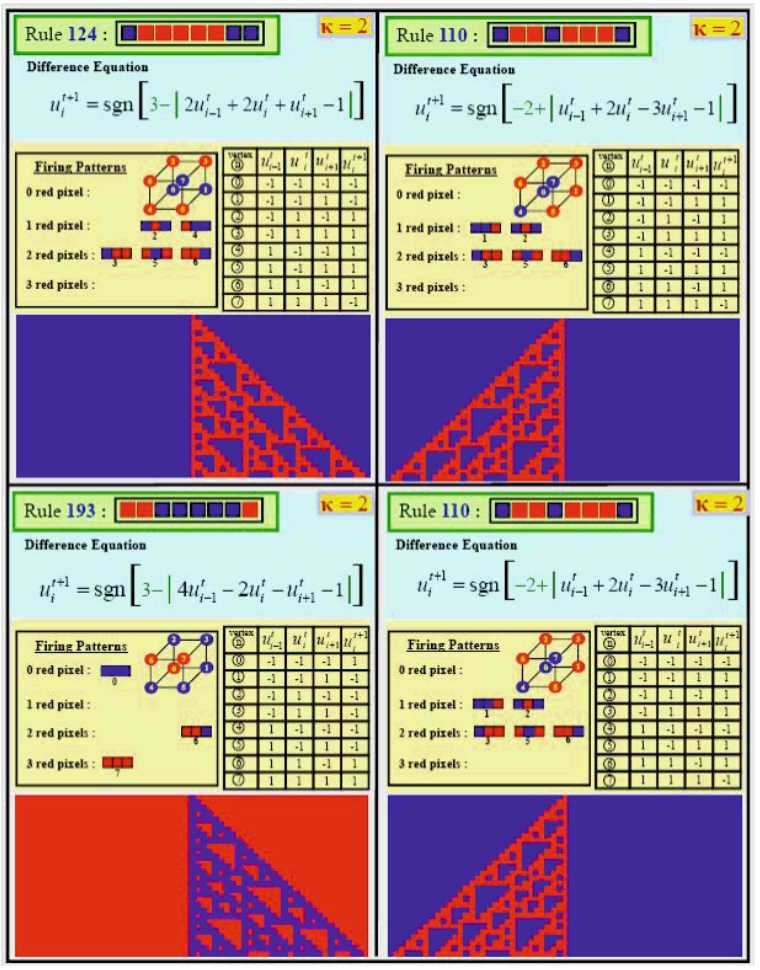


Fig. 8. The evolutions of rules 110, 124, 193 reveals global symmetrical relationships

classes with complexity index $\kappa = 3$. They are identified by ε_m^3 with $m = 1, 2, \ldots, 9$.

This result is significant because it asserts that one only needs to study in depth the dynamics and long-term behaviors of 88 representative local rules. Moreover, since 38 among these 88 dynamically distinct rules have complexity index $\kappa = 1$, and are therefore trivial, we are left with only 50 local rules (41 rules with $\kappa = 2$ and 9 rules with $\kappa = 3$) that justify further in-depth investigations.

3.3 Symmetry with Global Transformations

It can be proven that every local rule belongs to a global equivalence class determined by certain global transformations. There are three global transformations,

namely, *global complementation* $\overline{\mathbf{T}}$, *left-right complementation* $\mathbf{T}^{\star}$, and *left-right transformation* $\mathbf{T}^{\dagger}$ which are distinguished as *symmetry transformations* in the universe of cellular automata. The four rules 110, 124, 137, and 193 are globally equivalent to each other in the sense that their long term (as $t \to \infty$) dynamics are mathematically identical with respect to the three global transformations $\mathbf{T}^{\dagger}$, $\mathbf{T}^{\star}$, and $\overline{\mathbf{T}}$.

The intuitive meaning of these symmetry transformations can easily be seen in Fig. 9. In this picture, all four patterns of rules 110, 124, 137, and 193 have 60 rows corresponding to iterations numbers $t = 0, 1, 2, \ldots, 59$, and 61 columns, corresponding to 61 cells ($n = 60$). All patterns have a random initial condition ($t = 0$), or its reflection, complementation, or both. The two patterns 124 and 110 on top are generated by a left-right transformation $\mathbf{T}^{\dagger}$, and are related by a bilateral reflection about an imaginary vertical line situated midway between the two patterns. The two patterns 193 and 137 below are likewise related via $\mathbf{T}^{\dagger}$ and exhibit the same bilateral reflection symmetry. The two vertically situated local rules 137 and 110, as well as 193 and 124 are related by a global complementation $\overline{\mathbf{T}}$. The two diagonally-situated local rules 124 and 137, as well as 193 and 110 are related by a left-right complementation $\mathbf{T}^{\star}$.

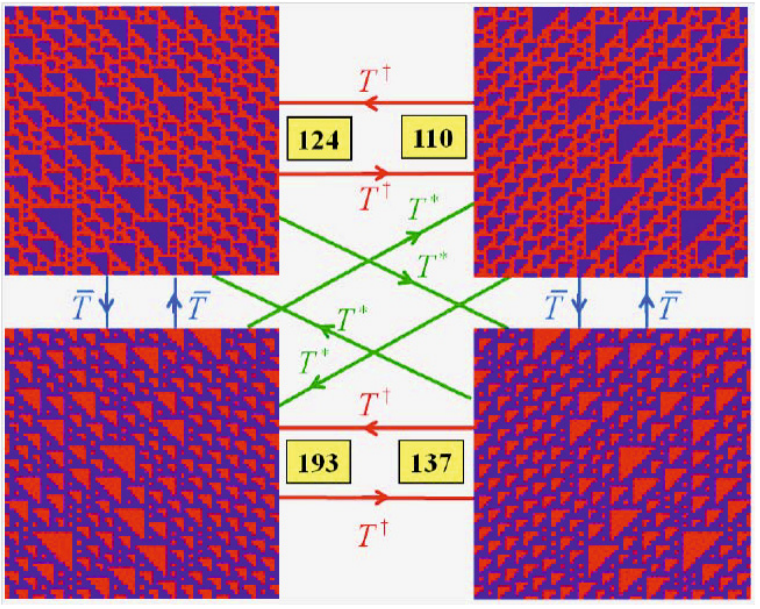


Fig. 9. Global equivalence of rules 110, 124, 137, and 193

The *geometrical definition* of these symmetry transformations is easy to understand and can even be imagined with help of our toy cubes of cellular automata. Mathematically, these transformations are defined by 3 x 3 *matrices* $\overline{\mathbf{T}}_{\mathbf{u}}$, $\mathbf{T}^{\star}_{\mathbf{u}}$, and $\mathbf{T}^{\dagger}_{\mathbf{u}}$. Each of the three matrices transforms the three axes (u_{i-1},

u_i, u_{i+1}), drawn through the center of the *Boolean cube* into a transformed set of axes (u'_{i-1}, u'_i, u'_{i+1}). These matrix representations also only need basic mathematics. An analytical definition is given in [12].

3.4 Global Symmetry of Klein's Vierergruppe V

The three global transformations $\mathbf{T}^\dagger$, $\mathbf{T}^\star$, and $\overline{\mathbf{T}}$ are generated from elements of the classic *noncyclic four-element Abelian group* $\mathbf{V}$, originally called the "*Vierergruppe*" by the German mathematician Felix Klein [16]. The four elements of $\mathbf{V}$ are denoted by the 3 x 3 matrices $\mathbf{T}_0$, $\overline{\mathbf{T}}_\mathbf{u}$, $\mathbf{T}^\star_\mathbf{u}$, and $\mathbf{T}^\dagger_\mathbf{u}$. The symbol $\mathbf{T}_0$ denotes the identity, or unit matrix, of any dimension. The actual transformations, however, that allow us to establish the long-term correlations among members of each of the 88 *global equivalence classes* of all 256 cellular automata are the 4 x 4 matrices $\mathbf{T}_0$, $\mathbf{T}^\dagger$, $\mathbf{T}^\star$, and $\overline{\mathbf{T}}$. Fig. 10 shows that they are related by the group multiplication table of Klein's Vierergruppe $\mathbf{V}$. This is the only abstract mathematical group which makes it possible to predict the *long-term correlations* among all members of the four remarkable rules 110, 124, 137, and 193.

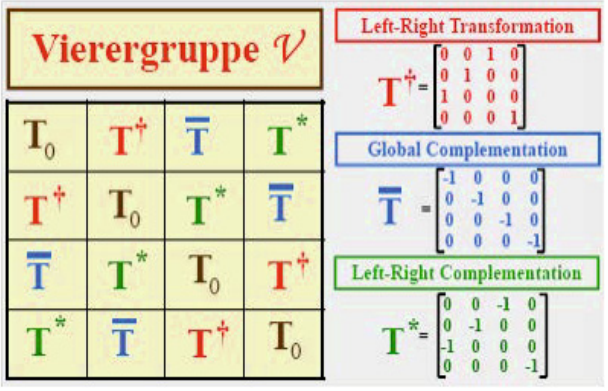


Fig. 10. Global Symmetry and Klein's Vierergruppe

These results are global in the sense of asymptotic time behavior as $t \to \infty$. It proves that even though there are 256 distinct local rules of 1-dimensional cellular automata, there are only 88 distinct global behaviors, a fundamental result predicted by the identification of 88 global equivalence classes ε^κ_m.

3.5 The Holy Grail of Symmetry and Computability

Since the local rule 110 has been proved to be capable of *universal computation*, it follows that all four local rules of the Vierergruppe $\mathbf{V}$ are *universal Turing machines*. The fundamental importance of the universality result was to exploit the *symmetry* of the Boolean cubes in order to identify equivalence classes among

the 256 rules. The discovery of the *Vierergruppe* **V** and the *rotation group* **R** had led to the major logical classifications of the 256 local rules into 88 *global equivalence classes* ε_m^κ and 30 *local equivalence classes* S_m^κ . The significance of the 88 global equivalence classes ε_m^κ is similar to the classification of computational algorithms into various complexity classes, for example, the *N*- or *NP*-classes, in the sense that any property that applies to one member of ε_m^κ applies to the other members in the same global equivalence class.

The *universality* of the four rules 110, 124, 137, and 193 and their identical long-term dynamic behaviors, with respect to the symmetry transformations of the Vierergruppe **V**, are encapsulated in the commutative diagram shown in Fig. 11. Thus, Klein's Vierergruppe represents the fundamental symmetry law of the 256 two-state one-dimensional cellular automata. It is the "*Holy Grail*" of a unified theory in the universe of these cellular automata, containing all information about their nonlinear dynamics.

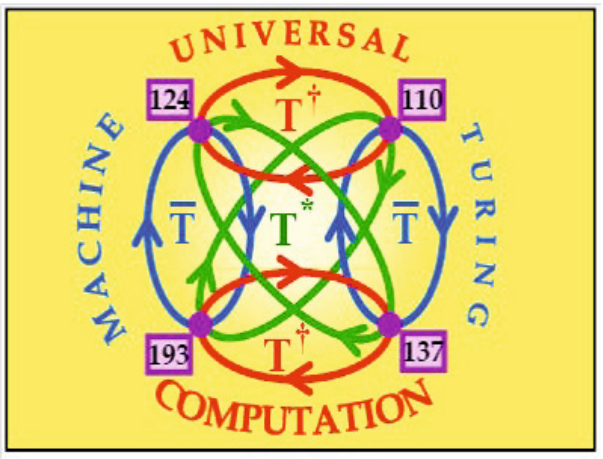


Fig. 11. Universal symmetry and computability in the universe of cellular automata

4 Outlook to a Computational Universe of Dynamical Systems

One-dimensional cellular automata with $L = I+1$ cells are *complex systems* with *nonlinear dynamics* [1, 11, 15] determined by one of the 256 local rules *N*. Their *state spaces* contain all distinct states of cellular rows $(x_0^t, \ldots, x_{I-1}^t, x_I^t)$ at step t of time (iteration or generation). An entire list of consecutive rows with no two rows identical and including the initial configuration is called an *orbit* in the state space of a cellular automaton. On that background, the well-known attractor dynamics of complex systems can also be studied in the theory of cellular automata [12].

Summing up all these insights, we are on the way to conceive the universe as an automaton and dynamical system. The success of this research program depends on the digitization of physics. The question "*Is the Universe a computer*" leads to the question: How far is it possible to map the laws of physics onto computational digital physics? [6] Digitization is not only exciting for answering philosophical questions of the universe. *Digitization* is the key paradigm of modern research and technology. Nearly all kind of research and technical innovation depend on computational modeling. The emerging complexity of nature and society cannot be handled without computers with increasing computational power and storage.

In order to make this complex computational world more understandable, cellular automata are an excellent tool. NKS and our analytical approach show that many basic principles of the expanding universe and the evolution of life and brain can be illustrated with cellular automata. The emergence of new structures and patterns depends on phase transitions of complex dynamical systems in the quantum, molecular, cellular, organic, ecological, and societal world [10]. Cellular automata are recognized as an intuitive modeling paradigm for complex systems with many useful applications [7]. In cellular automata, extremely *simple local interactions* of cells lead to the *emergence of complex global structures*. This *local principle of activity* is also true in the world of complex systems with elementary particles, atoms, molecules, cells, organs, organisms, populations, and societies [2]. Although local interactions generate a complex variety of being in the universe, they can be mathematically reduced to some fundamental laws of symmetry.

Symmetries play a key role in the physical world as well as in the universe of automata. In philosophy of science, they have been considered *universal principles* of *Platonic truth* and *beauty* [8]. The scientific search for symmetries reminds us of Parsifal's quest for the Holy Grail. The legend of Parsifal was written by the minnesinger Wolfram von Eschenbach (c. 1170–c. 1220). It may be a random accord of names that a "Wolfram" also wrote "A New Kind of Science" for cellular automata. In the 19^{th} century, Richard Wagner composed his famous opera based on Wolfram's legend of Parsifal. In Wagner's interpretation, it is the quest of the "poor fool" Parsifal for the Holy Grail. The question is still open whether the scientific search for a final symmetry or "world formula" will also be a "foolish" quest.

References

[1] Alligood, K.T., Sauer, T.D., Yorke, J.A.: Chaos: An Introduction to Dynamical Systems. Springer, New York (1996)
[2] Chua, L.O.: CNN: A Paradigm for Complexity. World Scientific, Singapore (1998)
[3] Chua, L.O., Yoon, S., Dogaru, R.: A nonlinear dynamics perspective of Wolfram's new kind of science. Part I: Threshold of complexity. International Journal of Bifurcation and Chaos (IJBC) 12(12), 2655–2766 (2002)
[4] Chua, L.O., Sbitnev, V.I., Yoon, S.: A nonlinear dynamics perspective of Wolfram's new kind of science. Part II: Universal neuron. International Journal of Bifurcation and Chaos (IJBC) 13(9), 2377–2491 (2003)

[5] Chua, L.O., Sbitnev, V.I., Yoon, S.: A nonlinear dynamics perspective of Wolfram's new kind of science. Part III: Predicting the unpredictable. International Journal of Bifurcation and Chaos (IJBC) 14, 3689–3820 (2004)
[6] Deutsch, D.: Quantum theory, the Church-Turing principle and the universal quantum computer. Proceedings of the Royal Society of London A 400, 97–117 (1985)
[7] Hoekstra, A.G., Kroc, J., Sloot, P.M.A. (eds.): Simulating Complex Systems by Cellular Automata. Springer, Berlin (2010)
[8] Mainzer, K.: Symmetries of Nature. De Gruyter, New York (1996) (German 1988: Symmetrien der Natur. De Gruyter, Berlin)
[9] Mainzer, K.: Symmetry and Complexity: The Spirit and Beauty of Nonlinear Science. World Scientific, Singapore (2005)
[10] Mainzer, K.: Thinking in Complexity. The Computational Dynamics of Matter, Mind, and Mankind, 5th edn. Springer, Berlin (2007)
[11] Mainzer, K. (ed.): Complexity. European Review (Academia Europaea) 17(2), 219–452 (2009)
[12] Mainzer, K., Chua, L.O.: The Universe as Automaton. From Simplicity and Symmetry to Complexity. Springer, Berlin (2011)
[13] Martin, B.: A universal cellular automaton in quasi-linear time and its S-m-n form. Theoretical Computer Science 123, 199–237 (1994)
[14] Rendell, P.: A Turing machine in Conway's Game of Life, extendable to a universal Turing machine. In: Adamatzky, A. (ed.) Collision-Based Computing. Springer, New York (2002)
[15] Shilnikov, L., Shilnikov, A., Turaev, D., Chua, L.: Methods of Qualitative Theory in Nonlinear Dynamics I-II. World Scientific, Singapore (1998-2001)
[16] Speiser, A.: Die Theorie der Gruppen von endlicher Ordnung, 4th edn. Birkhauser, Basel (1956)
[17] Turing, A.M.: On computable numbers with an application to the Entscheidungs problem. Proceeedings of the London Mathematical Society 2(42), corrections, ibid 43, 544–546 (1936-1937)
[18] Wolfram, S.: A New Kind of Science. Wolfram Media, Inc., Champaign Il (2002)

Figures (IJBC = International Journal of Bifurcation and Chaos)

Fig. 1: IJBC 2008 vol. 18, no. 9, p. 2490, Fig. 1a-d
Fig. 2: IJBC 2008 vol. 18, no. 9, p. 2496, Fig. 3a-d
Fig. 3: IJBC 2003 vol. 13, no. 9, p. 2378, Fig. 1b
Fig. 4: IJBC 2002 vol. 12, no. 12, Table 2, p. 2666 (rule 2), 2693 (rule 110), 2703 (rule 150), 2724 (rule 232)
Fig. 5: IJBC 2003 vol. 13, no. 9, p. 2417, Table 5 (rule 110)
Fig. 6a: IJBC 2002 vol. 12, no. 12, p. 2749, Fig. 14
Fig. 6b: IJBC 2002 vol. 12, no.12, p. 2742, Fig. 9
Fig. 6c: IJBC 2002 vol. 12, no. 12, p. 2746, Fig. 12
Fig. 7: IJBC 2004 vol. 14, no. 11, p. 3697, Fig. 5
Fig. 8 : IJBC 2004 vol. 14, no. 11, p. 3699, Fig. 6
Fig. 9 : IJBC 2004 vol. 14, no. 11, p. 3700, Fig. 7
Fig. 10: IJBC 2004 vol. 14, no. 11, p. 3818, Fig. 17a
Fig. 11: IJBC 2004 vol. 14, no. 11, p. 3818, Fig. 17b

Chapter 6
A New Kind of Science: Ten Years Later

David H. Bailey*

Lawrence Berkeley National Laboratory, Berkeley, CA, USA
DHBailey@lbl.gov

Abstract. It has been ten years since Stephen Wolfram published his magnum opus *A New Kind of Science* [13]. It is worth re-examining the book and its impact in the field.

1 Highlights of ANKS

The present author personally read the book with great interest when it was first published. Of particular interest then and now are the many illustrations, particularly those of the complex patterns generated by certain cellular automata systems, such as rule 30 and rule 110, as contrasted with the very regular patterns produced by other rules. In this regard, graphical analyses of these cellular automata rules join a select group of modern mathematical phenomena (including, for example, studies of the Mandlebrot set, chaotic iterations and certain topological manifolds) that have been studied graphically as well as analytically.

In looking again at ANKS, the present author is struck today, as in 2002, with the many interesting items in the endnote section, which occupies 350 pages of two-column, small-font text. In some respects, the endnotes of ANKS constitute an encyclopedia of sorts, covering, in concise yet highly readable form, historical background, mathematical foundations and scientific connections of a wide variety of topics related to modern-day computing. Much of this material remains as cogent and interesting today as when it was written over ten years ago. Some of the particularly interesting endnote passages are the following:

1. Wolfram's entry on "History of experimental mathematics" (pg. 899) contains a number of interesting insights on the practice of using computers as exploratory tools in mathematics.
2. In his entry "Randomness in markets" (pg. 1014), Wolfram notes that in modern-day financial markets, large price fluctuations are significantly more common than a Gaussian distribution would imply—a phenomenon amply affirmed in the 2007–2009 worldwide financial crash.
3. Under "Einstein equations" (pg. 1052–1054), Wolfram presents a brief and yet informative introduction to the equations of general relativity, together with some interesting computational perspectives in the general arena of relativity and cosmology.

* Supported in part by the Director, Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy, under contract number DE-AC02-05CH11231.

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 67–76.
DOI: 10.1007/978-3-642-35482-3_6

4. Under "Quantum phenomena" (pg. 1056–1065), Wolfram presents an even more detailed overview of quantum mechanics, including an introduction to Feynman diagrams, quantum field theory and Bell's inequality, all with interesting connections to computation.
5. Under "Data Compression" (pg. 1069–1074), Wolfram includes numerous details of state-of-the-art data compression algorithms.
6. Beginning with a series of articles under the heading "Undecidability and Intractibility" (pg. 1136–1149), Wolfram presents a fairly technical but nonetheless quite coherent introduction to many of the topics of modern theoretical computer science, including undecidability, computational complexity, Turing machines, NP-completeness and quantum computers.
7. Immediately following the material on theoretical computer science is a similarly detailed introduction (pg. 1149–1177) to modern developments in the foundations of mathematics, with implications for computing.
8. In a fascinating section "Intelligence in the Universe" (pg. 1177–1191), Wolfram discusses such topics as the origin of life, extraterrestrial life, the nature of animal and human intelligence, Fermi's paradox and speculations, from a computational point of view, as to why our search for extraterrestrial intelligence so far as been unsuccessful.

2 Experimental Mathematics

With regards to item 1 in the list above (experimental mathematics), it is worth pointing out that while Wolfram is very well versed in rigorous mathematical proof, nonetheless he confesses (pg. 899), "by now I have come to trust the correctness of conclusions based on simple systematic computer experiments much more than I trust all but the simplest proofs." Wolfram laments the fact that so few others in the field of modern-day mathematics are willing embrace this computational-experimental paradigm (pg. 899):

> [E]ven now, unlike essentially all other branches of science, mainstream mathematics continues to be entirely dominated by theoretical rather than experimental methods. And even when experiments are done, their purpose is essentially always just to provide another way to look at traditional questions in traditional mathematical systems.

Fortunately, this is one arena where substantial progress has been made in the past ten or fifteen years. Nowadays many research mathematicians use the computer to manipulate symbolic expressions, generate sequences, visually inspect numerical data, check analytical work, compute expressions to very high numeric precision, and otherwise explore the mathematical universe using the latest computer technology. This trend has been greatly facilitated by continuing improvements in *Mathematica* and other mathematical software packages. Just as important for this phenomenon is the entry into the field of a large number of junior-level mathematicians who are completely comfortable with computer-based tools, and who instinctively look to the computer as the first step in investigating a mathematical question.

3 A New Formula for Pi

Perhaps one of the most interesting of the recent computer-discovered mathematical facts is what is now known as the "BBP" formula for $\pi = 3.14159\ldots$:

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right), \tag{1}$$

which can also be written

$$\pi = 4\sum_{k=0}^{\infty} \frac{1}{16^k(8k+1)} - 2\sum_{k=0}^{\infty} \frac{1}{16^k(8k+4)} - \sum_{k=0}^{\infty} \frac{1}{16^k(8k+5)} - \sum_{k=0}^{\infty} \frac{1}{16^k(8k+6)}. \tag{2}$$

Here $\sum$ is the usual mathematical notation for summation. Thus formula (2), for instance, is merely shorthand for

$$\begin{aligned} \pi = 4 &\left(1 + \frac{1}{16(8+1)} + \frac{1}{16^2(2\cdot 8+1)} + \frac{1}{16^3(3\cdot 8+1)} + \cdots \right) \\ -2 &\left(\frac{1}{4} + \frac{1}{16(8+4)} + \frac{1}{16^2(2\cdot 8+4)} + \frac{1}{16^3(3\cdot 8+4)} + \cdots \right) \\ - &\left(\frac{1}{5} + \frac{1}{16(8+5)} + \frac{1}{16^2(2\cdot 8+5)} + \frac{1}{16^3(3\cdot 8+5)} + \cdots \right) \\ - &\left(\frac{1}{6} + \frac{1}{16(8+6)} + \frac{1}{16^2(2\cdot 8+6)} + \frac{1}{16^3(3\cdot 8+6)} + \cdots \right). \end{aligned} \tag{3}$$

These infinite series converge quite rapidly—if one adds up just the first four terms displayed above for each series (i.e., truncating the sums at the $\cdots$ sign), the result will be a value of π correct to six digits.

However, the most remarkable feature of BBP formula is that it permits one to calculate a string of binary (base 2) or hexadecimal (base-16) digits of π beginning at an arbitrary position n, without needing to calculate any of the preceding $n-1$ digits. See [4], [9] or [10, pg. 118–125] for details. Indeed, a *Mathematica* implementation of this surprisingly simple scheme is presented in ANKS (pg. 912).

Recently Tsz-Wo Sze of Yahoo! Cloud Computing demonstrated a closely related variant of this scheme by calculating binary digits of π beginning at position *two quadrillion* [7]. The first 25 binary digits beginning at this point are: 01110011011000000100101001.

An even more interesting aspect of the BBP formula, one particularly relevant to the present discussion, is the fact that it was discovered by a computer. Indeed, it may be the first instance in the history of mathematics where a significant new formula for π was found by a computer. This all happened in 1995, when Canadian mathematician Peter Borwein was considering whether or not it was possible to calculate the n-th digit of a mathematical constant such as π by some

shortcut that avoided the necessity of computing all digits up to and including the n-th digit. He and Simon Plouffe found a way to compute the n-th binary digit of the natural logarithm of two, namely $\ln 2 = 0.693147\ldots$, by manipulating the following well-known formula for $\ln 2$:

$$\ln 2 = \sum_{k=1}^{\infty} \frac{1}{k2^k} = \frac{1}{2} + \frac{1}{3 \cdot 2^3} + \frac{1}{4 \cdot 2^4} + \frac{1}{5 \cdot 2^5} + \cdots . \tag{4}$$

After this discovery, Borwein and Plouffe immediately asked whether they could do the same mathematical "trick" for π. It all depended on finding a similar formula for π. Peter Borwein, who was very familiar with the mathematical literature regarding π, was not aware of any such formula for π, and it seemed exceedingly unlikely that such a formula would have escaped detection by the many thousands of great mathematicians who have studied π through the ages. But Plouffe embarked on a computer search for such a formula, using a 200-digit computer implementation (provided by the present author) of mathematician-sculptor Helaman Ferguson's integer relation "PSLQ" algorithm, which finds integer linear relations among an input set of numerical values. After several months of fits and starts, Plouffe and his computer found formula (1). The rest, as they say, is history.

Since 1995, researchers have discovered similar digit-calculating formulas for numerous other fundamental constants of mathematics, in most cases by similar computer searches using the PSLQ algorithm. See [10, Chap. 3] or [7] for details.

4 Ramanujan's Continued Fraction

Srinivasa Ramanujan (1887–1920), born to a poor family in India, learned mathematics mostly by studying math books on his own. His genius was recognized by British mathematician G. H. Hardy, who invited him to come work with him in Cambridge. Ramanujan's mathematical achievements have been recognized as among the greatest of all time, in spite of the fact that he died at the tender age of 32.

One of the many topics that he addressed in his notebooks is the following class of continued fractions. Given $a, b, \eta > 0$, define

$$R_\eta(a,b) = \cfrac{a}{\eta + \cfrac{b^2}{\eta + \cfrac{4a^2}{\eta + \cfrac{9b^2}{\eta + \ddots}}}}. \tag{5}$$

This complicated-looking expression simply means to evaluate the indicated compound fraction out to some level, and then take the limit as more and more terms are included. Ramanujan discovered the beautiful fact that

$$\frac{R_\eta(a,b) + R_\eta(b,a)}{2} = R_\eta\left(\frac{a+b}{2}, \sqrt{ab}\right), \tag{6}$$

for certain $a, b > 0$ parameterized by elliptic functions. And indeed this is true for all $a, b > 0$, as Berndt notes in his annotation of Ramanujan's notebook [8].

Just as ANKS was being completed, a group of mathematicians (including Jonathan Borwein, Richard Crandall, David Borwein, Raymond Mayer and others) applied the tools of experimental mathematics to study these continued fractions. This started with a simple attempt to numerically validate (6), which, in turn, meant numerically computing formula (5).

Unfortunately, a first attempt to numerically compute $R_1(1,1)$, as a prototype problem, failed miserably—after a lengthy computation only three reliable digits were produced: $0.693\ldots$. But researchers recognized this value as close to the value of $\ln 2 = 0.693147\ldots$, and then discovered that convergence of the continued fraction is worst when $a = b$ (i.e., the initial problem they selected was a poor choice).

Eventually a number of very interesting results were obtained, including an algorithm to compute (5) in the complex plane, and so to determine exactly when it converged. For instance, with the help of *Maple* and *Mathematica* and a scatter plot, these researchers discovered that the fraction converges and (6) holds exactly when (a, b) lie in the cardioid defined by $|a| + |b| \geq 2\sqrt{|a||b|}$. They then determined an elliptic function representation from which the simple formula

$$\mathcal{R}_1(a, a) = 2 \int_0^1 \frac{t^{1/a}}{1 + t^2}\, dt, \tag{7}$$

true for all nonzero real numbers a, followed easily. No such formula is known for $\mathcal{R}_\eta(a, b)$ with $a \neq b$ or not real, although striking results have been obtained in cases such as $\mathcal{R}_\eta(ia, ia)$, for real a, which exhibits true chaos [2].

Study of convergence of these Ramanujan continued fractions was facilitated by reducing them to the following discrete dynamical system: Given complex numbers a and b as in (5), set $t_0 = 1, t_1 = 1$ and then iterate

$$t_n := \frac{1}{n} + \omega_{n-1}\left(1 - \frac{1}{n}\right) t_{n-2}, \tag{8}$$

where $\omega_n = a^2$ or b^2, depending on whether n is even or odd. It can be shown that $\mathcal{R}_\eta(a, b)$ diverges if and only if the sequence $\sqrt{n}t_n$ remains bounded.

If one studies this iteration based solely on its numerical values, nothing much is evident—one only sees that $t_n \to 0$ fairly slowly. However, if one looks at this iteration pictorially, significantly more can be learned. In particular, if one plots these iterates in the complex plane, scaled by $\sqrt{n}$, with iterations colored blue or red depending on odd or even n, then some remarkable fine structure appears—see Figure 1.

With assistance of such plots, the behavior of these iterates (and the Ramanujan continued fractions themselves) is now quite well understood. A *Cinderella* applet exploring the dynamics of these iterations is available at `http://carma.newcastle.edu.au/jon/dynamics.html`. When a and b are

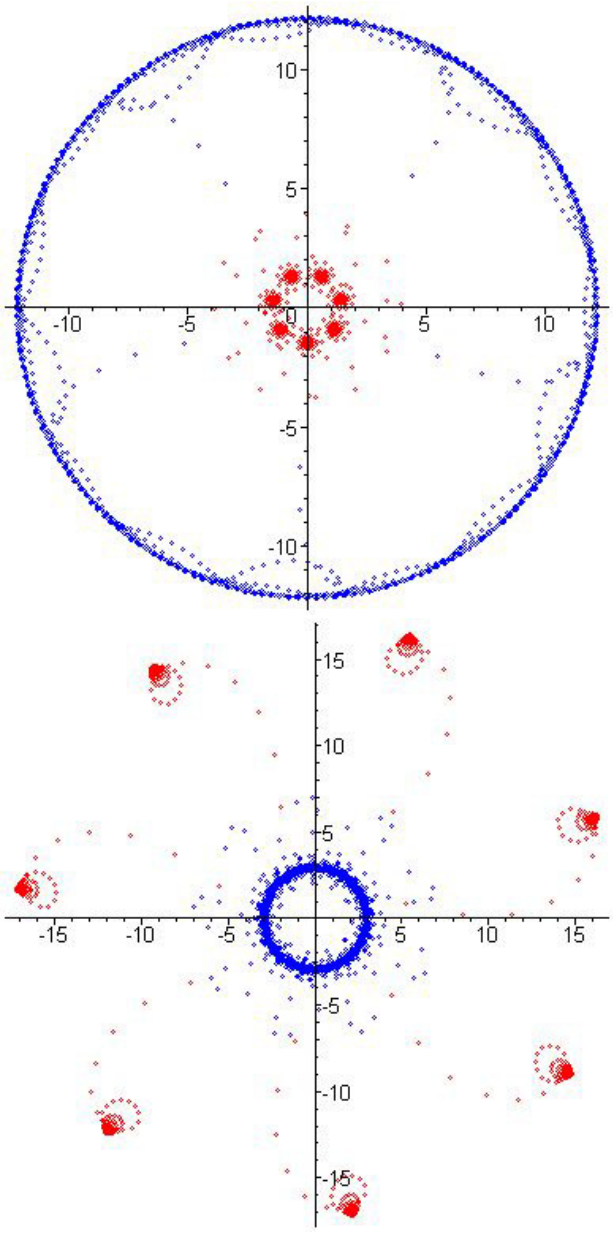


Fig. 1. Dynamics and attractors discovered by plotting $\sqrt{n}t_n$ for various cases with $|a| = |b| = 1$

complex numbers such that $|a| = |b| = 1$, a circle appears when either of a, b is not a root of unity, but k isolated spirals are seen when one of them is a k-th root of unity (i.e., when $a = e^{2i\pi/k}$ for some integer k).

In short, Ramanujan continued fractions and related iterations join the distinguished category of mathematical objects that have been profitably studied via computer graphics, in company with Wolfram's cellular automata, chaotic sequences, and the Julia-set structures of roots of algebraic equations.

5 Formulas for the Riemann Zeta Function

The Riemann zeta function, which is defined by the simple formula

$$\zeta(s) = \sum_{k=0}^{\infty} \frac{1}{n^s} \;=\; 1 + \frac{1}{2^s} + \frac{1}{3^s} + \frac{1}{4^s} + \cdots , \tag{9}$$

is one of the most important objects in modern mathematics, with applications in physics, probability theory, applied statistics and number theory. A premier unsolved problem of mathematics, for which the Clay Mathematics Institute has offered US$1,000,000 for solution, is to rigorously prove the "Riemann hypothesis," namely the assertion that all of the nontrivial solutions of the equation $\zeta(s) = 0$ in the complex plane lie precisely on a particular vertical straight line.

Among the many questions explored by mathematicians over the past century regarding the Riemann zeta function are whether the following intriguing formulas, which have been known for several decades, can be generalized:

$$\zeta(2) = 3 \sum_{k=1}^{\infty} \frac{1}{k^2 \binom{2k}{k}} \;=\; 3 \left(\frac{1}{2} + \frac{1}{4 \cdot 6} + \frac{1}{9 \cdot 20} + \cdots \right) \tag{10}$$

$$\zeta(3) = \frac{5}{2} \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^3 \binom{2k}{k}} \;=\; \frac{5}{2} \left(-\frac{1}{1 \cdot 2} + \frac{1}{8 \cdot 6} - \frac{1}{27 \cdot 20} + \cdots \right) \tag{11}$$

$$\zeta(4) = \frac{36}{17} \sum_{k=1}^{\infty} \frac{1}{k^4 \binom{2k}{k}} \;=\; \frac{36}{17} \left(\frac{1}{1 \cdot 2} + \frac{1}{16 \cdot 6} + \frac{1}{81 \cdot 20} + \cdots \right). \tag{12}$$

Here the notation $\binom{n}{m}$ is shorthand for the binomial coefficient, namely the number of combinations of n objects taken m at a time. Do similar formulas exist for integer arguments greater than four? Numerous mathematicians have tried to find such formulas, but, until recently, none were known.

In 1997, using a combination of integer relation algorithms, the "Pade approximation" facility and some other computational techniques available in Mathematica and similar mathematical software systems, Jonathan Borwein (Peter Borwein's brother) and David Bradley discovered the following unanticipated general formula ([11] or [1, pg. 70–77]):

$$\sum_{k=0}^{\infty} \zeta(4k+3)\, x^{4k} = \frac{5}{2} \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^3 \binom{2k}{k} (1 - x^4/k^4)} \prod_{m=1}^{k-1} \left(\frac{1 + 4x^4/m^4}{1 - x^4/m^4} \right). \tag{13}$$

Here the notation $\prod_{m=1}^{k-1}$ means the product of the term to the right of $\prod$ for m from 1 to $k-1$. Formula (13) permits one to read off an infinity of formulas for

$\zeta(4n+3)$, beginning with formula (11) above when $n = 0$, simply by comparing coefficients of x^{4k} on the left-hand side and the right-hand side of (13).

In 2007, following a similar but much more deliberate computer-experimental procedure, as detailed in [5] or [1, pg. 70–77], a similar general formula was discovered for $\zeta(2n+2)$:

$$\sum_{k=0}^{\infty} \zeta(2k+2)\, x^{2k} = 3 \sum_{k=1}^{\infty} \frac{1}{k^2 \binom{2k}{k} (1 - x^2/k^2)} \prod_{m=1}^{k-1} \left(\frac{1 - 4\,x^2/m^2}{1 - x^2/m^2} \right). \qquad (14)$$

As with (13), one can now read off an infinity of formulas, beginning with formula (10) above when $n = 0$. This general formula was then proved using what is known as the Wilf-Zeilberger algorithm [13]. A comparable general formula for $\zeta(2n+4)$ has also been found, giving formula (12) above when $n = 0$, but a similar general formula for all $\zeta(4n+1)$ is not yet known.

It is worth emphasizing the fact that formula (14) above was both discovered and proven by computer. There is no reason for human mathematicians to panic, as considerable human ingenuity was involved in both steps. But this result is a harbinger of a future in which the computer is as essential to the mathematician, for both discovery and proof, as a particle collider is to a high-energy physicist, or as a DNA sequencer is to a molecular biologist.

6 Proof versus Experiment

Although Wolfram repeatedly champions the experimental approach in *A New Kind of Science*, he also acknowledges that experimental explorations are no substitute for rigorous proof. This principle has been amply underscored during the past few years by discoveries of some remarkable examples that serve as cautionary tales to those who too glibly apply experimental methods.

One particularly sobering example is the following:

$$I = \int_0^{\infty} \cos(2x) \prod_{n=1}^{\infty} \cos(x/n)\, \mathrm{d}x, \qquad (15)$$

where $\int_0^{\infty}$ means the usual signed area under the curve that we study in calculus. Calculating the numerical integral of this oscillating function (see Figure 2) to high accuracy is a nontrivial challenge, but can be done using a scheme described in [6]. When this integral was first computed to 20-digit accuracy, its value appeared to be $\pi/8 = 0.392699\ldots$. But when more than 50 digits were obtained, upon careful comparison with the numerical value of $\pi/8$:

$$I = 0.39269908169872415480783042290993786052464\underline{54341187231595926}\ldots$$
$$\pi/8 = 0.39269908169872415480783042290993786052464\underline{61749218882276 21}\ldots,$$

it is clear that the two values disagree beginning with the 43rd digit! In other words, the integral I is *not* $\pi/8$. At first the authors of this study felt that there

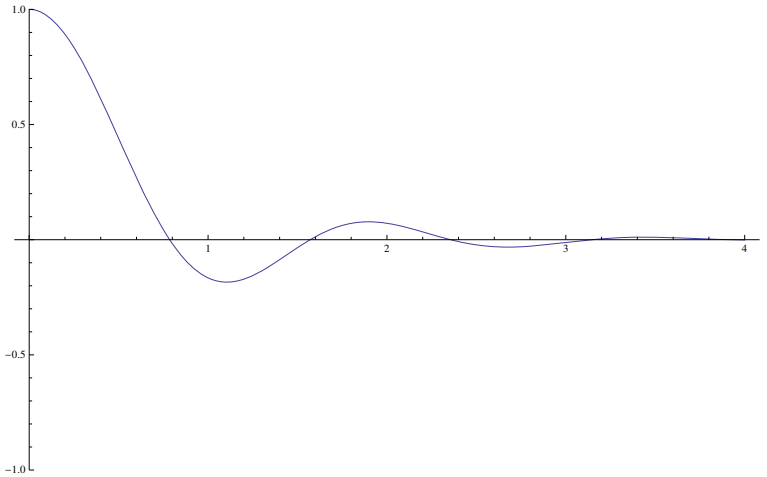


Fig. 2. Graph of the oscillating function $\cos(2x)\prod_{n=1}^{\infty}\cos(x/n)$

must be some "bugs" in the computer programs calculating the integral, but none were found.

Richard Crandall [12] later explained this mystery. In the course of analyzing physically motivated "running out of fuel" random walks, he showed that $\pi/8$ is given by the following very rapidly convergent series expansion, of which the integral (15) above is merely the first term:

$$\frac{\pi}{8} = \sum_{m=0}^{\infty} \int_0^{\infty} \cos[2(2m+1)x] \prod_{n=1}^{\infty} \cos(x/n)\,\mathrm{d}x. \tag{16}$$

As mentioned above, one term of this series is accurate to 43 digits; two terms are accurate to more than 500 digits; three series suffice for at least 8000 digits, and so on.

7 Conclusion

Considerable progress has been made since the publication of AKNS in identifying opportunities and techniques for experimental mathematics. New formulas have been discovered, interesting features have been identified in plots of mathematical structures, and computer-based techniques have been developed to *prove* certain types of results, as well as to discover them in the first place.

However, examples such the one mentioned in the previous section, where mathematical objects differ significantly from what one might think after performing an initial computation, draw attention to the fact that there has not yet been substantial and intellectually rigorous progress in the way experimental mathematics is presented in research papers, textbooks and classroom instruction, or in how the mathematical discovery process is organized. This is an

arena where works such as *A New Kind of Science* can have significant impact. The present author, for one, looks forward to this dialogue. See [3] for more discussion.

References

[1] Bailey, D., Borwein, J., Calkin, N., Girgensohn, R., Luke, R., Moll, V.: Experimental Mathematics in Action. A K Peters, Natick (2007)

[2] Bailey, D.H., Borwein, J.M.: Experimental mathematics: Examples, methods and implications. Notices of the American Mathematical Society 52, 502–514 (2005)

[3] Bailey, D.H., Borwein, J.M.: Exploratory experimentation and computation. Notices of the American Mathematical Society 58, 1410–1419 (2011)

[4] Bailey, D.H., Borwein, J.M., Borwein, P.B., Plouffe, S.: The quest for Pi. Mathematical Intelligencer 19(1), 50–57 (1997)

[5] Bailey, D.H., Borwein, J.M., Bradley, D.M.: Experimental determination of Apery-like identities for $\zeta(2n+2)$. Experimental Mathematics 15, 281–289 (2006)

[6] Bailey, D.H., Borwein, J.M., Kapoor, V., Weisstein, E.: Ten Problems in Experimental Mathematics. American Mathematical Monthly 113(6), 481–409 (2006)

[7] Bailey, D.H., Borwein, J.M., Mattingly, A., Wightwick, G.: The computation of previously inaccessible digits of π^2 and Catalan's constant. Notices of the American Mathematical Society (April 11, 2011) (to appear), preprint available at `http://crd-legacy.lbl.gov/~dhbailey/dhbpapers/bbp-bluegene.pdf`

[8] Berndt, B.C.: Ramanujan's Lost Notebook, vol. 1-5. Springer (1985) (date of vol. 1)

[9] Borwein, J.M.: The Life of Pi (2011) (manuscript), `http://www.carma.newcastle.edu.au/~jb616/pi-2011.pdf`

[10] Borwein, J.M., Bailey, D.H.: Mathematics by Experiment, 2nd edn. AK Peters, Natick (2008)

[11] Borwein, J.M., Bradley, D.M.: Empirically determined Apery-like formulae for $\zeta(4n+3)$. Experimental Mathematics 6, 181–194 (1997)

[12] Crandall, R.E.: Theory of ROOF Walks (2007), `http://www.perfscipress.com/papers/ROOF11_psipress.pdf`

[13] Wilf, H.S., Zeilberger, D.: Rational functions certify combinatorial identities. Journal of the American Mathematical Society 3, 147–158 (1990)

[14] Wolfram, S.: A New Kind of Science. Wolfram Media, Champaign (2002)

Part III

Mechanisms in Biology, Social Systems and Technology

Chapter 7
More Complex Complexity: Exploring the Nature of Computational Irreducibility across Physical, Biological, and Human Social Systems

Brian Beckage[1], Stuart Kauffman[2], Louis J. Gross[3], Asim Zia[4], and Christopher Koliba[4]

[1] Department of Plant Biology, University of Vermont, Burlington, VT, USA
[2] Department of Mathematics, Department of Biochemistry, and Complexity Group, University of Vermont, Burlington, VT, USA
[3] National Institute for Mathematical and Biological Synthesis University of Tennessee, Knoxville, TN, USA
[4] Department of Community Development and Applied Economics, University of Vermont, Burlington, VT, USA

Abstract. The predictability of many complex systems is limited by computational irreducibility, but we argue that the nature of computational irreducibility varies across physical, biological and human social systems. We suggest that the computational irreducibility of biological and social systems is distinguished from physical systems by functional contingency, biological evolution, and individual variation. In physical systems, computationally irreducibility is driven by the interactions, sometimes nonlinear, of many different system components (e.g., particles, atoms, planets). Biological systems can also be computationally irreducible because of nonlinear interactions of a large number of system components (e.g., gene networks, cells, individuals). Biological systems additionally create the probability space into which the system moves: Biological evolution creates new biological attributes, stores this accumulated information in an organism's genetic code, allows for individual genetic and phenotypic variation among interacting agents, and selects for the functionality of these biological attributes in a contextually dependent manner. Human social systems are biological systems that include these same processes, but whose computational irreducibility arises as well from sentience, i.e., the conscious perception of the adjacent possible, that drives social evolution of culture, governance, and technology. Human social systems create their own adjacent possible through the creativity of sentience, and accumulate and store this information culturally, as reflected in the emergence and evolution of, for example, technology. The changing nature of computational irreducibility results in a loss of predictability as one moves from physical to biological to human social systems, but also creates a rich and enchanting range of dynamics.

Keywords: computational irreducibility, complexity, evolution.

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 79–88.
DOI: 10.1007/978-3-642-35482-3_7

1 Introduction

Systems change through time–whether the system of interest is a galaxy, a forest ecosystem, a social network, or a circulatory system. This continuous process of change in the system state can be thought of as computation [35]: the state of the system is updated based on its current and past states. In a forest, for example, the growth and recruitment of trees is dependent on the spatial arrangement of trees through processes such as competitive interactions, light shading, and seed dispersal as well as environmental externalities [4]. In some systems, precise predictions of the future state of the system can be made without having to perform the intervening computations. In these systems, prediction is possible because simplified models exist that can be used to bypass the intervening computations intrinsically performed by the system . Astronomical models, for example, can predict the spatial and temporal distribution of sunlight on earth, and describe the past orbital forcing of the climate system [20]. In other systems, like a forest ecosystem, predicting the detailed state of the system is very difficult without allowing the system to update itself on its own characteristic time scale [3]. Systems that require the computation of intervening system states on their characteristic time scale in order to predict future states are computationally irreducible. Computational irreducibility therefore implies the absence of simplifying models that can reproduce future system states without information loss. The dynamics of a system that is computationally irreducible cannot be known without allowing for the evolution of the system on its own time scale. While any process that is computationally irreducible may seem to imply an equivalent degree of unpredictability a priori, we suggest that this is not the case. We argue that the processes that drive computational irreducibility differ across physical, biological and social systems, and that these differences result in some forms of computational irreducibility being 'more irreducible' than others. Computational irreducibility does not imply that predictions are impossible, but that they come at the cost of information loss. In cellular automata, for example, cells can be spatially aggregated into larger units with an associated set of updating rules in a process of coarse-graining [18] [19]. Prediction in some computationally irreducible systems is possible through coarse-graining, but comes at the cost of information loss through spatial and temporal averaging. We suggest, then, that gains in prediction require increasing information loss in physical, biological, and human social systems, and thus some systems are more computationally irreducible than others. We argue that the basis for these differences lie in the different processes operating in physical, biological, and human social systems.

2 Physical Systems

Physics has been particularly successful at prediction. Physicists, for example, were able to predict the existence of black holes from a singularity in the equations describing the physical system [31] [32]. Engineers routinely use the laws of physics to design and build skyscrapers, bridges, airplanes, and to send spacecraft to distant planets, and these efforts are usually successful. We don't mean

to imply that physics is axiomatic and its laws universal but rather that, while mathematical representations of physical laws may be only approximate descriptions of underlying physical reality, the approximations can be quite good. The approximate laws of physics seem to be much more useful for prediction than the approximate laws of biological or human social systems.

We argue that physical systems tend to be more predictable than living systems because computational irreducibility in these systems is driven by a smaller set of less complex processes. Computational irreducibility in physical systems largely results from the interactions of particles or objects governed by a fixed set of rules, analogous to simple cellular automata [35]. Physical systems can become computationally irreducible with a relatively small number of interacting objects, e.g., the three body problem [35], and systems with large numbers of interacting components are likely to be computationally irreducible. The evolution of a large volume of a gas, for example, may be computational irreducible even as the gas molecules interact with each other and their surrounding environment according to known physical laws. An approximate, statistical description of the mean state of a gas is still possible, however, without an exact description of the velocities and locations of each molecule: the temperature and pressure of a gas can be described using the ideal gas law. Physical systems that are computationally irreducible can often become predictable from a macro-level perspective due to the averaging of a very large number of separate interactions, albeit with the loss of information. This is analogous to the coarse-graining of cellular automata described earlier.

The computational irreducibility of physical systems is related to the Halting Problem in a Universal Turing Machine [34]. A computation is said to be incompressible when the sequential behavior of a computer program cannot be computed in any shorter way than to allow the program to run and note its successive states. The central features of a Turing machine include a head with a set of pre-stated symbols standing for internal states, a tape marked in squares of perhaps infinite length, a prestated alphabet of symbols (e.g., 0 and 1) that can be read from and written to the tape by the reading head [10]). Given a set of symbols on the tape, and the reading head over one square with a symbol written on it, the head reads from the tape and, depending upon the symbol, its internal state will not move or move one square to the left or right, erase the symbol on the square below it, write a symbol on that square and go from one internal state to another internal state. Then this set of operations will iterate. Given any initial state of the tape and reading head, the next states of the tape and head can be computed for $1, 2, 3, \ldots, N$ finite number of steps ahead. A Turing machine is a subset of classical physics.

We define the computationally irreducibility of physical systems and other Turing-like systems as first order computationally irreducible. This is the simplest mode of computational irreducibility in that the set of rules governing system evolution and the possible states of the particle or node are fixed, e.g., the set of potential states of a cell in a simple automaton, and do not change as the system itself evolves. We suggest that coarse-graining approaches to prediction

would be most effective in systems with first order computational irreducibility, i.e., they would gain greatest predictive capacity with minimal information loss. We argue that biological systems and human social systems have a different set of processes governing system evolution than those found in physical systems and associated with first order computational irreducibility.

3 Biological Systems

Biological systems are computationally irreducible for qualitatively different reasons than physical systems. While the same processes that yield first order computational irreducibility in physical systems also operate in biological systems, i.e., large number of interacting components, the set of rules governing these interactions and the potential states of the system components (e.g., cells in a CA, particles, organisms) evolve along with the overall state of the system. We refer to this as second order computational irreducibility–a more complex computational irreducibility than the first order computational irreducibility. The second order nature of the computational irreducibility of biological systems–meaning that the rules and set of states of fundamental units can evolve–follows from nearly universal attributes of biological systems: i) contingency of the function and selective value of biological attributes on interactions with other organisms and their environment, ii) the creation of new attributes and functions through biological evolution, and iii) individual variability in biological attributes even among organisms of the same species. Note that we use the term 'biological evolution' to refer to Darwinian evolution in biological systems as distinguished by 'system evolution', which describes changes in the state of a system through time, although biological evolution often leads to system evolution of biological systems.

Functional contingency. Biological attributes have a set of potential functions, and the set of these functions is contextually dependent on interactions with other organisms and the environment. A subset of the functions associated with an attribute may be useful in the current context of an organism and its interactions with other organisms and its environment, while other function of an attribute may be useful in other future (or past) contexts. Feathers in dinosaurs, for example, may have initially functioned in thermal regulation and only later provided additional functionalities that were coopted for flight [6] [33]. The swim bladder, a sac found in some fish that is partly filled with air and partly with water and the ratio of which determines and adjusts neutral buoyancy in the water column, is believed to have arisen from the lungs of lung fish, providing a new functionality to an existing structure [29]. Even the human capacity for reason and logic may have been a new functionality of biological traits with origins in the context of group dynamics of social organisms [26]. Some components of the set of functions of existing biological attributes might have causal consequences that are of selective significance in new environments. Functions of biological structures that are of no selective advantage in the current environment but that become selectively advantageous in later environments, typically with a

new functionality, are referred to as pre-adaptations or exaptations. We assert that the potential functions of biological attributes are both indefinite in number and unorderable, and, importantly, that no algorithm can list them all. We argue that this means that the set of rules governing system evolution changes and contributes to the second order computational irreducibility of biological systems.

Biological evolution. Biological evolution is a central process that distinguishes the evolution of the biosphere from other physical systems [24]. Biological systems create and accrue attributes such as new structures, biochemical pathways, gene regulatory networks, etc. through biological evolution. These attributes provide the basis for biological function and exaptations of the previous section. The process of biological evolution is immensely creative and unpredictable, and forms a positive feedback loop that leads to further biological evolution. The evolution of feathers in dinosaurs and their ultimate use in flight resulted in the emergence of a completely new set of ecological niches, and an associated proliferation of species of birds. The emergence of flight in birds, in turn, has allowed for the long range transportation of seeds and organisms to islands and inland water bodies (e.g., [25]), opening even new ecological niches, providing the basis for new functionality of existing biological structure, and for continued evolutionarily development of biological attributes. Seabirds are, for example, responsible for substantial nutrient flows from oceans to terrestrial ecosystems, and their presence or absence can determine whether a landscape is in one of two alternative stable states–a grassland or closed shrubland [9].

Individual variation. Biological systems are distinguished from purely physical systems by individual variation of agents. Individual organisms often differ from other individuals of the same species [7]. Much of this variation is derived from underlying genetic differences and these genetic differences provide the basis for differences in biological attributes, e.g., behaviors, functions, and environmental responses and the raw material for biological evolution. Individual variation within species has been postulated to be a key mechanism driving patterns of and maintaining species diversity in ecological communities [4]. Species phenology, for example, describes the seasonal timing of demographic processes such as flowering in trees (e.g., [27]). Individual variation in response to environmental cues (e.g., day length, temperature) means that some trees will bud out and flower earlier in the spring than others. An earlier phenology could increase the likelihood of seeds colonizing and capturing new available (empty) sites in a forest or, alternatively, increase the risk of being adversely impacted by a late spring frost. The consequences of individual variation, thus, depend on the environmental and ecological context. Individual variability means that the rules for updating a system can vary from individual to individual even if the environmental context is identical. In a cellular automaton, this is analogous to cell to cell variation in the updating rule for a specific cell type, for instance, among different white cells even with identical neighborhoods.

Synthesis. Biological evolution creates attributes of organisms and the biological system creates the context that determines the functionality and utility of these attributes. Biological evolution led to photosynthesis, and photosynthesis then resulted in abundant free oxygen in the atmosphere (e.g., [30] [21]. Biological attributes that enabled aerobic respiration in the presence of free oxygen were advantageous in this new context. Free oxygen and aerobic respiration, subsequently allowed for a wide array of niches that did not exist before and these niches could be occupied by species with new or pre-adapted functional attributes (e.g., [28]). Biological systems create and modify their own adjacent possible through construction of or extension of biological function or niche space that is immediately adjacent to current niche space. The creation of new biological opportunities allows for the emergence of new organisms, new functionalities, and a new adjacent possible. This process is enormously creative and unpredictable a priori. Biological systems are thus second order computationally irreducible, because the rules for updating and the potential states of the system change as the system evolves. The evolution of the biosphere is non-algorithmic. We claim that no algorithm can pre-state all possible biological attributes, their potential functions, or how these functions might be of selective advantage in potential future environments. The unpredictability of biological systems is thus radically unlike the computational incompressibility of physical systems, the Halting problem on a universal Turing machine or, a fortiori, unlike the irreducibility of cellular automata.

4 Human Social Systems

Human social systems are a specialized case of a biological system with an additional source of computational irreducibility: sentience. We use 'sentience' to refer to the state of being conscious, self-aware, and having free will. Humans are sentient beings that are able to perceive their own possibilities within the context of their environment. A person might, for instance, conceive of a network of linked computers that would later become the internet and allow for the world wide web. The creation of the internet and world wide web then provides the basis for other innovations that are dependent on the existence of the internet, e.g. social networking websites, cloud computing, etc. The creation of the internet allowed for the possibility of these subsequent innovations–the internet resulted in a new and expanded adjacent possible. All of these innovations–the internet, social networking websites, and cloud computing–require a person(s) that imagined or perceived the possibility of these innovations in a given context. Similar sequences of creative expansion of the adjacent possible can be found in many contexts outside of technology–from music and visual art to the development of law and systems of governance. Sentience thus acts to create what is possible adjacent to what currently exists in a manner analogous to biological evolution, and this process proceeds in a positive, self-reinforcing feedback loop: Innovation creates the opportunity for more innovation.

Sentience and the perception of possibility distinguish the computational irreducibility of human social systems from physical and other biological systems.

The processes that contribute to the computational irreducibility of physical and biological systems also apply to human social systems, i.e., interactions among many system components, biological evolution, and individual variation, functional contingency. Sentience operates in addition to these processes and sets the computational irreducibility of human social systems apart from these other systems. We thus characterize human social systems as having third order computational irreducibility. Third order computational irreducibility is distinguished by the sentient perception of what is possible in a given context, and drives the evolution of technology, economics, governance, and other components of the human social system. We expect that human social systems will be less predictable than biological or physical systems, meaning that predictive gains from coarse-graining will result in larger information loss than occurs in these other systems.

In the context of human social systems, the adjacent possible is related to the concept of affordances. Affordances are the attributes of an object or environment that allow an action to be performed [12]. Affordances are action possibilities that humans perceive as, for example, the many potential uses of a screw driver (e.g., turning a screw, opening a can, puncturing a tire). Affordances are in many ways analogous to the process of biological evolution 'discovering' the function of attributes of organisms in the context of an organism's environment. The relationships between humans and their environment can thus lead to perceived possibilities, actions, and cognition, and is dynamic, reciprocal, and contextual [23]. While our discussion has focused on individuals and consciousness, human social systems operate across hierarchical levels of structure. Social systems include individuals, small groups of people, more expansive social organizations and institutions, and networks of organizations [22]. Each of these levels of social organization contributes to the computational irreducibility of social systems, but the sentience of individuals–and the inherit variability among individuals–is the defining process that distinguishes human social systems from other purely biological systems. The computational irreducibility that stems from sentience is compounded by the interactions between and among the other components of social systems. The agency of an individual person can affect higher levels of social organization (e.g., through leadership and contagion of beliefs), but social groups and organizations also impact the actions and identities of individuals. These feedbacks and linkages between individuals and groups have likely been made stronger and more fluid with the advent of social media, and are central to understanding and predicting trajectories of human social systems. Lastly, the role of culture in accumulating and transferring information among individuals is a central feature of human social systems that is akin to information storage in the genetic code in biological systems. *Challenges.* Designing algorithms for essentially non-algorithmic problems has been problematic since the onslaught of Turing-complete machines (e.g., [8]). In human social systems, the problem of framing affordances has not yet been programmed. Whether it is programmable or not is a question that is central to the field of computational complexity, artificial intelligence and robotics. Agent based models (ABMs) have opened up

new vistas of scientific discovery to simulate decision-making by heterogeneous agents in artificial societies (e.g. [13]), but there are significant limits to the algorithmic approach for simulating both creative decision making by intelligent agents in rapidly shifting environments and social dynamics, a problem that was even acknowledged by Turing [10]. Fundamental assumptions that are ingrained in each algorithm about the behavioral rules, creative decision-making, learning, treatment of uncertainty and so forth, constrain the modeling of emergence, self-organization and adaptation in complex social systems. Different types of algorithms and Turing-complete machines such as agent based models, genetic algorithms and artificial neural networks, have opened up new vistas for modeling creative decision making in finite, discrete, computational steps (e.g., [17] [16] [14]). Human social systems with heterogeneous agents with the capability for creative decision making in rapidly shifting social environments may significantly limit the potential for algorithms to model and predict the trajectory of these systems. Our understanding of emergence, self-organization and adaptation in complex systems populated by sentient agents that undertake creative decision-making is limited by algorithms.

5 Conclusions

The limits to predictive capacity imposed by computational irreducibility is increasingly important as we confront complex and interlinked problems that incorporate natural and human social systems. Predicting the trajectory of earth's climate system, for example, is an important but difficult problem because it incorporates human social, biological, and physical systems. Computational irreducibility is an inevitable feature of complex systems, but we argue that not all forms of computational irreducibility are equivalent. The underlying processes that lead to computational irreducibility and the potential for gains in predictive capacity vary across physical, biological, and social systems. Physical systems have the simplest kind of computational irreducibility, which we define as first order computational irreducibility, in which neither the set of potential states nor the rules for updating the states change as the system evolves. The potential for system prediction is likely to be the greatest with first order computational irreducibility but with the loss of information. Biological systems have a more intransigent computational irreducibility because the potential system states and updating rules change as the system evolves. Functional contingency, biological evolution, and individual variation are three underlying processes that lead to this second order computational irreducibility of biological systems. Humans perceive and create their own adjacent possible and this sentience leads to human social systems being characterized by third order computational irreducibility. The increasingly difficult forms of computational irreducibility across physical, biological, and human social systems, and the low predictive capacity found in these living systems is offset by their remarkably rich, diverse, and creative dynamics. Although we argue that ultimately, the evolution of the biosphere is non-algorithmic, there is much to be learned in the pursuit of the frontier of

first, second and third order computational irreducibility, and this will challenge computational modelers to reach the outer limits of computational irreducibility.

Acknowledgements. Brian Beckage gratefully acknowledges the support of the National Science Foundation (Award 0950347). This work was supported by the National Institute for Mathematical and Biological Synthesis, an Institute sponsored by the National Science Foundation, the U.S. Department of Homeland Security, and the U.S. Department of Agriculture through NSF Award EF-0832858, with additional support from The University of Tennessee, Knoxville.

References

[1] Arora, S., Barak, B.: Computational Complexity: A Modern Approach (2009) ISBN 0521424267
[2] Bateson, G.: Steps to an ecology of mind (1972) ISBN 0876689500
[3] Beckage, B., Gross, L., Kauffman, S.: The limits to prediction in ecological systems. Ecosphere 2(11), 125 (2011), doi:10.1890/ES11-00211.1
[4] Beckage, B., Gross, L., Platt, W., Godsoe, W., Simberloff, D.: Individual variation and weak neutrality as determinants of species diversity. Frontiers of Biogeography 3(4), 145–155 (2012)
[5] Berlinski, D.: The advent of the algorithm: the idea that rules the world (2000) ISBN 0756761662
[6] Bock, W.J.: Explanatory history of the origin of feathers. American Zoologist 40, 478–485 (2000)
[7] Clark, J.S.: Individuals and the variation needed for high species diversity in forest frees. Science 327, 1129–1132 (2010)
[8] Cooper, B.: The incomputable reality. Nature 482, 465–465 (2012)
[9] Croll, D.A., Maron, J.L., Estes, J.A., Danner, E.M., Byrd, G.V.: Introduced Predators Transform Subarctic Islands from Grassland to Tundra. Science 307, 1959–1961 (2005)
[10] Dyson, G.: The dawn of computing. Nature 482, 459–460 (2012)
[11] French, R.M.: Dusting Off the Turing Test. Science 336, 164–165 (2012)
[12] Gibson, J.J.: The Theory of Affordances. In: Shaw, R., Bransford, J. (eds.) Perceiving, Acting, and Knowing (1977) ISBN 0-470-99014-7
[13] Gilbert, N., Conte, R.: Artificial Societies: The Computer Simulation of Social Life. UCL Press, London (1995)
[14] Grimm, V., Railsback, S.F.: Individual-based modeling and ecology. Princeton University Press (2005)
[15] Hodges, A.: The man behind the machine. Nature 482, 441–441 (2012)
[16] Holland, J.H.: Genetic algorithms. Scientific American, 44–50 (1992) ISSN 0036-8733
[17] Holland, J.H.: Adaptation in Natural and Artificial Systems (1975) ISBN 0472084607
[18] Israeli, N., Goldenfeld, N.: Computational Irreducibility and the Predictability of Complex Physical Systems. Physical Review Letters 92, 074105 (2004)
[19] Israeli, N., Goldenfeld, N.: Coarse-graining of cellular automata, emergence, and the predictability of complex systems. Physical Review E 73, 026203 (2006)

[20] Karl, T.R., Trenberth, K.E.: What is climate change? In: Lovejoy, T.E., Hannah, L.J. (eds.) Climate Change and Biodiversity. Yale University Press, New Haven (2005)
[21] Kasting, J.F.: Theoretical constraints on oxygen and carbon dioxide concentrations in the Precambrian atmosphere. Precambrian Research 34, 205–229 (1987)
[22] Koliba, C., Meek, J., Zia, A.: Governance networks in public administration and public policy. CRC Press/Taylor & Francis, Boca Raton, FL (2010)
[23] Letiche, H., Lissack, M.: Making Room for Affordances. Emergence: Complexity and Organization (E:CO) 11(3), 61–72 (2009) ISSN 1532-7000
[24] Longo, G., Montvil, M., Kauffman, S.: No entailing laws, but enablement in the evolution of the biosphere. arXiv:1201.2069v1 [q-bio.OT] (2012)
[25] McAtee, W.L.: Distribution of Seeds by Birds. American Midland Naturalist 38, 214–223 (1947)
[26] Mercier, H., Sperber, D.: Why do humans reason? Arguments for an argumentative theory. Behavioral and Brain Sciences 34(2), 57–74 (2011)
[27] Morin, X., Lechowicz, M.J., Auspurger, C., O'Keefe, J., Viner, D., Chuine, I.: Leaf phenology in 22 North American tree species during the 21st century. Global Change Biology 15, 961–975 (2009)
[28] Nisbet, E.G., Sleep, N.H.: The habitat and nature of early life. Nature 409, 1083–1091 (2001)
[29] Perry, S.F., Wilson, R.J., Straus, C., Harris, M.B., Remmers, J.E.: Which came first, the lung or the breath? Comp. Biochem. Physiol. A Mol. Integr. Physiol. 129(1), 37–47 (2001)
[30] Rutten, M.G.: The history of atmospheric oxygen. Origins of Life and Evolution of Biospheres 2, 5–17 (1970)
[31] Schwarzschild, K.: Uber das gravitationsfeld eines massenpunktes nach der Einsteinschen theorie. Sitzungsberichte der Deutschen Akademie der Wissenschaften zu Berlin, Klasse fur Mathematik, Physik, und Technik, p. 189 (1916a)
[32] Schwarzschild, K.: Uber das Gravitationsfeld einer Kugel aus inkompressibler Flussigkeit nach der Einsteinschen Theorie. Sitzungsberichte der Deutschen Akademie der Wissenschaften zu Berlin, Klasse fur Mathematik, Physik, und Technik, p. 424 (1916b)
[33] Sumida, S.S., Brochu, C.A.: Phylogenetic Context for the Origin of Feathers. American Zoologist 40, 486–503 (2000)
[34] Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. Proc. Lond. Math. Soc. 42(2), 230–265 (1936); correction ibid 43, 544–546 (1937)
[35] Wolfram, S.: A New Kind of Science. Wolfram Media, Urbana

Chapter 8
A New Kind of Finance

Philip Z. Maymin

NYU-Polytechnic Institute, USA
philip@maymin.com

Abstract. Finance has benefited from the Wolfram's *NKS* approach but it can and will benefit even more in the future, and the gains from the influence may actually be concentrated among practitioners who unintentionally employ those principles as a group.

Keywords: algorithmic finance, computable economics, cellular automata, iterated finite automaton, agent-based modeling.

The insights and techniques from Stephen Wolfram's *A New Kind of Science* [26]—namely that simple systems can generate complexity, that all complexity is maximal complexity, and that the only general way of determining the full effects of even simple systems is to simulate them–are perhaps most useful, and least applied, in the field of finance.

The influence of *NKS* on the current state of finance depends on the particular area of finance being studied. In the area of market-based finance, a unique minimal model of financial complexity has been discovered. In the area of government-based finance, the same minimal model has been used to test the effects of different regulatory regimes. Those are academic results; finance is ultimately a practitioner's field. From the perspective of practitioners, a result linking computational efficiency and market efficiency has been found.

In short, finance has benefited from the *NKS* approach but it can and will benefit even more in the future, and the gains from the influence may actually be concentrated among practitioners who unintentionally employ those principles as a group.

What is finance, anyway? It can be hard enough to pronounce, let alone define. Should it be *FIE-nance* , or *fih-NANCE*? I've studied, researched, and practiced in the field for most of my life, and I still don't know how to pronounce it. Fortunately, I'm not alone. Dictionaries list both as acceptable pronunciations.

Perhaps it depends on whether the word is used as a verb or a noun. After some prodding, English American speakers will usually agree that the former is the proper pronunciation for the verb form, as in when you *FIE-nance* a car, and the latter for the noun form, as in when you protest the bailouts of companies involved in high *fih-NANCE*. The British feel just as strongly that one form is a verb and the other a noun—but the opposite ones.

The term originated from the French *fin*, marking the end of a contract through the fulfillment of an obligation or debt. As such, finance has a noble

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 89–99.
DOI: 10.1007/978-3-642-35482-3_8

libertarian heritage. But it shares the same root as the unfortunately authoritative English word "fine", meaning a penalty payment to a government.

This pronunciation ambiguity is not just a curiosity. It is an omen and a symptom of the deep divide in the study of finance between market-based approaches and government-based approaches. And it turns out that this deep divide explains why in finance the *NKS* perspective is both so sorely needed and so often neglected.

1 Market-Based Approaches

Markets resulting from voluntary trade tend to be complex phenomena. A typical price chart shows wild swings, big jumps, bubbles, and crashes. These are even more obvious when we look at the chart of returns instead of prices. (Recall that the return from one day to the next is the percentage you would have earned if you bought it one day and sold it on the next.)

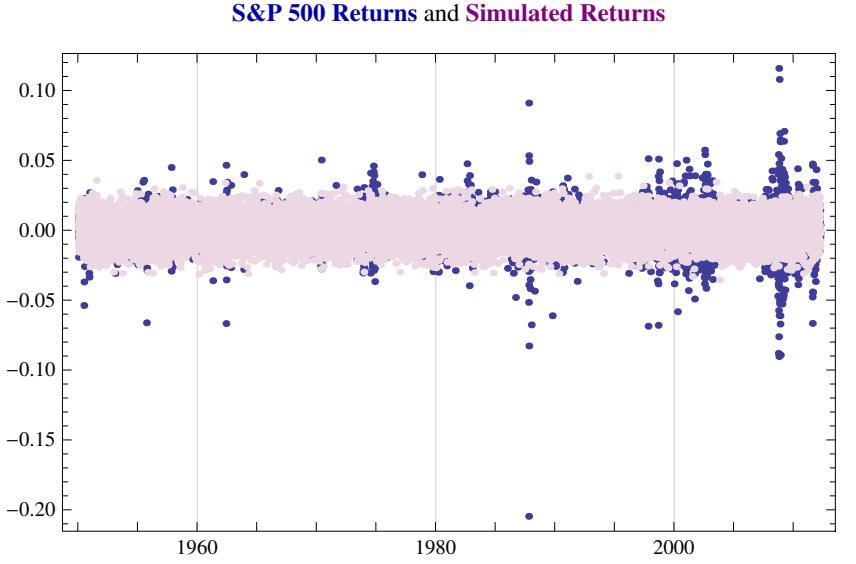


Fig. 1. The dark dots are the actual daily returns of the Standard & Poor's 500, the most widely followed broad based U.S. market index. The light dots overlaid on top are simulated returns from the Normal distribution having the same mean and standard deviation as the actual returns. You can see that the blue dots vary wildly, much more than could be expected from a Gaussian distribution. In addition, these periods of higher volatility tend to cluster together. And finally, there is Black Monday, October 19, 1987, when the market fell by 20 percent.

As Wolfram has noted, most academic market-based approaches to explaining or understanding these complexities essentially ignore the vast amount of seeming randomness and focus on the few pockets of predictability. For example, momentum, the idea that winners will keep winning and losers will keep losing,

seems to be a persistent feature of many markets, and has been the subject of thousands of scholarly papers after its first documentation by Jegadeesh and Titman [3]. But the effect of momentum, while profitable, is still rather small compared to the vast degree of randomness.

Virtually the only tools used for this standard strand of research are regression analysis, attempting to explain individual security or portfolio returns through a fixed number of factors, and portfolio construction, attempting to sort portfolios into buckets based on some factors or indicators and explore the difference in future performance between the highest and the lowest buckets.

In *NKS*, Wolfram explored the alternative approach of trying to model the randomness directly rather than ignoring it. He proposed a one-dimensional cellular automaton model where each cell represents an agent's decision to buy or sell, and the running totals of black cells can be used to infer a market price. Jason Cawley has generalized this model in a *Mathematica* demonstration[1].

In a sense, cellular automata models for financial prices are a subset of the more general recent approach of agent-based modelling. Here, agent behavior is programmed into several varieties, initial proportions of each are chosen, and the interactions between those agents generates market transactions and prices. Gilbert [2] offers a comprehensive introduction and treatment of this literature. The Santa Fe Institute created an artificial stock market a few decades ago; Ehrentreich [1] focuses on agent-based finance and specifically on the lessons of this market. The ability to create multi-agent models has become even easier with the introduction of specialized environments for such tasks such as NetLogo [9].

However, all such agent-based models, including Wolfram's, rely on multiple agents interacting and trading with each other, often with multiple securities too. In the spirit of *NKS*, we should ask: could a single representative investor trading a single security generate complexity?

This was exactly the question I asked during the NKS Summer School of 2007. I realized that because there was no one for the lonely representative investor to trade with, and no other assets for him to compare his to, he would have to be a technical trader, someone who makes decisions based solely on the past history of prices. Technical traders are also called chartists because they often rely on graphical representations of past prices, such as when moving averages of different lookback windows cross, or when the prices seem to form a recognizable visual pattern. Indeed, given the recognition in *NKS* that our natural visual ability was well-adapted to discerning complexity, it seemed reasonable to assume that some of the skills of a technical trader could possibly result in complexity in the price series directly.

Although technical traders can rely on any function of historical prices, a simpler and yet still fully general approach would be to model a trader as evaluating an arbitrary algorithm taking as input the previous prices, or price changes, or even just the signs of those price changes, starting with the most recent first.

The primary benefit of the NKS Summer School is working one-on-one with the author. Indeed, Wolfram suggested using an iterated finite automaton (c.f.

[1] `http://demonstrations.wolfram.com/TimeSeriesFromCellularAutomata/`

Wolfram [11]) to model the internal algorithm of the trader. An iterated finite automaton (IFA) takes one list of symbols and outputs another, and can have internal states. It is thus a collection of rules of the form:

{state1,input} → {state2,output}

Trivially, no single-state IFA generates complexity. Among all of the 256 possible two-state IFAs, there turned out to essentially be only one unique trading rule that generated some form of complexity. Using Wolfram's IFA numbering scheme, this was trading rule 54, depicted by the graphical network below.

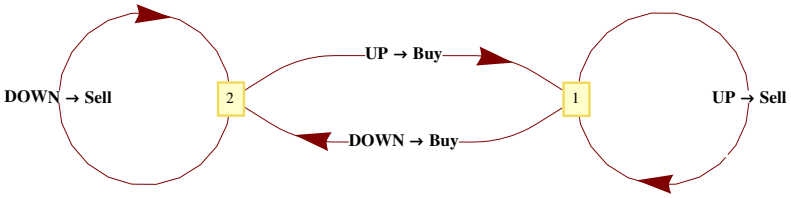


Fig. 2. The boxes represent the two internal states of the trader's rule. He starts every day in state 1. He looks back at the n most recent price changes, starting with the most recent, and follows the arrows until he reaches the nth-most recent one.

Suppose his lookback window is $n = 3$ days. If today is Thursday, then he would have to look back at the market price change on Wednesday, Tuesday, and Monday, in that order. Let's say the market was down Wednesday. So he leaves state 1 following the DOWN arrow, which leads him to state 2. That DOWN arrow also outputs a Buy signal. This can be viewed as his current thinking on what to do in the market, but it is not his final decision because he has not looked at all of the past few days that he intended to.

Next he would need to look at Tuesday's price change. Suppose it too was down. Then he would follow the DOWN arrow out of his current state, state 2. This arrow leads him back to state 2, and updates his current thinking to Sell.

Ultimately, his decision on whether to buy or sell will now depend on what the price change on Monday was: if the market had been down, he would now sell, and if it had been up, he would now buy. Whatever he does, the market follows, because he is the representative investor. So if he were to Buy, no transactions would actually take place, because he has no one to trade with, but the level of the market would go up so he is now indifferent about buying more. The next day, he starts the process all over again, starting with the most recent price change, which happened to be up.

This rule 54 generates quite complex behavior, for virtually any lookback window. The graphs below show the price processes for a variety of lookback windows (See Fig. 3).

So it is true that the absolute simplest model of trading can indeed generate complex price patterns, validating the key insights of *NKS* and finally answering

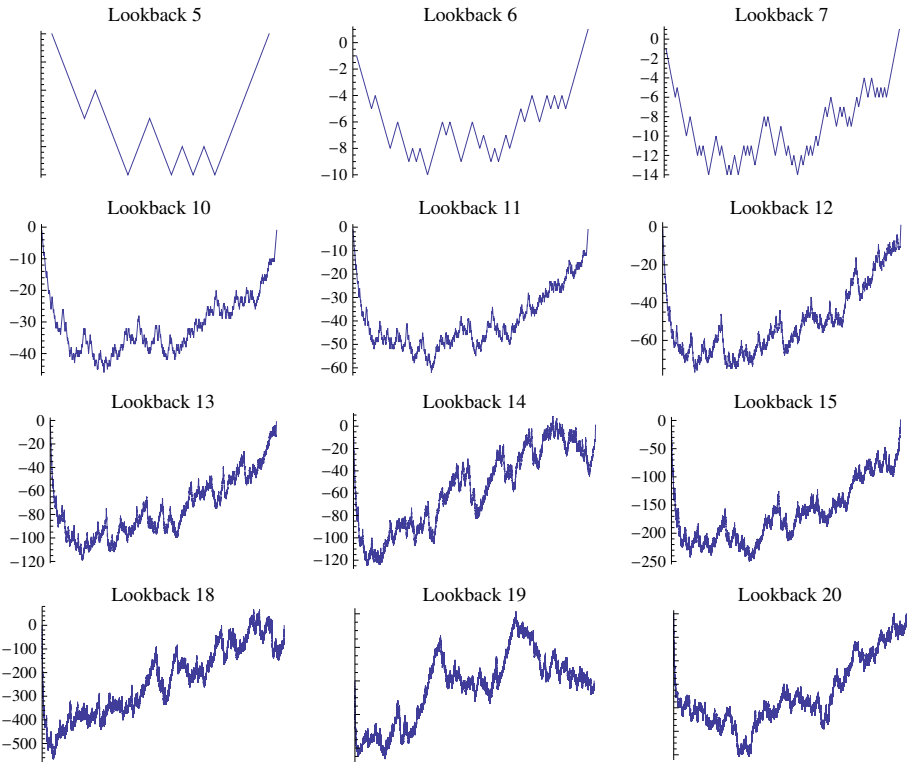


Fig. 3. Notice how the prices jump down drastically before coming back up. Because the prices are always deterministically calculated, they will eventually cycle, and could in principle start anywhere along the cycle. Thus, the big jump down could occur later in the cycle. Furthermore, the lookback window can be made larger so that the cycle time is longer than the age of the universe: looking back just 22 ticks means the cycle time is more than four million ticks.

a question that Wolfram had worked on for decades. With just a single trader and a single asset, and only two internal states, there is essentially a unique rule that generates complex security prices. This is the minimal model of financial complexity (Maymin, [5]).

But how complex is the generated price series? We have seen above that real markets suffer from many irregularities. Specifically, the stylized facts about market returns relative to independently distributed Normal returns are that real returns have higher kurtosis (fatter tails), negative skewness (more extreme jumps down), and a rich panoply of autocorrelations (generating mean reversion or momentum at different horizons).

By taking a lookback window of $n = 22$ and partitioning the up and down ticks into buckets large enough to interpret their rolling sum as a daily return, we can estimate the implied kurtosis, skewness, and correlations of the resulting

price series. Surprisingly enough, it turns out that all of the troublesome stylized facts of real markets occur in the generated price series as well!

Thus, the unique, simple, and minimal model of financial complexity, with no parameters to tweak, serendipitously ends up explaining much of what we see in real markets.

What does the rule do, exactly? Do such traders exist? In general, there need be no easy description of a trading rule. But in this case, there happens to be a very simple explanation. Notice that state 1 is an UP-absorbing state: any UP day will bring the trader to state 1. Similarly, state 2 is a DOWN-absorbing state. Thus, rule 54 ultimately merely compares the two earliest days of its lookback window: if the price change $n-1$ ticks ago were the same as the price n ticks ago, then the investor would sell; if they were different, he would buy.

An alternative interpretation is that the representative investor look each tick and compares it to the previous one. If they are the same, whether both up or both down, he sells; if they are different, either up and then down or down and then up, he buys. However, his order does not take effect immediately but rather experiences a delay of $n-1$ ticks. Put this way, such a trading rule can be expressed as a combination of four commonplace rules: profit taking in bull markets, momentum in bear markets, buying on dips, and buying on recoveries.

Naturally, the minimal model can be extended to multiple states, multiple assets, and multiple traders, and complexity again emerges, with more variety as well. But it is interesting that even the minimal model is able to fit actual returns so well, and so much better than random walks or Brownian motions, the standard assumptions of non-NKS-influenced finance.

Clearly, the *NKS* approach is useful in market-based finance. So why is it not more frequently used in academic circles? The reason is selection bias.

The bane of academic financial research is selection bias. Selection bias in data can falsely suggest that certain assets or industries had high expected returns, only because those were the only ones who survived long enough to be in the dataset. Selection bias may even be latent and quite subtle: one of the longest puzzles in finance is the equity premium puzzle documented by Mehra and Prescott [8] in 1985, noting that historical average returns have been far too high to be explained by risk aversion, the standard explanatory tool of financial economics. But we will never know if the selection bias of having had a booming stock market for many decades is what allowed us the luxury of asking why have our stock returns been so large.

But by far the biggest concern is selection bias of the models, also known as data snooping. If we posit a model that is influenced by what we have seen, then tests of the model are contaminated. At the extreme, you can always optimize the parameters of any family of models to get the best possible fit, but you will never know if you are not just overfitting noise.

Partially in an attempt to combat this problem, and partially because finance is often viewed as a discipline of economics, academic literature in the area is virtually required to motivate any analysis with detailed reasoning why the model makes sense *a priori*. Of course, it is impossible to tell by reading a paper

whether the model indeed was formulated prior to any observation of the data or whether it was retrofit onto it later, or, less obviously, whether it was just the lucky one of many models tested that happened to work. Academics rarely (though not never) publish the results of failed models.

This attachment to motivation is the biggest hurdle to wider acceptance of the useful tools and techniques of the *NKS* framework. Mining the computational financial universe requires abandoning all preconceptions of what should or should not work and instead trying hundreds, thousands, millions of possibilities to see what does indeed work. By the Principle of Computational Irreducibility, the motivation game can not work in general, and can even be a hindrance to the truth. The *NKS* approach to market-based finance requires overcoming enormous inertia to flip standard academic practice completely on its head.

That's a tough row to hoe, but there have been some other inroads. Explicitly, Zenil and Delahaye [12] investigate the market as a rule-based system by comparing the distributions of binary sequences from actual data with those resulting from purely algorithmic means. On a more implicit level, many otherwise standard-seeming financial results seem to be more willing to test literally all possible strategies or combinations, reserving their motivation and justification only to the form of the model. The tide may not have started to turn yet, but the waves are starting to froth.

2 Government-Based Approaches

Markets resulting from government fiat tend to be simple price fixings. Even the ostensibly more general price floors or ceilings end up being price fixings anyway because otherwise the legislation is useless. So a time series of government-controlled prices tend to look like a constant, experiencing nearly zero volatility... until the government can no longer control the price and the pent-up volatility explodes all at once. Imagine a currency peg about to break or the stock market hitting an automatic circuit breaker curbing trading. When trading resumes, the true price will likely be very different from the most recently reported price.

In exploring regulatory issues and their possible effects on markets, there are two traditional approaches: theoretical and econometric. The theoretical approach solves for the equilibrium in a particular standard model and evaluates how it changes under different regulatory regimes. The econometric approach attempts to analyze past regulatory changes to isolate the effects of unanticipated regulatory changes. These two approaches sometimes agree and sometimes disagree, and each has its own pitfalls.

A unifying way of viewing both approaches is to observe that they each effectively assume a particular process for the evolution of market prices, and then translate regulatory changes into different values for the particular parameters. Theoretical models attempt to solve for what the new parameters will be while the econometric models attempt to estimate them from the historical record.

There is a third way, the *NKS* way: one could use a rules-based approach with regulatory overrides. Specifically, one could imagine the rule 54 trader wanting

to sell the asset but being stopped by government forces intent on propping up the market.

This is now a question of computational search. It is unclear ahead of time what the effect will be. The best way to find out is to simulate it. In Maymin [4], I did just that, and showed that regulation in general makes market price processes appear to be more Normal and less complex (until, of course, the regulation can no longer be afforded). Particular periods, however, could actually appear even worse than the non-regulated version. Further, the results from pricking bubbles and propping up crashes are not symmetrical: specifically, if regulations were to prick apparent bubbles, then propping up apparent crashes makes no additional difference (see Fig. 4).

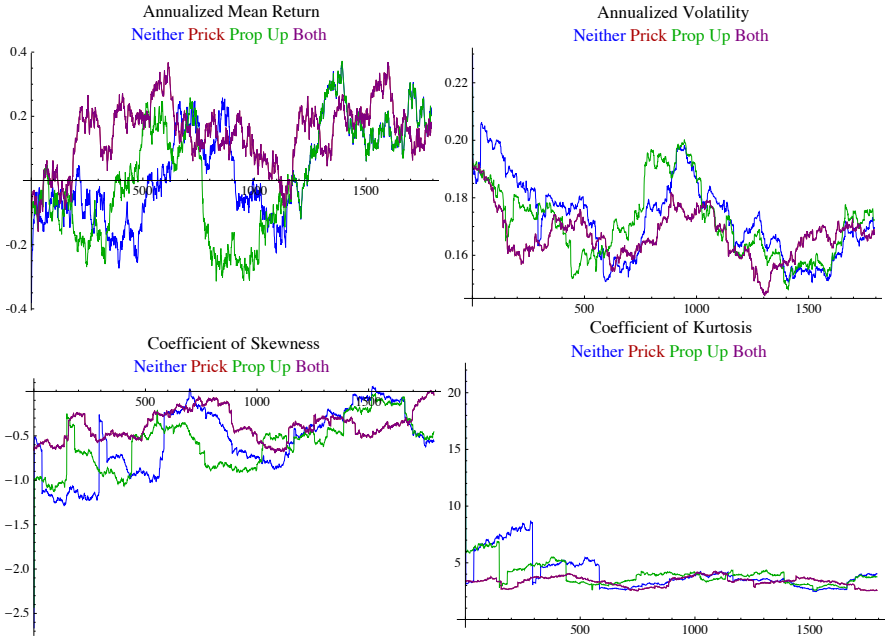


Fig. 4. The graphs show rolling moment estimates from these four different regulatory regimes

An even more direct result can be found in Maymin and Lim [7] where we compare regulations directly on a cellular automaton model. In the context of environmental regulations, suppose each cell represents an entity that can choose whether or not to pollute. And suppose the rule governing whether you pollute or not depends entirely on what you and your neighbors did in the previous instance. For concreteness, let's say it is Wolfram rule 110, which he has shown to be computational universal, or maximally complex (see Fig. 5).

With the *NKS* approach to regulation in general, both financial and otherwise, we are able to see the effects of varying kinds of regulatory overrides on top of a

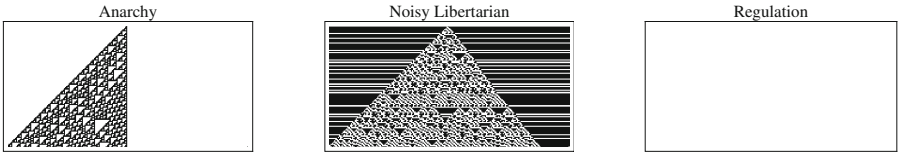


Fig. 5. Call anarchy the state of no overriding law, neither *a priori* regulation nor *ex post* justice. Then the half of the people on the right hand side never pollute, while those who occasionally pollute exhibit interesting, indeed maximal, complexity. Under complete *a priori* regulation, no one would pollute ever, leading to zero complexity. But under *ex post* albeit noisy justice in which with some probability those who polluted last time will now be polluted on by those who had abstained, maximal complexity is restored. Furthermore, even that half of the population that would not have polluted under anarchy now does occasionally pollute. Bearing in mind that pollution is a cost with associated benefits, and that some amount of pollution is likely to be optimal, we can draw conclusions about which system accomplishes what we want.

simple system of otherwise static rules. I expect that for the government-based strand of finance research, the *NKS* approach will eventually come to dominate the field, as it represents the only way I can see of performing true experiments on the possible effects of different proposed regulations.

3 Practitioners

While academics and regulators play a loud part in finance, the silent supermajority are practitioners: traders, investors, and speculators who have a vested interested in keeping quiet and keeping secrets. Practitioners do not care how to pronounce the word "finance," and they switch randomly from one to the other. They represent by far the most important constituency. Can an *NKS* approach help them too?

In one sense, they represent the heart of the *NKS* approach. Markets are complex but by the Principle of Computational Equivalence they are no more complex than other maximally complex things. Complex things can often be modeled by simple rules. When even the simplest of rules constitute an astronomical number of possibilities, the only possible approach, by the Principle of Computational Irreducibility, is exhaustive or random search. Thus, together, practitioners are essentially mapping and mining the financial computational universe, even if they are doing so unintentionally and occasionally redundantly.

It turns out that this task of finding a profitable strategy in past prices is one of the hardest computational problems on the planet. Indeed, I have shown that this task is as hard as solving satisfiability or the traveling salesman problem. In other words, markets will be efficient—that is, there will be no profitable trading strategies based on past prices because they would have all been discovered and exploited—only if all other difficult problems have also been solved.

Surprisingly enough, I have also shown the converse: that if the markets happen to be efficient, then we can actually use those markets to solve the other difficult problems. We can, in effect, "program" the market to solve general computational problems.

Thus, market efficiency and computational efficiency turn out to be the same thing. This paper, Maymin [6], sparked the creation of *Algorithmic Finance*, a new journal and indeed a new field launched specifically to continue the insights from merging computational efficiency and market efficiency. I am the managing editor of the journal and Stephen Wolfram is on the advisory board. With this journal, we hope to continue the journey of exploring *NKS*-inspired approaches to the field of finance.

4 Conclusions

The insights from *NKS* are general, deep, and broad: simple rules can generate complexity; beyond a small threshold, all complexity is maximal complexity; the only way of evaluating even simple systems that generate complexity is to run them and see. In finance, these insights are critical for understanding markets and their evolution, particularly as trading moves ever closer to complete automation.

Both the journal *Algorithmic Finance* and the field of algorithmic finance rely on these insights to grow. Applications as varied as high frequency finance and automated trading, the heuristics of behavioral investors, news analytics, statistical arbitrage, and dynamic portfolio management all reside at the intersection of computer science and finance, and could, and have, and will continue to benefit from the tools of *NKS*.

References

[1] Ehrentreich, N.: Agent-Based Modeling: The Santa Fe Institute Artificial Stock Market Model Revisited. Springer (2007)
[2] Gilbert, N.: Agent-Based Models. Sage Publications (2007)
[3] Jegadeesh, N., Titman, S.: Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency. Journal of Finance 48(1), 65–91 (1993)
[4] Maymin, P.Z.: Regulation Simulation. European Journal of Finance and Banking Research 2(2), 1–12 (2009)
[5] Maymin, P.Z.: The Minimal Model of Financial Complexity. Quantitative Finance 11(9), 1371–1378 (2011)
[6] Maymin, P.Z.: Markets are Efficient If and Only If P=NP. Algorithmic Finance 1(1), 1–11 (2011)
[7] Maymin, P.Z., Lim, T.W.: The Iron Fist vs. the Invisible Hand: Interventionism and libertarianism in environmental economic discourses. World Review of Entrepreneurship, Management and Sustainable Development 8(3), 358–374 (2012)
[8] Mehra, R., Prescott, E.C.: The equity premium: A puzzle. Journal of Monetary Economics 15(2), 145–161 (1985)

[9] Wilensky, U.: NetLogo. Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston (1999), http://ccl.northwestern.edu/netlogo/
[10] Wolfram, S.: A New Kind of Science. Wolfram Media (2002)
[11] Wolfram, S.: Informal essay: Iterated finite automata (2003), http://www.stephenwolfram.com/publications/recent/iteratedfinite/
[12] Zenil, H., Delahaye, J.-P.: An Algorithmic Information Theoretic Approach to the Behaviour of Financial Markets. Journal of Economic Surveys 25(3), 431–463 (2011)

Chapter 9
The Relevance of *Computation Irreducibility* as *Computation Universality* in Economics*

K. Vela Velupillai

Department of Economics/ASSRU, University of Trento, Italy

Abstract. Stephen Wolfram's *A New Kind of Science* [26] should have made a greater impact in economics—at least in its theorising and computational modes—than it seems to have. There are those who subscribe to varieties of agent-based modelling, who do refer to Wolfram's paradigms–a word I use with the utmost trepidation–whenever simulational exercises within a framework of cellular automata is invoked to make claims on complexity, emergence, holism, reduction and many such 'buzz words'. Very few of these exercises, and their practitioners, seem to be aware of the deep mathematical–and even metamathematical–underpinnings of Wolfram's innovative concepts, particularly of computational equivalence and computational irreducibility in the works of Turing and Ulam. Some threads of these foundational underpinnings are woven together to form a possible tapestry for economic theorising and modelling in computable modes.

Keywords: Computational equivalence, computational irreducibility, computation universality.

1 A Preamble on the Origins of the Visions[1]

> "From that point on [i.e., from January, 1982], Wolfram's bibliography, his list of scientific production, goes from no cellular automata at all to 100 per cent cellular automata. *He decided that cellular automata can do anything.* From that moment on Stephen Wolfram became the Saint Paul of cellular automata."
>
> Toffoli, in ([10], p. 238); italics added.

Even as we celebrate a decade of experiences and adventures with **A New Kind of Science**[2], we should bear in mind that the romantic seeds of the

* I am greatly indebted to my friend, Hector Zenil, for the kind invitation to contribute to this important commemorative volume. One minor caveat should be added here. I subscribe to the entirely sensible view of Chris Moore & Stephan Mertens that 'old fashioned topics, like formal languages and automata' are better left out of discussions at the frontiers of the theory of computation ([7], p. xvii).

[1] I shall assume readers of this volume will have at least a nodding acquaintance with the concepts of the *Principle of Computational Equivalence*, *Computational Irreducibility* and the related notion of *Algorithmic Incompressibility* (cf., for example [6], chapter 6).

[2] Henceforth, referred to as **NKS**.

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 101–111.
DOI: 10.1007/978-3-642-35482-3_9

vision that became the core of NKS, *computational irreducibility*, were sown at a quintessential private[3] *Island in the Sun* in the Carribean, Moskito Island in January, 1982, thirty years ago–and that this is also the year we commemorate the Birth Centennial of Alan Turing, whose genius lies at the heart of Wolfram's sustained efforts at creating a new paradigm for scientific practice, particularly from an *inductive*[4] point of view.

It seems to me, as an economist who practices its formalisation in computable and dynamical systems theory modes, that the *Computational Irreducibility* vision comes against the backdrop of an all permeating *Complexity Vision* of economics. I am convinced that this is, proverbially, a case of placing the cart(s) before the horse(s): it is computability and dynamical systems theory - and their fertile interaction - that underpins the computational irreducibility vision which, in turn, provides one kind of foundation[5] for a complexity vision, particularly of economics.

Emergence, *order*, *self-organisation*, *turbulence*, *induction*, *evolution*, *criticality*, *adaptive*, *non-linear*, *non-equilibrium* are some of the words that characterise the conceptual underpinnings of the 'new' *sciences of complexity* that seem to pervade some of the frontiers in the natural, social and even the human sciences. Not since the heyday of *Cybernetics* and the more recent brief-lived ebullience of *chaos* applied to a theory of everything and by all and sundry, has a concept become so prevalent and pervasive in almost all fields, from Physics to Economics, from Biology to Sociology, from Computer Science to Philosophy as *Complexity* seems to have become. An entire Institution, with high-powered scientists in many of the above fields, including several Nobel Laureates from across the disciplinary boundaries as key permanent or visiting members, has come into existence with the specific purpose of promoting the *Sciences of Complexity*[6].

[3] Owned by Ed Fredkin who himself, with the **Fredkin-Zuse Thesis** which, put in an ultra simple way, states that **The Universe is a Computer**, espouses a vision that is embodied in **NKS**. Fredkin articulated this vision much later than he had conceptualised it in his actual adherence to working with such a metaphor as his guiding scientific, inductive, principle.

[4] I do not want to emphasise Inductive at the expense of *Abductive*, especially by contrasting the former exclusively as an alternative to Deductive. The mischief indulged in by economists, particularly those advocating blind agent-based modelling in economics and finance, claiming that their practice makes the case against formal mathematics in its deductive underpinnings, enhancing the case for a 'new mathematics' that is inductively based, shunts research towards pointless ends.

[5] I have in mind, as another kind of foundation, the philosophy and epistemology that came with the vision of the *British Emergentists*.

[6] I am referring, of course, to the *Santa Fe Institute*, which, refreshingly, has thought it prudent to have a permanent Economics division from the outset. But, on a sceptical note, the almost untrammelled enthusiasm for a unified vision for all of the disciplines has the danger, in my opinion, of making essentially moral, human and social sciences like economics a handmaiden to the concepts and methods of the natural sciences and, in this sense, we seem to be travelling along well trodden and mistaken paths of the past. Vico's famous dictum keeps coming back to haunt my mind: *'Corsi e ricorsi . . .'*!

I have found Duncan Foley's excellent characterisation of the *objects* of study by the 'sciences of complexity' in ([3], p.2) (italics added), extremely helpful in providing a base from which to approach the study of a subject that is technically demanding, conceptually multi-faceted and philosophically and epistemologically highly inhomogeneous:

> Complexity theory represents an ambitious effort to analyse the functioning of highly organized but decentralized systems composed of very large numbers of individual components. The basic processes of life, involving the chemical interactions of thousands of proteins, the living cell, which localizes and organizes these processes, the human brain in which thousands of cells interact to maintain consciousness, ecological systems arising from the interaction of thousands of species, the processes of biological evolution from which *new species emerges*, and *the capitalist economy, which arises from the interaction of millions of human individuals, each of them already a complex entity, are leading examples.*

It is one thing to observe similarities at a phenomenological and structural level. It is quite another to claim that one 'science', with its own characteristic set of methods, can encapsulate their study in a uniform way, thus providing rationale for an interdisciplinary approach to all of them. Here again, I believe the elegant attempt to go just below the surface similarities of phenomena and structure, and define the conceptual and methodological underpinnings of this new 'science' in Foley (*ibid*), is most illuminating:

> What these [highly organized but decentralized] systems share are a **potential** to configure their component parts in an astronomically large number of ways (they are *complex*), constant change in response to environmental stimulus and their own development (they are *adaptive*), a strong **tendency** to achieve recognizable, stable patterns in their configuration (they are *self-organising*), and an avoidance of stable, **self-reproducing** states (they are *non-equilibrium systems*). The task complexity science sets itself is the **exploration of the general properties of complex, adaptive, self-organizing, non-equilibrium systems**.
>
> The *methods* of complex systems theory are highly empirical and inductive. ... A characteristic of these ... complex systems is that their components and rules of interactions are *non-linear* .... **The computer plays a critical role in this research**, because it becomes impossible to say much directly about the dynamics of non-linear systems with a large number of degrees of freedom using **classical mathematical analytical methods**.
>
> *ibid*, p.2; bold emphasis added.

Note, however, that the discourse is about *potentials* and *tendencies* and, therefore, in an economic context, but not only in it, there could be scope for *design* or

policies. Moreover, the 'avoidance of stable, self-reproducing states' is an indictment against mechanical growth theories, of a macroeconomic sort, with their uninteresting, stable, attractors.

This is exactly where the notion of *computational irreducibility*, interpreted in economic contexts as *computation universality*, especially within what I have come to call *Computable Economics*[7], plays an important role, both at the level of individual behaviour and institutional design - or, both microeconomically and macroeconomically. In the former case, i.e., microeconomically, at the level of individual behaviour, the notion allows one to show that the much maligned Simonian concept of Bounded Rationality encapsulates, naturally and generally, the normal practice of rational behaviour; in the latter case, i.e., macroeconomically, it allows one to derive an impossibility theorem on policy.

One final cautionary note has to be added, lest the unwary practitioner of indiscriminate reliance on 'pattern recognition' of computer graphics is lulled into thinking that the pixels on the screen are independent of the mathematics of the computer—i.e., recursion theory. It is necessary to remember the following classical result from recursion theory:

Letφ and ζ be a partial and a total function, respectively. Then:

- φ is partial recursive *iff* its graph is *recursively enumerable*.
- ζ is recursive *iff* its graph is *recursive*.

This kind of result is alien to the practitioners of varieties of agent-based modelling, who seem to rely on blind computer graphics for significant inductive propositions. Stephen Wolfram never made such a mistake—nor did, naturally, Turing!

2 Computational Irreducibility as Computation Universality, Implied by the Principle of Computational Equivalence

> "And what the *Principle of Computational Equivalence* implies is that in fact almost any system whose behaviour is not obviously simple will tend to exhibit *computational irreducibility*."
> **NKS** ([26], p. 745); italics added.

I have, within the framework of *Computable Economics*, attempted to formalise the notion of *computation universality*, as against *computationally irreducible*, in terms of the computing *trajectories* of finite automatons and Turing Machines, respectively, with the formalisation of the notion of *trajectories* in terms of formal *dynamical systems*. Moreover, my definition of the dynamics of a *complex* economic system as one capable of computation universality was, I realised with

[7] A brief, but rigorously intuitive–in the senses in which the *Church-Turing Thesis* can be referred to in this way–characterisation of what I mean by *Computable Economics* can be gleaned from [4]. A more detailed, but already dated, characterisation is in [11].

the benefit of hindsight, exactly similar to a 'system whose behaviour is *not obviously simple*' and, therefore, one which 'will tend to exhibit computational irreducibility'. What I have claimed, explicitly, is an equivalence between the notion of *complex* and *not obviously simple*. I go further and assert that it is not possible to prove–by means of any notion of proof–a formal equivalence between the notion of *complex* and the phrase *not obviously simple*–except, for example, by means of invoking something like computationally *reducible* and enunciating a thesis that *simple* is equivalent to *reducible*[8]. It was, then, straightforward to identify the behaviour of a finite automaton with one that is computationally reducible, for example one that is not subject to the *Halting problem for Turing Machines* or, more pertinently, one that is only capable of *trivial* dynamics, with the notion of 'trivial' given a formalism via *Rice's Theorem*. In this way *simplicity*, *reducibility* and *triviality* can be coupled to the computing dynamics of *finite automata* and *complexity*, i*rreducibility* and *non-triviality* to those trajectories that are routine for a *Turing Machine*.

Finally, Wolfram's characteristically fertile assumption, one which I have come to call **Wolfram's Thesis** ([26], p. 715)

Claim 1. *All processes can be viewed as computations*

I claim that this is a kind of 'dual' to the following variant of the Church-Turing Thesis (cf. [1], p. 34):

Claim 2. *Every rule is a recursive rule*

With this conceptual background at hand, I can illustrate the derivation of two fundamental results in microeconomics and macroeconomics–one on the notion of rational behaviour and the other on the feasibility of effective policy in a complex (dynamic) economy.

Definition 3. ***Invariant set***
A set (usually compact) $S \sqsubset U$ *is* ***invariant*** *under the flow* $\varphi(.,.)$ *whenever* $\forall t \in \mathbb{R}$, $\varphi(.,.) \sqsubset S$.

Definition 4. ***Attracting set***
A closed invariant set $A \sqsubset U$ *is referred to as the* ***attracting set*** *of the* ***flow*** $\varphi(t,x)$ *if* $\exists$ *some neighbourhood* V *of* A, *s.t* $\forall x \in V$ *&* $\forall t \geq 0$, $\varphi(t,x) \in V$ *and:*

$$\varphi(t,x) \to A \text{ as } t \to \infty \tag{1}$$

Remark 5. *It is important to remember that in dynamical systems theory contexts the attracting sets are considered the* ***observable*** *states of the dynamical system and its flow.*

[8] Or the notion of simplicity to be formally identified with *reducibility in a computational sense*. I am convinced that this corresponds exactly to the way Kemeny used the notion of *simplicity in induction* in his classic contribution which lies at the basis of *algorithmic complexity* ([5];[6]).

Definition 6. *The basin of attraction of the attracting set A of a flow, denoted, say, by Θ_A, is defined to be the following set:*

$$\Theta_A = \cup_{t \leq 0} \varphi_t(V) \tag{2}$$

where: $\varphi_t(.)$ denotes the flow $\varphi(.,.), \forall t$.

Remark 7. *Intuitively, the basin of attraction of a flow is the* ***set of initial conditions*** *that eventually leads to its attracting set - i.e., to its limit set (limit points, limit cycles, strange attractors, etc).*

Definition 8. ***Dynamical Systems capable of Computation Universality***
A dynamical system capable of computation universality is one whose defining initial conditions can be used to program and simulate the actions of any arbitrary Turing Machine, in particular that of a Universal Turing Machine.

Proposition 9. *Dynamical systems characterizable in terms of limit points, limit cycles or 'chaotic' attractors, called 'elementary attractors', are not capable of universal computation.*

Theorem 10. *There is no effective procedure to decide whether a given observable trajectory is in the basin of attraction of a dynamical system capable of computation universality*

Proof. The first step in the proof is to show that the basin of attraction of a dynamical system capable of universal computation is recursively enumerable but not recursive. The second step, then, is to apply Rice's theorem to the problem of membership decidability in such a set. First of all, note that the basin of attraction of a dynamical system capable of universal computation is recursively enumerable. This is so since trajectories belonging to such a dynamical system can be effectively listed simply by trying out, systematically, sets of appropriate initial conditions. On the other hand, such a basin of attraction is not recursive. For, suppose a basin of attraction of a dynamical system capable of universal computation is recursive. Then, given arbitrary initial conditions, the Turing Machine corresponding to the dynamical system capable of universal computation would be able to answer whether (or not) it will halt at the particular configuration characterising the relevant observed trajectory. This contradicts the unsolvability of the Halting problem for Turing Machines. Therefore, by Rice's *theorem*, there is no effective procedure to decided whether any given arbitrary observed trajectory is in the basin of attraction of such recursively enumerable but not recursive basin of attraction. ■

Given this result, it is clear that *an effective theory of policy is impossible in a complex economy.* Obviously, if it is effectively undecidable to determine whether an observable trajectory lies in the basin of attraction of a dynamical system capable of computation universality, it is also impossible to devise a policy—i.e., a recursive rule—as a function of the defining coordinates of such an observed or observable trajectory. Just for the record I shall state it as a formal proposition:

Proposition 11. *An effective theory of policy is impossible for a complex economy*

Remark 12. *The 'impossibility' must be understood in the context of effectivity and that it does not mean specific policies cannot be devised for individual complex economies. This is similar to the fact that non-existence of general purpose algorithms for solving arbitrary Diophantine equations does not mean specific algorithms cannot and have not been found for special, particular, such equations.*

What if the realized trajectory lies outside the basin of attraction of a dynamical system capable of computation universality and the objective of policy is to drive the system to such a basin of attraction? This means the policy maker is trying to design a dynamical system capable of computational universality with initial conditions pertaining to one that does not have that capability. Or, equivalently, an attempt is being made, by the policy maker, to devise a method by which to make a Finite Automaton construct a Turing Machine, an impossibility. In other words, an attempt is being made endogenously to construct a 'complex economy' from a 'non-complex economy'. Much of this effort is, perhaps, what is called 'development economics' or 'transition economics'. Essentially, my claim is that it is recursively impossible to construct a system capable of computation universality using only the defining characteristics of a Finite Automaton. To put it more picturesquely, a *non-algorithmic step* must be taken to go from systems *incapable* of self-organisation to ones that are capable of it. This interpretation is entirely consistent with the original definition, explicitly stated, of an '*emergent property*' or an '*emergent phenomenon*', by George Henry Lewes. This is why 'development' and 'transition' are difficult issues to theorise about, especially for policy purposes.

Next, consider a (rational) *problem solving* entity to be an *information processing system* ([8]). The strategy for my formalization exercise can be summarized in the following sequence of steps:

- Extract the procedural content of orthodox rational choices (in theory).
- Formalize such a procedural content as a process of computation.
- Given the formalized procedural content as a process of computation, to be able to discuss its computational complexity.
- Show the equivalence between a process of computation and a suitable dynamical system.
- To, then, show the possibility of non-maximum rational choice.
- Then, to show that such behaviour is that which is manifested by a boundedly rational, satisficing, agent.

The following results encapsulates, formally, the content of the first three steps of the above six-step scheme:

Theorem 13. *The process of rational choice by an economic agent is formally equivalent to the computing activity of a suitably programmed (Universal) Turing machine.*

Proof. By construction. See §3.2, pp. 29-36, [11]. ■

Remark 14. *The important caveat is 'process' of rational choice, which Simon–more than anyone else–tirelessly emphasized by characterizing the difference between 'procedural' and 'substantive' rationality; the latter being the defining basis for Olympian rationality (cf. [9], chapter 1), the former that of the computationally underpinned problem solver facing decision problems. Any decision–rational or not–has a time dimension and, hence, a content in terms of some process. In the Olympian model the 'process' aspect is submerged and dominated by the static optimization operator, By transforming the agent into a problem solver, constrained by computational formalisms to determine a decision problem, Simon was able to extract the procedural content in any rational choice. The above result is a summary of such an approach.*

Theorem 15. *Only dynamical systems capable of computation universality are consistent with rationality in the sense that economists use that term in the Olympian Model.*

Proof. See pp. 49-50, [11]. ■

Remark 16. *This result, and its proof, depend on theorem 13 and, therefore, its background basis, as explained in the Remark following it, given above. In this way, following the Simon's vision and the definition of rationality is divorced from optimization and coupled to the decision problems of an information processing problem solver, emphasizing the procedural acts of choice.*

Theorem 17. *Non-Maximum Rational Choice*

*No trajectory of a dynamical system capable of universal computation can, in any '**useful sense**' (read: '**cannot obviously**') be related to optimization in the Olympian model of rationality.*

Proof. See [13]. ■

Remark 18. *The claim here is, then, that optimization in the Olympican model of rationality is computationally reducible.*

Theorem 19. *Boundedly rational choice by an information processing agent within the framework of a decision problem is capable of computation universality.*

Proof. An immediate consequence of the definitions and theorems of this section. ■

Remark 20. *From this result, in particular, it is clear that the Boundedly Rational Agent, satisficing in the context of a decision problem, encapsulates the only notion of rationality that can 'in any useful sense' be defined procedurally.*

3 A Computable Economist's *Paen* to the NKS Vision[9]

> "Mathematics is not a finished object based on some axioms. It *evolves* genetically. This has not yet quite come to conscious realization. ...
>
> *Mathematics will change. Instead of precise theorems*, of which there are now millions, *we will have, fifty years from now, general theories and vague guidelines*, and the individual proofs will be worked out by graduate students or by computers.
>
> *Mathematicians fool themselves when they think that the purpose of mathematics is to prove theorems, without regard to the broader impact of mathematical results.* Isn't it strange.
>
> *In the next fifty years there will be, if not axioms, at least agreements among mathematicians about assumptions of new freedoms of constructions, of thoughts.* Given an undecidable proposition, there will be a preference as to whether one should assume it to be true or false. Iterated this becomes: some statements may be undecidably undecidable. This has great philosophical interest
>
> Ulam in ([2], pp.310-312); italics added.

Ulam was always a prescient mathematician. In an almost uncanny confirmation of his audacious prediction, Stephen Wolfram's NKS seems to have set out an implementable version of the vision of the kind of *mathematics*—but not quite *science*—that Ulam may have had in mind. It is particularly appropriate that Wolfram achieved this implementable vision of Ulam's vision of *A New Kind of Mathematics* utilising Cellular Automata as his computational paradigm–especially since it was Ulam, together with von Neumann, who pioneered the use of this medium for the kind of questions that are fundamental in the *new sciences of complexity*. Indeed, Wolfram's explicit philosophical and epistemological stances on the nature and evolution of mathematics buttresses my own vision of the mathematical foundations of Computable Economics:

> "[L]ike most other fields of human enquiry mathematics has tended to define itself to be concerned with just those questions that its methods can successfully address. And since the main methods traditionally used in mathematics have revolved around *doing proofs*, questions that involve *undecidability* and *unprovability* have inevitably been avoided. ...
>
> The main point .... is that in both the systems it studies and the questions it asks mathematics is much more a product of its history than is realised."
>
> **NKS** ([26] p. 792); italics added

It is precisely these questions that have *not* been avoided in *Computable Economics* and precisely because, inadvertently, those of us who were busy

[9] The *paen* of this section is a slightly modified version of an initial attempt to pay *homage* to **NKS**, in [12].

developing this alternative vision of doing economics in the mathematical mode underpinned our methodologies and epistemologies in what, with hindsight, could be identified with the Principle of Computational Equivalence (**PCE**), the notion of computational irreducibility and Wolfram's Thesis–albeit in other, but equivalent, formalisms.

If economics is formalised using traditional mathematics, it will be crippled by the poverty of the history that determines that tradition–but it may also be enhanced by the richness of that tradition. It is just that the richness seems an entirely internal history, completely unrelated to the ontology of economics. This, I think, is the point made by Ulam and Wolfram in their indictments of the traditional philosophy and epistemology of mathematics and science, respectively. Economics in the mathematical mode, a step child and a handmaiden to both of these endeavours, in applying the concepts and methods emerging (sic!) from their traditional evolutionary history, would therefore by crippled to the same extent as these noble products of the human mind.

Essentially, computational irreducibility of a computational process, by **PCE** and **Wolfram's Thesis**, implies also *unpredictability*[10]. Wolfram is then able to show that traditional science is computationally *reducible* and, hence, can succeed in local predictability. This, in turn, is because the mathematical laws that encapsulate traditional science are simple enough to be analytically tractable and, hence, are computationally reducible.

In other words, he extracts the implicit processes intrinsic to, and implied by, any mathematical law or formalism in traditional science, and uses the **PCE**, and its phenomenological consequence–computational irreducibility–to evaluate their effective and predictable content to show the simplistic, unrealistic and undesirable nature of mathematical formalism in the traditional sciences.

I believe **NKS** succeeded admirably in showing that the appearance of the successes of traditional science, using the criteria of simplicity in the laws that encapsulate the phenomena to be explained and the predictions that are extracted from them, are the results of a subterfuge: that of concentrating on using analytically solvable mathematical formalisms to encapsulate natural laws and thereby ignoring the *processes* that have to be used to evaluate the solution to any such formalism. In this way, the methodology of traditional science circumvented the implications of computational irreducibility and, hence, ignored **PCE**.

But he may have forgotten that Brouwer could have also been, together with Ulam, a compatriot–with his *Choice Sequences*, which enriching **PCE** and *computational irreducibility* in ways that would have enhanced the positive aspects of his negative results.

All this could have been said, *pari passu*, of the achievements - or, more accurately put, the *non-achievements*–of traditional mathematical and analytical economics.

[10] To the triptych of *undecidability*, *unprovability* and *unpredictability* must be added *unsolvability* and *uncomputability* as the five cardinal unifying conceptual bases that underpin *Computable Economics* (cf. [4]).

References

[1] Beeson, M.J.: Foundations of Constructive Mathematics. Springer, Heidelberg (1985)
[2] Cooper, N.G. (ed.): From Cardinals to Chaos - Reflections on the Life and Legacy of Stanislaw Ulam. Cambridge University Press, Cambridge (1989)
[3] Foley, D.K.: Unholy Trinity: Labor, Capital and Land in the New Economy: The Graz Schumpeter Lectures. Routledge, London (2003)
[4] Kao, Y.-F., Ragupathy, V., Velupillai, K.V., Zambelli, S.: Noncomputability, Unpredictability, Undecidability and Unsolvability in Economic & Finance Theories. Complexity (forthcoming, 2012)
[5] Kemeny, J.G.: The Use of Simplicity in Induction. Philosophical Review LXII, 391–408 (1953)
[6] Li, M., Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications, 2nd edn. Springer-Verlag, New York, Inc. (1997)
[7] Moore, C., Mertens, S.: The Nature of Computation. Oxford University Press, Oxford (2011)
[8] Newell, A., Simon, H.A.: Human Problem Solving. Prentice-Hall, Inc., Englewood Cliffs (1972)
[9] Simon, H.A.: Reason in Human Affairs. Basil Blackwell, Oxford (1983)
[10] Regis (ed.): Who Got Einstein's Office: Eccentricity and Genius at the Institute for Advanced Study. Addison-Wesley, Boston (1987)
[11] Velupillai, K.V.: Computable Economics. Oxford University Press, Oxford (2000)
[12] Velupillai, K.V.: The Unreasonable *In*effectiveness of Mathematics in Economics. Cambridge Journal of Economics 25(6), 849–872 (2005)
[13] Velupillai, K.V.: The Impossibility of an Effective Theory of Policy in a Complex Economy. In: Salzano, M., Colander, D. (eds.) Complexity Hints for Economic Policy, pp. 273–290. Springer, Heidelberg (2007)
[14] Wolfram, S.: A New Kind of Science. Wolfram Media, Champaign (2002)

Chapter 10
Computational Technosphere and Cellular Engineering

Mark Burgin

Department of Mathematics, University of California, USA

Abstract. The basic engineering problem is to build useful systems from given materials and with given tools. Here we explore this problem in the computational technosphere of computers, smartphones, networks and other information processing and communication devices created by people. The emphasis is on construction of different kinds of information processing automata by means of cellular automata. We call this engineering problem cellular engineering. Various types and levels of computing systems and models are considered in the context of cellular engineering.

Keywords: cellular automaton, computational equivalence, engineering, modeling, construction, model of computation, grid automaton.

1 Introduction

Stephen Wolfram [11] suggested the *Principle of Computational Equivalence*, which asserts that systems found in the natural world can perform computations up to a maximal ("universal") level of computational power, and that most systems do in fact attain this maximal level of computational power. Consequently, most systems performing recursive computations are computationally equivalent in general and equivalent to cellular automata in particular. Here we consider a technological counterpart of this Principle, which is related not to nature but to the *technosphere* created by people. The technosphere is the world of all technical devices. In it, computers and other information processing systems play the leading role. Taking all these devices, we obtain the *computational technosphere*, which is an important part of the technosphere as a whole. The computational technosphere has its own Principle of Computational Equivalence. It is called the *Church-Turing Thesis*. There are different versions of this Thesis. In its original form, it states that *the informal notion of algorithm is equivalent to the concept of a Turing machine* (the Turing's version) or that *any computable function is a partial recursive function* (the Church's version). The domineering opinion is that the Thesis is true as it has been supported by numerous arguments and examples. As a result, the Church-Turing Thesis has become the central pillar of computer science and implicitly one of the cornerstones of mathematics as it separates provable propositions from those that are not provable. In spite of

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 113–124.
DOI: 10.1007/978-3-642-35482-3_10

all supportive evidence and its usefulness for proving various theoretical results in computer science and mathematics, different researchers, at first, expressed negative opinion with respect to validity of the Church-Turing Thesis, and then build more powerful models of algorithms and computations, which disproved this Thesis. It is possible to find the history of these explorations in [6]. Here we go beyond the computational technosphere, suggesting the *Technological Principle of Computational Equivalence* for the whole technosphere. It asserts:

For any technical system, there is an equivalent cellular automaton.

This principle also has a constructive form:

For any technical system, it is possible to build (find) an equivalent cellular automaton.

Here we consider only the computational form of the *Technological Principle of Computational Equivalence*. It is expressed as the *Computational Principle of Technological Equivalence*:

For any information processing system, it is possible to build (find) an equivalent cellular automaton.

Note that in this Principle cellular automata are not restricted to classical cellular automata. There are much more powerful cellular automata. For instance, inductive cellular automata can solve much more problems than classical cellular automata or Turing machines. Building technical systems is an engineering problem. That is why in Section 2, we discuss computational engineering, which is rooted in the work of von Neumann who used a special kind of computational engineering, or more exactly, cellular engineering, for building self reproducing automata [28]. He also demonstrated that construction of complex systems using cellular automata allows one to essentially increase reliability of these systems. However, to be able to rigorously demonstrate validity of the *Computational Principle of Technological Equivalence*, as well as of the *Technological Principle of Computational Equivalence* and Wolfram's *Principle of Computational Equivalence*, it is necessary to ascribe exact meaning to terms used in these principles. That is why in Section 3, we introduce and analyze different types of computational and system equivalence. In Section 4, we demonstrate possibilities of cellular engineering in modeling and construction, giving supporting evidence for the *Computational Principle of Technological Equivalence*. Some of these results were obtained in [7], while other results are new.

2 Computational Engineering

It is possible to describe an engineering problem in the following way. Given working materials and tools for operation, build/construct a system (object) that satisfies given conditions. Here we consider a specification of such a problem for computational (information processing) systems. Thus, we have the following initial conditions:

- A class **K** of computational (information processing) automata is given.
- A class **H** of computational (information processing) automata is provided.
- A set **A** of composition operations is made available.
- A type ϕ of automata equivalence is offered.

Task 1: For any automaton H from **H**, construct an automaton K from **K** by means of operations from **A**, such that K is ϕ-equivalent to H.

Task 2: For any automaton H from **H**, construct an automaton A ϕ-equivalent to H using operations from **A** and automata from **K** as the building material for operations from **A**.

Note that in the second case, the automaton A does not necessarily belong to the class **K**.

The area where such problems are solved is *computational engineering*. When the class **K** consists of cellular automata, i.e., we construct using cellular automata as the construction media, the construction problem is in the scope of *cellular engineering* introduced and studied in [7]. Another basic problem of cellular engineering is construction of different automata, such as pushdown automata, Turing machines and others, using cellular automata as building bricks, blocks and modules. In this case, the result of construction is a grid automaton [6] in a general case and only in some cases it can be a cellular automaton, which is a particular case of grid automata.

Note that there is one more type of computational engineering problems. In it, we have the following initial conditions:

- A class **K** of computational (information processing) automata is given.
- A set **A** of composition operations is made available.
- A goal σ is offered.

Task 3: Using operations from **A** and automata from **K** as the building material for operations from **A**, construct an automaton A that allows one to achieve the goal σ. Usually such a goal σ represents realization of certain functions and satisfaction of selected conditions.

There are three main types of cellular engineering:

- *Process cellular engineering* is aimed at building a cellular automaton to reproduce, organize, model or simulate some process.
- *Function cellular engineering* is aimed at building a cellular automaton to reproduce, organize, model or simulate some function.
- *System cellular engineering* is aimed at building a cellular automaton to reproduce or model some system with its subsystems, components and elements.

Traditional engineering problems for cellular automata are mostly related to process cellular organization or reproduction, that is, how to get a process with necessary characteristics in a cellular automaton. Only sometimes functions are modeled like when cellular automata are used to model functioning of a Turing machine. System cellular engineering reproduces (models) a system with some

level of detailing. For instance, it is possible to represent a system at the level of its elements or at the level of its components.

The area of cellular automata can be divided into three big subareas: *CA science, CA computation*, and *CA engineering*. CA science studies properties of cellular automata and particular, their dynamics or how they function. CA computation uses cellular automata for computation, simulation, optimization, and generation of evolving processes. CA engineering is aimed at constructing different devices from cellular automata. All three areas are complementary to one another.

Ideas similar to the concept of cellular engineering were also discussed by Deutsch in the form of *constructor theory* and *verifiable metaphysics* [10].

Cellular automata are the simplest uniform models of distributed computations and concurrent processes. Grid automata are the most advanced and powerful models of distributed computations and concurrent processes, which synthesize different approaches to modeling and simulation of such processes [4, 6].

Informally a *grid automaton* is a system of automata, which are situated in a grid and called nodes. Some of these automata are connected and interact with one another. It is possible to find formal definitions and elements of the theory of grid automata in [4, 6].

Cellular automata are special cases of grid automata although, in general, grid automata are non-uniform. Our goal is not to substitute cellular automata by grid automata, but to use cellular automata as the basic level for building hierarchies of grid automata. The reason for doing this is to reduce complexity of the description of the system and its processes. For instance, computer hardware has several levels of hierarchy: from the lowest logic gate level to the highest level of functional units, such as system memory, CPU, keyboard, monitor, printer, etc. In addition, as Clark writes (cf. [15]), all good computer scientists worship the god of modularity, since modularity brings many benefits, including the all-powerful benefit of not having to understand all parts of a problem at the same time in order to solve it. That is why one more goal of this paper is to introduce modularity into the realm of cellular automata, making possible to get better understanding and more flexible construction tools without going into detailed exposition of the lower levels of systems. As a result, we develop a computing hierarchy based on cellular automata.

Cellular engineering is an approach complimentary to evolutionary simulation and optimization. Evolutionary simulation is aimed at modeling complex behavior by simple systems, such as cellular automata. Evolutionary optimization is aimed at improving systems by simple means of automata, such as cellular automata, which imitate natural evolutionary processes. Cellular engineering is aimed at constructing complex systems using simple systems, such as cellular automata. In evolutionary processes, systems are evolving subject to definite rules. In engineering, systems are purposefully constructed according to a designed plan.

3 Types of System Equivalence, Modeling and Construction

When researchers discuss equivalence of different models of computation, they are, as a rule, dealing only with one type of equivalence - functional equivalence. The reason is that initially computation performed only computation of functions. Later with an advent of electronic computers, computation enormously expanded its domain but the initial imprinting continues to influence computer science.

At the same time, there is a variety of different types and kinds of equivalence between computational models, automata, software systems, information processing systems and computer hardware. We consider only some of them:

1. Functional equivalence.
2. Linguistic equivalence.
3. Computational equivalence.
4. Structural equivalence.
5. Complexity functional equivalence.
6. Local functional equivalence.
7. Operational or process equivalence.
8. Local operational equivalence.

Let us consider definitions of these types. Two classes of algorithms/automata are *functionally equivalent* if they compute the same class of functions. Two classes of algorithms/automata are *linguistically equivalent* if they compute the same class of languages. Two classes of algorithms/automata are *computationally equivalent* if what is possible to compute in one class it is also possible to compute in the other class. Two classes of algorithms/automata are *operationally or processually equivalent* if they generate the same class of computational processes. Two classes of algorithms/automata are *locally operationally equivalent* if they can perform the same class of computational operations. Two classes of algorithms/automata are *functionally equivalent with respect to complexity* if they compute the same class of functions with the same complexity. Two classes of algorithms/automata are *functionally equivalent with respect to completion* if they compute the same class of functions with the same (level of) complexity. All these definitions describe direct types of equivalence. At the same time, there are more advanced but also useful transcribed types of equivalence. *Transcribed equivalence* includes coding and decoding. For instance, an automaton or a software system B is *functionally equivalent with transcription* to an automaton or a software system A if there are two automata (software systems) C, D, F and G such that for any input X to A, we have $A(X) = D(B(C(X)))$ and for any input Y to B, we have $B(Y) = G(B(F(X)))$. In these processes, C and F are coders of information, while D and G are decoders of information. Functional and processual types of equivalence bring us to the concept of modeling.

Definition 1. It is possible to model an abstract automaton A by a cellular automaton C if there is a configuration W of cells from A and a system R

of states of cells from W such that after initializing these states, the cellular automaton C works as the automaton A.

This modeling relation is related either to process cellular engineering or to function cellular engineering. It is necessary to remark that modeling relation plays an important role not only in information processing systems or other technical systems but also in all life processes and living systems [16, 12]. In some cases, individual cellular engineering allow us to perform cellular engineering for classes of automata.

Definition 2. It is possible to *model* a model **M** of computation in a class **C** of cellular automata if it is possible to model any automaton A from **M** by some cellular automaton C from **C**.

There are different types of modeling.

Definition 3. An abstract automaton A is called programmable in a cellular automaton C if there is a configuration W of cells from A and a system R of states of cells from W such that after initializing these states, the cellular automaton C works as the automaton A, that is, with the same input, C gives the same result as A.
This is a *function cellular engineering*. It is defined by the functional equivalence.

We remind [9] that there are two kinds of functional modeling: direct and transcribed. *Direct functional modeling* of an automaton or a software system A by an automaton or a software system B means that given any input X to A, it either does give any result or gives the same result as the automaton (software system) B with the same input.

Transcribed functional modeling includes coding and decoding [9]. Namely, an automaton or a software system B allows transcribed functional modeling of an automaton or a software system A if there are two automata (software systems) C and D such that for any input X to A, we have $A(X) = D(B(C(X)))$. In this process, C is the coder of information, while D is the decoder of information. The process of transcribed functional modeling is described by the diagram in Fig. 1.

As in a general case, we can realize function cellular engineering for classes of automata.

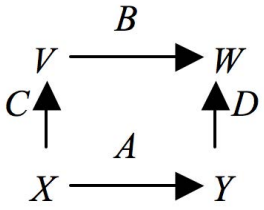


Fig. 1. The process of transcribed functional modeling

Definition 4. A model of computation **M** is called *programmable* in a class **C** of cellular automata if any automaton A from **M** is programmable in some cellular automaton C from **C**.

Definition 5. An abstract automaton A is called *constructible* in a cellular automaton C if there is a configuration W of cells from A and a system R of states of cells from W such that after initializing these states, the cellular automaton C works as the automaton A. and to each structural component D of A some part B of the automaton C is corresponded in such a way that B works as D.

This gives us the construction relation related to *system cellular engineering*. It is defined by the structural equivalence.

Note that both modeling relation "A models B" and construction relation "A is constructed in B" are special cases of the fundamental triad [5].

Definition 6. A model of computation **M** is called *constructible* in a class **C** of cellular automata if any automaton A from **M** is constructible in some cellular automaton C from **C**.

To construct definite devices, we need elements from which we construct and algorithms how to do this. There are three main element types (in information typology), which correspond to the three main types of information operations described in [3]:

- Computational elements or *transformers*.
- Transaction elements or *transmitters*.
- Storage elements or *memory cells*.

There are three element types (in dynamic typology), which correspond to their dynamic:

- Elements with a fixed structure.
- Reconfigurable elements.
- Switching elements.

Elements with a fixed structure have the same structure during the whole process. *Reconfigurable elements* can change their structure during the process. *Switching elements* tentatively change their structure in each operation. There are three element types of memory cells: *read-only cells, write-only cells*, and *two-way cells*, which allow both reading and writing.

4 Construction of Information Processing Systems with Cellular Automata

Let us consider a model of computation **M** that has a universal automaton U.

Theorem 1. A model of computation **M** is programmable in a class **C** of cellular automata if and only if a universal automaton U is programmable in some cellular automaton C from **M**.

Note that here the transcribed equivalence is used because usually universal automata, e.g., universal Turing machines, model other automata from the same class, e.g., other Turing machines, only with transcription [6].

Corollary 1. A model of computation **M** is programmable in a cellular automaton C if the automaton U is programmable in C.

For illustration, we give here a well-known result in the theory of cellular automata.

Theorem 2. The class **T** of all Turing machines is programmable in the class C_1 of one-dimensional cellular automata.

Lemma 1. If a class **A** of automata is programmable in a class **C** of automata and a class **C** of automata is programmable in a class **B** of automata, then the class **A** is programmable in the class **B**.

It is known (cf., for example, [6]) that any class of recursive algorithms, such as partial recursive functions, random access machines (RAM) or Minsky machines, as well as any class of subrecursive algorithms, such as recursive functions, pushdown automata or context free grammars, is programmable in the class **T** of all Turing machines. Thus, Lemma 1 and Theorem 2 give us the following result.

Theorem 3. Any class of recursive algorithms (any class of subrecursive algorithms) is programmable in the class C_1 of one-dimensional cellular automata.

Corollary 2. An arbitrary pushdown automaton is constructible in the class C_2 of two-dimensional cellular automata.

Building a two-dimensional cellular automaton CA from multilevel finite automata [7], it is possible to prove the following result.

Theorem 4. A two-dimensional cellular automaton can realize any finite grid of connections between nodes in a grid automaton G.

To realize all these types of elements in cellular automata, multilevel finite automata described in [7] are used.

Corollary 3. If all nodes in a finite grid automaton G have a finite number of ports and are programmable (constructible) in one-dimensional cellular automata, then the automaton G is programmable (respectively, constructible)

in a two-dimensional cellular automaton. Note that not any finite configuration is a finite automaton. For instance, at each step, a Turing machine is a finite configuration but it's not a finite automaton. Another example is when a node in a grid automaton can be an automaton that works with real numbers.

It is also possible to construct Turing machines in cellular automata.

Theorem 5. An arbitrary Turing machine with a one-dimensional tape is constructible in the class C_1 of one-dimensional cellular automata.

To prove this theorem, finite automata with inner structure are used.

Note that it is not the standard result that one-dimensional cellular automata can emulate a one-dimensional Turing machine. The standard result tells that an arbitrary Turing machine is programmable in the class C_1 of one-dimensional cellular automata. Theorem 5 establishes that an arbitrary Turing machine is constructible in the class C_1 . Constructability implies programmability but the converse is not true. For instance, any Turing machine with a two-dimensional tape is programmable in the class of Turing machines with a one-dimensional tape, but it is not constructible in this class.

As the class **T** has universal Turing machines, Theorems 1 and 5 imply the following result.

Corollary 4. The class **T** of all Turing machines with a one-dimensional tape is constructible in the class C_1 of one-dimensional cellular automata.

Global Turing machines or Internet machines introduced in [17] form a natural class of grid automata. An Internet machine is a finite grid automaton in which all nodes are Turing machines. Theorems 4 and 5 imply the following result.

Corollary 5. An Internet machine *IM* is constructible in the class **CA** of cellular automata. This implies the following result.

Corollary 6. The class **IM** of all Internet machines is constructible in the class **CA** of cellular automata.

Corollary 7. The class **T** of all Turing machines is constructible in the class **CA** of cellular automata.

In a similar way, it is possible to program inductive automata (inductive models of computation), which provide better modeling of contemporary computers and computer networks than traditional models, such as Turing machines [6].

Theorem 6. Any inductive Turing machine of the first order is programmable in the class **ICA** of inductive cellular automata.

Corollary 8. The class $\mathbf{IT}_1$ of all inductive Turing machines of the first order is programmable in the class **ICA** of inductive cellular automata.

Similar to Internet machines, it is useful to introduce inductive Internet machines, which also form a natural class of grid automata. Any Internet machine is a finite grid automaton in which all nodes are inductive Turing machines.

Theorems 4 and 6 imply the following result.

Corollary 9. Any inductive Internet machine *IM* is constructible in the class **ICA** of inductive cellular automata.

Computers and devices in global networks start processing data not only in the form of words, as conventional abstract automata do, but also more sophisticated structures. For instance, researchers forecast that future global networks will use graphs or heaps of soft protocol elements instead of multilayered protocol stacks used now [8, 2]. That is why it is important to represent not only words but other advanced structures using cellular automata. In addition, structures of computer hardware and software are much more sophisticated than linear structures of words.

Here is one result that demonstrates corresponding possibilities of cellular automata in modeling data structures.

Theorem 7. A two-dimensional cellular automaton can realize any finite graph or network.

In a similar way, cellular automata can realize many other data structures.

5 Conclusion

We discussed a new discipline – cellular engineering. Obtained results show how it is possible to construct and model sophisticated complex system using such relatively simple systems as cellular automata. The functional cellular engineering is one of the weakest forms, while the system cellular engineering is one of the strongest forms of cellular engineering.

Indeed, building a system with necessary properties solves the problem of creating a process with necessary features, while the latter solves the problem of constructing a function with necessary characteristics. Usually only functional cellular engineering has been considered, e.g., when cellular automata computed the same function as a Turing machine.

Modeling relation plays an important role in all life processes and living systems [16, 12]. Thus, it would be interesting to use cellular automata for modeling real living systems and not only some processes that resemble functioning of living systems as it is done in Artificial Life [1]. Moreover, in the context of pancomputationalism (cf., for example, [19, 11, 11, 13]) when the universe is treated as a huge computational structure or a network of computational

processes which following fundamental physical laws compute (dynamically develop) its own next state from the current one, the *Technological Principle of Computational Equivalence* can be the base for constructor theory discussed by Deutsch in [10].

It is interesting to know that the method developed in [6] for construction of Turing machines and grid automata in cellular automata gives a formal representation of the old Internet idea [8] that any component with more than one network interface can be a router.

References

[1] Bornhofen, S., Lattaud, C.: Outlines of Artificial Life: A Brief History of Evolutionary Individual Based Models. In: Talbi, E.-G., Liardet, P., Collet, P., Lutton, E., Schoenauer, M. (eds.) EA 2005. LNCS, vol. 3871, pp. 226–237. Springer, Heidelberg (2006)

[2] Braden, R., Faber, T., Handley, M.: From protocol stack to protocol heap: Role-based architecture. ACM SIGCOMM Computer Communication Review 33(1), 17–22 (2003)

[3] Burgin, M.: Information Algebras. Control Systems and Machines (6), 5–16 (1997) (in Russian)

[4] Burgin, M.: Cluster Computers and Grid Automata. In: Proceedings of the ISCA 17th International Conference on Computers and their Applications, Honolulu, Hawaii. International Society for Computers and their Applications, pp. 106–109 (2003)

[5] Burgin, M.: Unified Foundations of Mathematics, Preprint Mathematics LO/0403186, 39 p. (2004), electronic edition: `http://arXiv.org`

[6] Burgin, M.: Superrecursive Algorithms. Springer, New York (2005)

[7] Burgin, M.: Cellular Engineering. Complex Systems 18(1), 103–129 (2008)

[8] Crowcroft, J.: Toward a network architecture that does everything. Comm. ACM 51(1), 74–77 (2008)

[9] Burgin, M.: Measuring Power of Algorithms, Computer Programs, and Information Automata. Nova Science Publishers, New York (2010)

[10] Deutsch, D.: Physics, Philosophy and Quantum Technology. In: Proceedings of the 6th International Conference on Quantum Communication, Measurement and Computing. Rinton Press, Princeton (2003)

[11] Fredkin, E.: Digital Mechanics. Physica D, 254–270 (1990)

[12] Kineman, J.J.: Modeling relations in nature and eco-informatics: A practical application of Rosenian complexity. Chemistry and Biodiversity 4(10), 2436–2457 (2007)

[13] Lloyd, S.: A theory of quantum gravity based on quantum computation. Preprint in Quantum Physics (2006) (arXiv:quant-ph/0501135)

[14] von Neumann, J.: Theory of Self-Reproducing Automata. 1949 University of Illinois Lectures on the Theory and Organization of Complicated Automata, Edited and completed by Arthur W. Burks. University of Illinois Press, Urbana (1966)

[15] Peterson, L.L., Davie, B.S.: Computer Networks: A System Approach. Morgan Kaufmann Publishers, San Francisco (2000)

[16] Rosen, R.: Life itself: A Comprehensive Inquiry into the Nature, Origin and Fabrication of Life. Columbia University Press, New York (1991)
[17] Van Leeuwen, J., Wiedermann, J.: Breaking the Turing Barrier: The case of the Internet. Techn. Report, Inst. of Computer Science, Academy of Sciences of the Czech. Rep., Prague (2000)
[18] Wolfram, S.: A New Kind of Science. Wolfram Media, Champaign (2002)
[19] Zuse, K.: Rechnender Raum. Friedrich Vieweg & Sohn, Braunschweig (1969)

Part IV

Fundamental Physics

Chapter 11
The Principle of a Finite Density of Information

Pablo Arrighi[1] and Gilles Dowek[2]

[1] École normale supérieure de Lyon, and
Université de Grenoble, France
parrighi@imag.fr
[2] Institut national de recherche en informatique et en automatique (INRIA), France
gilles.dowek@inria.fr

Abstract. The possibility to describe the laws of the Universe in a computational way seems to be correlated to a principle that the density of information is bounded. This principle, that is dual to that of a finite velocity of information, has already been investigated in Physics, and is correlated to the old idea that there is no way to know a magnitude with an infinite precision. It takes different forms in classical Physics and in quantum Physics.

1 Why Cellular Automata?

Stephen Wolfram has advocated in [10] the idea that cellular automata might be a relevant formalism to study the laws of the Universe. This thesis can be seen as a consequence of three more fundamental principles that are implicit in the definition of the notion of a cellular automaton.

To define a cellular automaton, we must discretize space by partitioning it into an infinite set of identical *cells*, for instance cubes. We must also discretize time and observe the Universe at regular time steps. Then come three assumptions

1. that the state space of each cell is finite,
2. that the state space of a cell at a given time step depends only on the state of a finite number of neighbours at the previous time step,
3. that the state space is the same for all cells and the evolution acts the same everywhere and everywhen.

Assumption (3.) is a reformulation of a well-known principle of Physics: the homogeneity of space and time. So is assumption (2.), which is a reformulation of the bounded velocity of information. Assumption (1.), in contrast, seems to express the new idea, that the density of information also is bounded. Such a principle can be formulated as the fact that the amount of information that can be stored in a bounded region of space is bounded. It can also be formulated as the fact that the cardinality of the state space of a bounded region of space is bounded, since the amount of information storage is the logarithm of this cardinality.

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 127–134.
DOI: 10.1007/978-3-642-35482-3_11

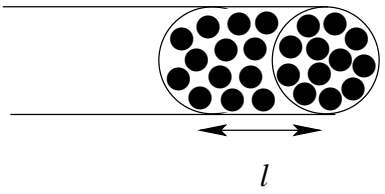
There are other assumptions in the definition of a cellular automaton, such as the fact that space-time is absolute and passive: the neighbourhood relation on cells does not evolve as their states do. Weakening these assumptions is possible, leading for instance to *causal networks* [10] or to *causal graph dynamics* [2].

There is a historical discrepancy between the idea of a finite velocity of information and that of a finite density of information. The first has had a clear status as a principle of Physics, since special relativity, and we even know the bound on the velocity. The second seems to be less established. Yet, it is not completely new, as all three principles (1.), (2.), and (3.), have been stated by Robin Gandy in [7], and the principle of a bounded density of information has also been stated by Jacob Bekenstein—although the Bekenstein bound is not on the amount of information but on the quotient of the amount of information and the energy contained in a sphere [5].

2 Two Dual Principles

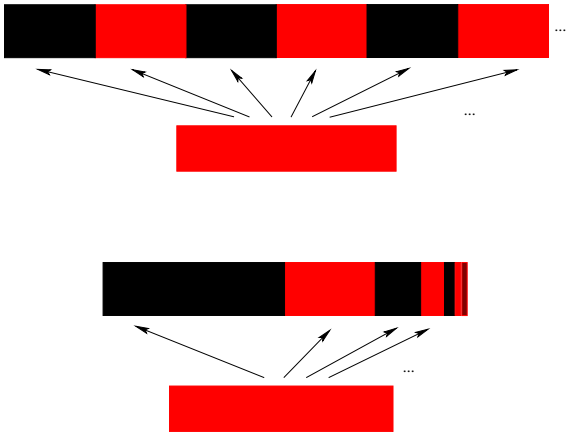
The principles of a bounded velocity and of a bounded density of information play a dual role.

For instance, the amount I of information that can be output by a wire of section S in a given time t is bounded by the amount of information contained in the cylinder of section S and length l, where l is the distance that information can travel in time t.



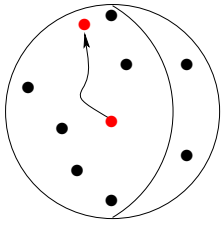
If information has a bounded velocity, then l is bounded and if moreover information has a finite density, the amount of information I is bounded. Thus we need both principles to prove that the information flow of a wire is bounded. If one of these principles were invalid, it would be easy to build a wire with unbounded flow.

A more evolved example is the impossibility to build a infinitely parallel computing machine, i.e. a machine formed with an infinite number of processors communicating with a common server. If information has a bounded density, then the size of each machine has a lower bound and some machines must be arbitrarily far from the server and if moreover information has a finite velocity, some machines are too far to communicate with the server in due time. Again, if one of these principles were invalidated it would be easy to build an infinitely parallel machine, either with processors of the same size communicating infinitely fast with the server, or with arbitrarily small machines, of size 1, 1/2, 1/4, 1/8, 1/16, ... that would lie at a finite distance of the server.



It seems that these two principles could even be summarized in one, as the fact that in a given time, a piece of information can only travel in a space which is itself populated by a finite amount of information.

Finally, the two principles are comparable, and both differ from the third: homogeneity of space and time. Indeed, the latter is pre-supposed by the very notion of a 'Physics law'. For instance, suppose that some law would be valid in Paris but not in Boston, then this would not be much of a law. The word 'law' itself carries a notion of universality.



But what about the other two? It turns out that a it is possible to do very respectable Physics without these principles. Yet, it is still the case a notion of applicability of a Physics laws underpins both the finite velocity of information and the bounded density of information principles. If a law broke the finite velocity of information principle, this would entail that, in order to determine the new state of a physical system after one second has past, one may potentially need to look at an unbounded number of neighbours. This is no doubt unpractical, unless far-away contributors can be neglected.

In the same way, if a law broke the finite density of information principle, in order to determine the new state of a physical system, one would need to use a description of the state of the neighbours that might potentially contain an unbounded amount of information. Again this law would hardly be applicable, unless this information can be rounded up in some way.

3 Newtonian Physics

It is well-know that Newtonian Physics contradicts the principle of a finite velocity of information. Whenever a bee flies half a centimeter, it instantaneouly modifies the gravitational field on a distant galaxy. It is only with General Relativity that we understood how to control the consequences of the flight of a bee.

Newtonian Physics not only contradicts the finite velocity principle, it also contradicts the principle of a finite density of information as the distance between the jaws of a caliper for instance is measured by a real number. So if we imagine an infinite amount of information expressed, for instance, as an infinite sequence of 0 and 1, we can see this infinite sequence as the digits of a real number and place the jaws of a caliper at this exact distance, recording this way the infinite amount of information in a finite volume.

Of course, we have all been taught that this idea is nonsense and that the distance between the jaws of a caliper is defined with a finite precision: three or four digits. It is surprising that this thesis that a physical magnitude is always defined with a finite precision has never been given the status of a Physics principle as have the homogeneity of space and time. Yet, this thesis may be a very early occurrence of this idea of a finite density of information.

4 The Finite-Density versus the Superposition Principle

The principle of a finite density of information seems to be challenged by quantum theory and its superposition principle. Indeed, regardless of whether a region of space is bounded or not, whenever its state space contains two states $\mathbf{u}$ and $\mathbf{v}$, then it must also contain all the linear combinations $\lambda\mathbf{u} + \mu\mathbf{v}$ up to a renormalization factor. Hence it is infinite, and the finite density principle cannot be formulated as the fact that the set of states of a bounded region of space is finite.

These complex amplitudes λ and μ can be compared with probabilities. But this comparison has its limits. Indeed, since probabilities are positive real numbers, they can only add up constructively to make and event more likely. In contrast amplitudes are complex numbers, then can also be substracted to one another, i.e. add up destructively, making an event either more likely of less likely; or sometimes even impossible.

This is why when two different scenarios happen in a superposition, we cannot quite think of them as two non-interacting branches of what may happen in the Universe: these amplitudes cannot be ignored. This makes the superposition principle is a difficulty when extending the finite-density principle to quantum theory.

Yet, the infinity of the state space does not mean that the amount of possible outcomes, when measuring the state of this region, is itself infinite, because in quantum theory, we cannot measure the state of the system, but only one of its observables. Thus, an alternative formulation of the bounded density of information principle is that each projective measurement of a bounded region,

at any given point in time, may only yield a finite number of possible outcomes. This requirement amounts to the fact that the state space of a bounded region of space is a finite-dimensional vector space. This constitutes a good quantum alternative to the formulation of the finite density of information principle, one which does not demand anything to be actually measured in any way.

5 The Finite-Density Principle versus Correlations

Moreover, yet another problem arises in the quantum setting, mamely that of 'correlations'; which in this context are also referred to as 'entanglement'. To understand what entanglement is, one must apply the superposition principle again, but this time to pairs of systems. For instance if systems A and B may be in state $\mathbf{uu}$, meaning that both are in state $\mathbf{u}$, or in state $\mathbf{vv}$, meaning that both are in state $\mathbf{v}$, then the state $\lambda(\mathbf{uu}) + \mu(\mathbf{vv})$ is also allowed. This entangled state corresponds to a superposition of two global, and everywhere different scenarios. It is possible to get an intuition of the meaning of this entangled state by appealing to our understanding about usual probabilities. In probability theory, it is a commonplace that knowing the marginal probabilities over the individual systems does not entail knowing the joint probability distributions. For instance Alice and Bob may have a half-half chance of having a blue or red ball in each of their boxes, but it could be that each of their boxes have been prepared in such a way that both balls are of the same colour. The state $\lambda(\mathbf{uu}) + \mu(\mathbf{vv})$ tells the same story not for probabilities, but for amplitudes.

This is an issue, because the bounded velocity of information principle seems very weak in that respect: after all, all it says is that the state of each individual system is determined by that of the neighbours, but it does not say how the entanglement between the individual states is determined. If we retake the probability analogy; the bounded velocity of information principle tells you that the probabilities of Alice finding a red or a blue ball in her box are determined locally; but what about the probability that this ball is of the same color as Bob's?

Fortunately on this issue of whether the correlations can be determined locally, quantum theory is a bit friendlier than probability theory. Indeed, it has been shown that whereas the bounded velocity of information principle is way too weak in the context of probabilistic cellular automata [3], surprisingly it turns out to be sufficient in the context of quantum cellular automata [4] for the evolution to be described as a composition of local functions mapping neighbourhoods to neighbourhoods, that is functions from a finite-dimensional vector space to itself. Another way to phrase this is that in the quantum setting, any evolution that respects the bounded velocity of information principle necessarily takes the form of a circuit of gates being applied on systems and their neighbours. Thus correlations can be 'tamed' and the global evolution broken into functions from finite-dimensional vector spaces to themselves.

6 A Computable Quantum Universe

Yet, moving from a finite set to a vector space, even a finite-dimensional one, is a big leap and many advantages of describing the Universe as a cellular automaton might be lost by doing so. Indeed, the main advantage of having a finite state space S for cells was that the local evolution function of the cellular automaton, that is a function from S^n to S, where n is the number of neighbours of a cell, was a function from a finite set to a finite set. Such a function can be described by a table giving the image of each element of the domain and it is always computable. This point was the key to the computability of cellular automata, and thus to the idea that the Universe is computable if it can be described as a cellular automaton. This idea of a computable Universe—the physical Church-Turing thesis—is one of the main goals of both Robin Gandy and Stephen Wolfram. Fortunately, another feature of the quantum theory comes to repair the damages created by the superposition principle. According to the quantum theory, theses local functions of a quantum cellular automata cannot be any function from neighbourhoods to neighbourhoods, but must be linear. There are much less linear functions than functions from a finite-dimensional vector space to another. In particular any such linear function can be described by a matrix, and matrix multiplication boils down to additions and multiplications that are computable operations. Thus if scalars are chosen in an adequate manner as discussed below, all such functions are computable. Thus, infinity is tamed by finite-dimension and linearity and this formulation of the principle of a finite density of information is sufficient to prove the Universe computable.

7 Scalars

When saying that addition and multiplication of scalars are computable, we need to be a little careful, as scalars are complex numbers and computability is mostly defined for natural numbers.

Computability can be extended from natural numbers to real and complex numbers [9], but then the picture is a somewhat different. Not only functions from complex numbers to complex numbers can be computable or not, but complex numbers themselves can be computable or not. Thus, although multiplication is computable, multiplication by a fixed scalar may be non-computable if the fixed scalar is non-computable.

Michael Nielsen has noticed that having a process building a superposed state $\lambda\mathbf{u} + \mu\mathbf{v}$, where λ and μ are non computable numbers, can *per se* lead to non computability, as this state can be tomographed by repeated measurements and λ be extracted with increased precision from the state.

Thus, a physical system has two ways to encode an infinite amount of information: either by having an infinite dimensional state space, or by encoding an infinite amount of information in a scalar, exactly in the same way as an infinite amount of information can be encoded in a distance in Newtonian Physics.

To express the finite-density principle, it seems that it not sufficient to restrict to finite-dimensional vector spaces for the state spaces of cells, but we must also restrict to a smaller set of scalars, yielding an intersting problem for physicists about the fields that can be used for the scalars in quantum theory.

With this precise formulation of the finite density of information, we can formally prove that the physical Church-Turing thesis is a consequence of the three hypotheses we have mentioned: homogeneity of space and time, bounded velocity of information and bounded density of information [1].

Some scientists, such as David Deutsch [6] propose to take the physical Church-Turing thesis as a principle of Physics. Some others, such as Michael Nielsen [8], propose to deduce this thesis from more fundamental principles of Physics. Like Robin Gandy, we have chosen to start from these three fundamental principles. An advantage of this approach is that these three principles are falsifiable provided explicit bounds are given. Thus, from a logical point of view, they are purely universal formulas, while the physical Church-Turing thesis itself has a much more complex logical form $\forall\exists\forall$, namely for all physical system m, there exists a computable function f such that forall initial state x, the evolution of m leads from x to $f(x)$.

8 Conclusion: Cellular Automata and Quantum Cellular Automata

It is one thing to say that the Universe is computable and thus can be simulated by any model of computation, e.g Cellular Automata. It is another to seek to closely model physical systems in a space-preserving manner. Pursuing this second, more demanding aim leads us to investigate how to model the state space of a bounded region of space. Supposing that this state space is finite is adequate in the classical setting, but must be refined in the quantum setting to a finite-dimensional vector space over a reasonable field. Indeed it seems that simulating a quantum systems by classical cellular automata not only leads to an exponential slowdown, but also fails to respect the geometry of the system. This justifies Feynman's proposal to model quantum systems with Quantum Cellular Automata. This quantum extension of the Cellular Automata model shows its liveliness and robustness.

References

[1] Arrighi, P., Dowek, G.: The physical Church-Turing thesis and the principles of quantum theory. Int. J. Found. of Computer Science (2011) (to appear)
[2] Arrighi, P., Dowek, G.: Causal graph dynamics. Pre-print arXiv:1202.1098 (2012)
[3] Arrighi, P., Fargetton, R., Nesme, V., Thierry, E.: Applying Causality Principles to the Axiomatization of Probabilistic Cellular Automata. In: Löwe, B., Normann, D., Soskov, I., Soskova, A. (eds.) CiE 2011. LNCS, vol. 6735, pp. 1–10. Springer, Heidelberg (2011)
[4] Arrighi, P., Nesme, V., Werner, R.: Unitarity plus causality implies localizability (full version). Journal of Computer and System Sciences 77(2), 372–378 (2011)

[5] Bekenstein, J.D.: Universal upper bound to entropy-to-energy ratio for bounded systems. Phys. Rev. D 23, 287–298 (1981)
[6] Deutsch, D.: Quantum theory, the Church-Turing principle and the universal quantum computer. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences (1934-1990) 400(1818), 97–117 (1985)
[7] Gandy, R.: Church's thesis and principles for mechanisms. In: The Kleene Symposium. North-Holland Publishing Company, Amsterdam (1980)
[8] Nielsen, M.A.: Computable functions, quantum measurements, and quantum dynamics. Phys. Rev. Lett. 79(15), 2915–2918 (1997)
[9] Weihrauch, K.: Computable analysis: an introduction. Springer (2000)
[10] Wolfram, S.: A new kind of science. Wolfram Media Inc. (2002)

Chapter 12
Do Particles Evolve?

Tommaso Bolognesi

CNR/ISTI, National Research Council, Pisa, Italy

Abstract. After some reflection on the messages that I have found most inspiring in Wolfram's NKS book, ten years after its publication, in this paper I speculate on a few, highly attractive new developments that NKS-style experimental research might undergo, and that I have myself begun to explore in recent years. According to these visions, the grand challenge that the emergent, localized structures of elementary automaton 110, or similar 'particles', must face in the next ten years is to evolve into populations of increasingly complex individuals, up to forms of (artificial) life, and to a fully blown biosphere.

On a more technical side, the paper illustrates some preliminary steps and results in the application of Genetic Algorithms to variants of Wolfram's Network Mobile Automata; the objective here is to investigate the emergent qualitative and quantitative properties of the causal sets associated to the automata computations, in view of their potential application as discrete models of physical spacetime.

1 Introduction

Ten years have passed since the interacting 'artificial particles' of elementary cellular automaton n. 110 have started flickering from the cover of the NKS book, sending puzzling signals to the scientific community and the general public, and raising controversy as to the role that this type of phenomena should play between recreational mathematics and theoretical physics.

What are the most inspiring contributions from the 1197 pages of the NKS volume? Starting in 2005, I have met several readers of the book, also due to my involvement with various editions of the NKS Summer School. Many of them would point to a specific chapter, but a surprisingly high number of readers would go as far as providing exact page numbers for their favorite passages or pictures. However, if I am to single out the thing that I found most stimulating and influential for my work, I would indicate the very spirit and investigation attitude that pervades all of its components.

This attitude has been sometimes compared to that of an entomologist, who patiently and carefully observes and classifies the variety of shapes and phenomena occurring in a specific corner of the universe; the difference is that the universe explored by NKS is the computational one, and the studied insects are the free, spontaneous computations of simple programs. Observing, naming, classifying things in the universe are the primordial cognitive activities of homo

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 135–155.
DOI: 10.1007/978-3-642-35482-3_12

sapiens: NKS transposes these attitudes, most directly, to the study of another universe, based on the bold conjecture that the two universes might coincide.

And one feeling that comes with these activities is pleasure. The impressive array of plots for the countless computational experiments carried out in [25] is indeed permeated by the pleasure of discovery. This is tangible, for example, in the series of plots for Elementary Cellular Automaton (ECA) 30, at pages 27-30, or in the longer series at pages 32-38, for ECA 110, that manage to convey the excitement of progressively disclosing the pseudo-randomness and the interacting localized particles of the two most celebrated members of the ECA family.

Exhaustive enumeration, simulation and visual inspection of the space of spontaneous computations, with a clear mind, free of expectations, are fundamental experimental activities in the NKS approach, and their primary objective is just *to see what happens*; this is best represented by the ECA tables at pages 54-56, and is nicely reflected in a reported dialogue between Richard Feynman and Stephen Wolfram:

- 'How did you know in advance that, out of 256 automata, n. 30 would behave pseudo-randomly'?

- 'I didn't; I ran them all'!

One experiences a progression of excitement levels, in moving from the predictability of class 1 behaviors to the surprises found in class 4. Needless to say, precious gems are very rare, and this makes their discovery particularly gratifying. Having myself compiled hundreds of tables analogous to those just mentioned, for various other models of computation, I reproduce in Figure 1, without providing details (see [2]), one that well reflects the amount of inspiration that I have derived from NKS, and the excitement that comes from spotting a gem (e.g. the plot for computation (17, 8)) among pebbles.

The most ambitious among the NKS experimental activities and goals is sometimes called 'universe hunting': finding the ultimate, digital code of nature. The barrier of computational irreducibility makes it impossible to predict the steps of an evolving, digital universe, other than in real time, i.e., as it evolves. Hence, any computational theory of the physical universe would fail to *efficiently* predict the evolution of the universe, as a *whole*, and in *full detail*; but it could always be used for simulating its infancy, in search for emergent features and entities that might have played some role in building up our mature, 13-billion year universe, or that might have even reached us, unchanged, across this huge time span.

So, what kind of gems might we expect to see emerge from universe hunting experiments in the next ten years? Although the NKS lesson is to carry out computational Big-Bang experiments without predefined expectations, in Section 2 I provide a wish list of phenomena whose emergence would represent a strong indication of the validity of the approach.

More concretely, Section 3 illustrates an application of Genetic Algorithms to Network Mobile Automata, meant to bias the evolution of these automata and assess their potential applicability to the discrete modeling of spacetime.

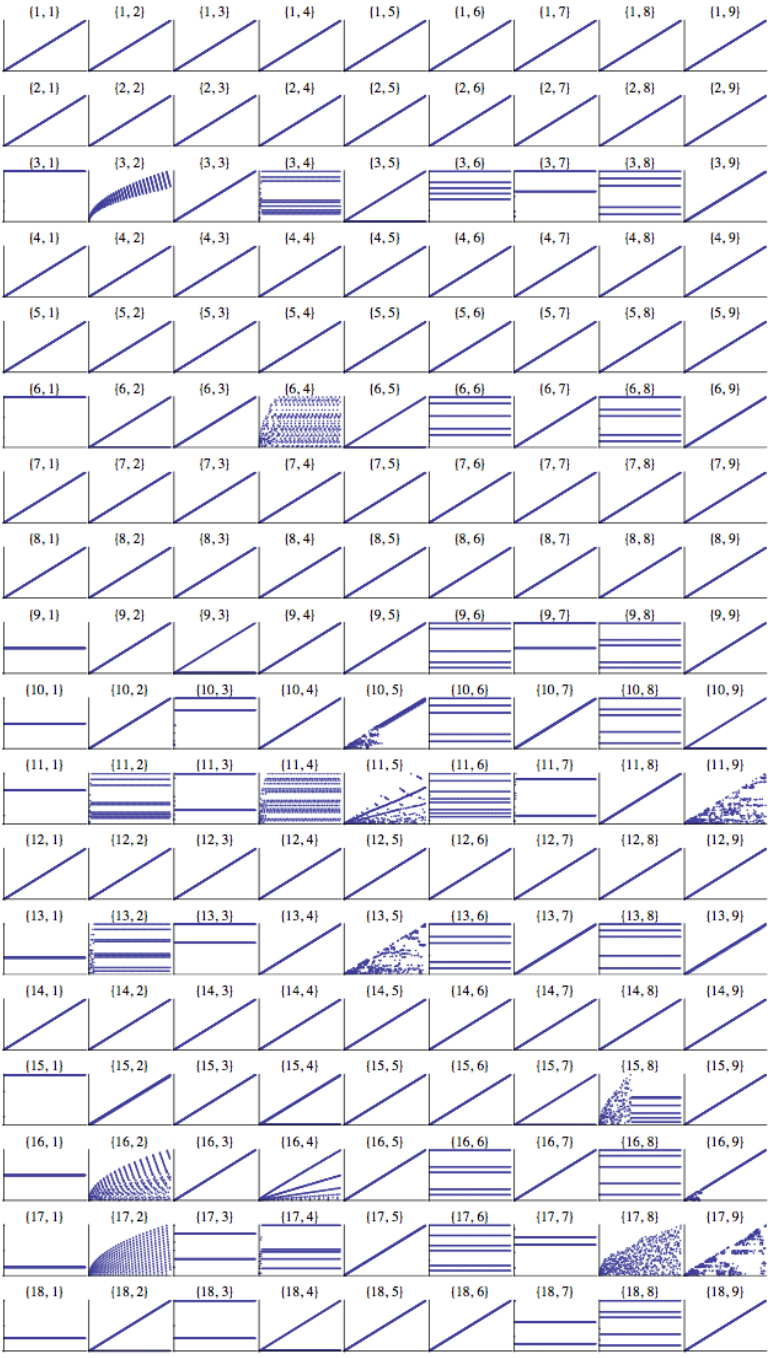


Fig. 1. Plots showing the dynamics of an ant moving on a planar trivalent graph according to the algorithm described in [2]. The ant behavior depends on two parameters, ranging, respectively, in 1-18 (rows) and 1-9 (columns).

2 Particle Evolution

The interest for the localized structures of ECA 110, in the context of a computation-oriented, fundamental theory of physics, depends on two facts: (i) they are strongly reminiscent of scattering phenomena and diagrams from particle physics; (ii) they can be used for performing Turing-complete computations. It seems therefore natural to consider these particles as an encouraging clue for a theory that tries to relate complexity in physics with emergence in computation, and to take them as a starting point for developing such a theory.

More precisely, if we accept the idea of a physical universe organized as a *layered architecture of emergence*, fueled by a simple computation at the bottom [7], we may wonder whether further layers of increased complexity can be found on top of some layer of simple (artificial) particles. Note that, when referring to 'particles', I am not restricting to the localized structures of cellular automata. Similar phenomena are also observed with other models (two-dimensional Turing machines, or *turmites*, are just one example [20, 21]), or in the causal sets derived from their computations [6]. In particular, the localized structures observed in a causal set can be interpreted as trajectories across spacetime, or particle worldlines.

In the sequel I speculate on a number of possible evolved forms of artificial particle; their appearance might contribute to the creation of an artificial universe of growing complexity, as close as possible (if not identical!) to our universe.

2.1 Soft vs. Hard Particles

Consider one of the trajectories that traverse an ECA 110 diagram. It is formed by a periodic pattern of black and white cells that moves in time (vertically), and often in space (horizontally). When it translates in space, the localized pattern repeats every few rows, being formed, each time, by a *different* set of a few adjacent cells: the pattern moves, the cells don't.

But in other models of computation, the elements of the memory support may themselves move around. For example, in the Trinet Mobile Automata to be discussed in the next section, the support is a dynamic planar graph whose polygonal faces are modified during the computation. A typical face update consists in increasing or decreasing by one unit the number of sides, and in this game, faces that are adjacent at some moment, drift apart later on. Think of a face as corresponding to an ECA cell, and of the face size (number of sides) as corresponding to the cell state (a bit). Then, in a computation, formed by a sequence of graph configurations, in principle we may observe *two* types of particle, which, in the absence of better names, I call 'soft' and 'hard':

Soft particles equivalent to the ECA particles, are persistent periodic patterns that move around, but are formed by ever changing underlying faces;

Hard particles are groups of faces that persistently stick together and move around while undergoing cyclic shape changes.

For visualizing the idea of dynamic, planar trivalent graphs, and of a face that moves around while changing its shape, it may be useful to consider Voronoi diagrams. By using the interactive demonstration in [22], the interested reader can actually create, destroy, or carry around, under the mouse cursor, a face of a random Voronoi diagram, while observing that all these operations correspond to the application of simple graph-rewrite rules (Pachner moves).

In [4, 6] I report about the emergence of some periodic trajectories in the causets of various models, but without relating these structures with the elements of the underlying memory support, which is the key for discriminating between the two types of particle. The search for soft particles, in particular, is computationally quite costly, but being able to detect them might reveal complex interaction scenarios that, so far, have only been observed with the (soft) particles of Cellular Automata.

2.2 Stateful Particles

The outcome of the interactions between two ECA 110 particles depends on the precise phase by which they collide, as illustrated in [25] (pp. 294-295) – a circumstance that contributes to the richness of interaction scenarios but makes it particularly hard to control these structures for specific computational purposes. Complex interactions could also be observed among particles whose behavior depends on some form of internal state: the idea is to move from stateless particles to stateful agents.

Systems of stateful entities that interact by various mechanisms – by sharing memory locations or by exchanging messages via an ether or via dedicated channels – are pervasive in computer science. Examples include Actors [12, 1], and process calculi such as CCS [17] and CSP [13]. Although these models have been primarily conceived for providing formal foundations to concurrent programming or specification languages, they identify the general features of concurrency, (massive) parallelism, interaction and communication, at such a high abstraction level that they become applicable also to natural systems in which those features are of relevance.

The gap between the relatively simple artificial, computational particles observed so far, and the concept of stateful agent, is huge. However, in a theory that conceives physical objects and their background as made of the same fabric, in which both are obtained from the dynamic geometry of spacetime, a useful condition for the emergence of advanced, stateful entities would be the appearance of 'shells' that encapsulate small regions of that fabric, separating a small inside from a large outside, an active object from a (relatively) passive background. Similar to a selectively permeable cell membrane, this closed surface would favor the persistence of the localized structure, promote some internal state stability, and regulate interactions with the environment.[1]

[1] Highly persistent structures can be observed in 2-D cellular automata, e.g. the 'iceballs' of Ice Nine: http://www.collidoscope.com/modernca/icenine.html

In [6] I identify a first rudimentary form of this 'causet compartmentation' phenomenon, and I argue that it can only occur with deterministic, algorithmic (as opposed to stochastic) causal sets.

2.3 Self Modifying Code

In a review of the NKS book, appearing in the Web, Ray Kurzweil writes:

> "Ultimately, the patterns are not that complex. They're all kind of similar and too low-level to explain the evolution of higher levels of intelligence."

After much experimental evidence from [25], we accept the idea that the iteration of simple rules can originate complex and unpredictable phenomena, but could these include a living cell? Several layers of emergence should be required for that; can they all originate from a *fixed*, initial set of rules? After about five years of experience with various formal systems, in particular with graph rewriting, I am becoming increasingly convinced that a big step ahead could be made if we allowed the rules themselves to take part into the evolutionary process, rather than having them fixed in advance.

Indeed, most scientists would probably look more favourably at a scenario in which physical laws emerge as the universe unfolds from the Big Bang, rather than one that attributes an absolute, transcendental status to those laws, placing them in Plato's hyperuranium. At least, this is the position expressed by eminent physicist John A. Wheeler, with his famous concept of 'Law Without Law', and with the idea that

> "Proud unbending immutability is a mistaken ideal for physics; this science now shares, and must forever share, the more modest mutability of its sister sciences, biology and geology".

The idea of a self-modifying program is, again, quite familiar in computer science. In logic programming, for example, a program can modify itself by absorbing new facts and rules during execution, realizing, in a way, a process of learning from experience. Note, however, that the execution of a self-modifying program is supported by software (compiler, interpreter, operating system) and hardware that are not self-modifying at all: this induces us to take a closer look at the idea that the program fueling the universe from the bottom (if any) be a self-modifying piece of code.

For fixing ideas, let that program be expressed as a Turing machine Y. We actually don't know what it formally means for Y to be self-modifying, since a Turing machine can only access its tape, not its own state transition table. On the other hand, we can simulate Y on some standard (not self-modifying!) universal Turing machine X, and do more: we can program X so that it progressively changes the behavior (the state transition rules) of Y as the simulated computation proceeds. In this case, though, the role of animating the universe would be ultimately attributed to X – a fixed behavior – not to Y!

In conclusion, even if novel and increasingly complex particle types were to emerge from Turing machines with dynamically changing behavior, such interesting cases could as well be found by exploring the space of 'spontaneous' computations of standard, *not* self-modifying, universal Turing machines.

Does this make the idea of self-modifying rules useless? Yes, in principle; but not in practice. The chances that a randomly chosen machine X actually implemented the scenario of a simulated machine Y with dynamically changing rules are prohibitively small. Thus, rather than waiting for cosmological times for this to happen, we could more pragmatically set up NKS-style experiments in which we *explicitly* implement various policies for changing the rules of the game, as the game unfolds.

2.4 Ant Multiplication

At p. 486 of [25], Wolfram writes:

> "At first it may seem bizarre, but one possibility that I believe is ultimately not too far from correct is that the universe might work not like a cellular automaton in which all cells get updated at once, but instead like a mobile automaton or Turing machine, in which just a single cell gets updated at each step".

Since in Turing machines and Mobile Automata the computation is carried out by a control unit that walks, like a little ant, on a memory support – typically a binary tape – we may refer to the bold idea above as the 'Touring Ant Universe (TAU) conjecture'. Most of the computational experiments described in [2, 3, 4, 6] refer to a form of Network Mobile Automata based on a stateless ant (to be discussed later), and are ultimately meant to investigate the TAU conjecture.

The idea of a Cosmos run by a memoryless ant is quite appealing; if true, if would represent a stupendously simple theory of everything. But, again, can we expect the whole layered architecture of our universe to emerge from the tireless work of a *single* ant, or should we rather think of multiple agents, with the expectation to obtain additional complexity from their collective operation?

With experiments on Generalized Mobile Automata [25], a model in which ants are allowed to reproduce, Wolfram has found that complex behavior is *more likely* to emerge when several such agents are simultaneously active.

On more theoretical grounds, Carl Hewitt shows that the paradigm of multiple, concurrent agents extends the notion of computation beyond Turing machines, proving that there are effective computations that can be performed by the already mentioned Actor model [12, 1], and not by Turing machines: Actors can implement *unbounded nondeterminism*, Turing machines can't [11].

There are therefore good reasons for exploring 'multiple-ant' systems, in the context of the computing universe conjecture. Indeed, this makes sense regardless of whether we assume a single agent (a TAU model) or a collection of agents (inspired, say, by the Actor model) as the ultimate, primitive mechanism operating at the root. In the first case, the rationale behind the experiments would

be to skip the first phases of evolution, and to assume that they might lead to the hypothetical, multiple-agent intermediate layer at which we decide to start the simulation.

2.5 Collectively Autocatalytic Systems of Particles

In [15] Kauffman elaborates the hypothesis that darwinian natural selection may not be the only source of order in the biosphere. The other source might be self-organization: life might derive from the spontaneous emergence of *collectively autocatalytic systems of molecules*. In essence, once a sufficiently large variety of molecules is allowed to interact and react in a primordial soup, it is almost inevitable that a subset of them 'take power': these molecules are at the same time the product and the catalysts of their reactions, and the group achieves a stationary regime representing a primitive form of self-sustaining metabolism. The beauty of this theory is in the simplicity of the underlying mechanism: the 'order for free' in the molecule soup is directly related to the well known phase transition that leads, in random graphs, to the formation of a giant connected component, as studied by Erdös.

We see no reason why this simple, abstract mechanism from random graph theory should not play a role also for a sufficiently variegated population of (soft or hard, stateless or stateful) particles, once these exhibit an ability to interact, combine, split, and perhaps even promote such events, as enzymes do.

3 Evolving Ants

The most ambitious form of computational universe conjecture suggests that a simple program, starting from simple initial conditions, is sufficient for originating a universe so complex as to host, for example, human intelligence. The huge complexity gap between the initial seed and the final result becomes much more tractable if we assume the already mentioned layered architecture of emergence, in which the emergent objects and behaviors of each layer, characterized by their own laws, provide the primitives for the emergence of the objects, behaviors and associated laws at the upper layer.

Note that the layered architecture concept can be understood to refer both to a sequence of actual evolutionary steps in the history of the physical universe, and to the way scientific theories refine one another. Regardless of which interpretation we choose (and they are not mutually exclusive), one way to avoid or mitigate the difficulties involved with 'pure' universe hunting – that means jumping directly to the bottom – is to *start from some intermediate level of evolution*: one imagines and explicitly introduces ad-hoc features in the investigated models, trying to anticipate what might have emerged from yet unknown lower layers, and finds out about the emergent phenomena that such features might trigger in the upper layers. Most of the visions about particle evolution presented in the previous section fit into this picture: we could manage to actually obtain them as the final outcome of a simulated evolutionary path, starting from more

elementary particles, or we could take them as initial, hypothetical conditions, and explore their consequences.

An effective way to speed-up the emergence of some specific property and to identify interesting intermediate levels of evolution, is offered by Genetic Algorithms. In my most recent experiments I have indeed started using them for identifying *Trinet Mobile Automata* whose computations provide *causets* (spacetime instances) of maximized *dimensionality*. Let us quickly survey the introduced concepts.

3.1 A 'Touring Ant' Model: Trinet Mobile Automata (TMA)

In *Trinet Mobile Automata* (TMA), a stateless control head, or *ant*, moves on a *trinet*, which is an undirected trivalent graph – one in which each node has three incident, undirected edges. The ant moves by short steps, each time applying some graph-rewrite rule before moving to the next location. I adopt well known graph rewrite rules, namely the 2-D Pachner moves sometimes called *Expand*, *Contract* and *Exchange*, that find application also in Loop Quantum Gravity (see, e.g., [16]). Usually the initial condition is a trinet consisting of two nodes connected by three parallel arcs, reminiscent of the ϕ greek letter; this is the smallest possible three-connected graph.[2]

In one TMA variant, qualified as *threshold-based* [2], the ant behavior is fully determined by the parameter triple (*threshold*, *expandCode*, *exchangeCode*). The ant is initially positioned on an edge of the ϕ-graph; it applies rule *Exchange* whenever this does not create trinet faces smaller than a fixed *threshold* (for this purpose, it is indeed sufficient to monitor the two faces sharing the current edge); otherwise it applies rule *Expand*. Then, depending on the applied rule, the ant moves to a next location, which is defined, relative to its current position, by parameters *expandCode* and *exchangeCode*, respectively (see [2] for details).

The Turing machine is another touring ant model, except that the ant moves on a tape and has an internal state.

3.2 Causal Sets ('Causets') and Their Dimensionality

The Causal Set Program is an approach to quantum gravity suggesting that, at the smallest scales, spacetime is best described in terms of the simple, discrete mathematical structure of causal sets [9, 23, 24, 10].

A *causal set* (or '*causet*') can be conveniently represented by a directed graph with no cycles, in which nodes represent spacetime events, the *number* of nodes in a subgraph defines the volume of a corresponding spacetime region, and the *order* relation defines the causal dependencies among events (in the continuum, these relations are usually described in terms of lightcones).

In the Causal Set Program, various *probabilistic* techniques have been investigated for building causets, e.g. *sprinkling* [24], or *transitive percolation* [23].

[2] A connected graph is n-connected when n is the minimum number of edges one has to remove in order to disconnect it.

As an alternative, in our recent work we have started exploring *deterministic* techniques, based on the idea that more interesting qualitative and quantitative properties should emerge with *algorithmic* causets [4, 6].

A property of great physical relevance for a causet is its *dimensionality*. Various techniques are available for estimating the dimensionality of a graph [19], and often their outcomes do not agree. *Node-shell-growth analysis*, used by Wolfram in [25], is based on detecting the growth rate of the node shells at progressive distance r from a fixed node: a growth rate of the order of r^k, with $k \geq 1$ and possibly noninteger, reflects a dimension $k+1$. In [8] we have in particular compared the Myrheim-Meyer dimension estimation technique [24], commonly used in the Causal Set Program, with the node-shell-growth technique, pointing out their differences in terms of outcome and applicability.

3.3 Deriving Causets from TMA Computations

The objective of my recent work [5, 4, 6, 8] has been to investigate the emergent qualitative and quantitative properties of the causets associated with the computations of various models, including TMA, in light of their potential applicability to the modeling of physical spacetime. The basic idea for deriving a causal set from a generic sequential computation is simple: each computation step becomes an event (node) in the causet, and a causal link is created from event e_i to event e_j whenever the latter reads a global state component that was written by the former. In concurrent models such as Actors [12, 1], no notion of global state is available, but partial orders of events are even more easily obtained, being essentially built into the model.

3.4 Genetic Algorithms (GA) for TMA

The GA approach, first proposed by John Holland in [14], is meant to identify programs that are fittest for achieving some predefined goal. The approach is quite directly modeled on the darwinian mechanisms of natural selection and evolution, and can be very effective in solving problems that are hard to address with traditional engineering approaches. An example of GA application in the context of cellular automata, called 'majority classification', is provided in [18]; the uninitiated reader can find in it a convenient, quick introduction to the technique. Below we identify the key elements of the GA approach, while instantiating them to our case of interest – that of obtaining high dimensional causets from TMA computations.

Individual an instance of a program, whose behavior can be fully coded by a *genome*. In our application, an individual is an instance of a Trinet Mobile Automaton.

Population, or generation a collection of individuals. We have considered populations of 100 mobile automata each.

Situation a configuration of the global state of the program/individual, or the part of it which is relevant for determining the next computation step, i.e.

the *reaction*. Much freedom is available, and much creativity possible in conceiving TMA situations. In the above mentioned, threshold-based TMA model [2], in which *planar* trinets are handled, only two situations were considered, corresponding to whether or not at least one of the two trinet faces sharing the current edge (where the ant is positioned) has less than a threshold number of sides.

Conceiving a reduced number of situations, and restricting the range of corresponding ant reactions, is useful for reducing the space of model instances to a manageable size, in view of its exhaustive exploration. But, with the GA approach, model space size is not a big concern, since evolution will find its own path across that space, ignoring large portions of it, and heading towards the regions with the fittest individuals. In fact, working with a larger space may increase the chances of finding interesting individuals. Thus, in this new GA experiment the situation is defined by the ordered triple:

$$(size(f_1), size(f_2), size(f_3))$$

of sizes (number of edges) of the three faces incident to the current trinet node – the node that the ant is facing. We have considered a range of 7 possibilities for face sizes: $2, 3, ..., 7, 8+$, where '8+' refers to a face with 8 or more sides. In conclusion, the ant distinguishes among $7^3 = 343$ possible situations.

Reaction the state change performed by the individual based on the current situation. In these GA experiments with TMA I have preserved the reaction policies of my previous experiments: the ant reacts to its current situation by choosing and applying graph rewrite rule *Expand* or *Exchange*, and by moving to one among 18 possible next locations in the neighborhood. In conclusion, the ant may react to a situation in $2 * 18 = 36$ possible ways.

Genome a function that associates a *reaction* to every possible *situation*:

$$genome : Situations \rightarrow Reactions$$

Each individual behaves according to its own genome. In light of the above definitions, the number of possible TMA genomes, i.e. different individuals, is 36^{343}, which is close to 10^{534} – a space size completely incompatible with exhaustive search.

Fitness a function that evaluates an individual's performance with respect to its intended goal. In light of the mentioned application to spacetime modeling, we are interested in a fitness function able to promote the production of causets with dimension above the already observed 1-D and 2-D cases. Out of several dimension estimation techniques, the above mentioned node-shell-growth analysis has been chosen, for reasons of simplicity, computational efficiency, and for its ability to extracts useful information also from totally ordered causets.[3]

[3] See [6] for a discussion on the apparently degenerate case of totally ordered causets.

In particular, I have focused on node-shell-growth relative to the causet root.

Informally, the implemented fitness function proceeds as follows. It attempts to match an initial subsequence of the node-shell-growth data, both by monomial $a * x^b$ and by exponential function $c * d^x$, using *Mathematica* function *FindFit*, where parameters a, b, c, and d are real constants, and x is the variable node-shell size. In both cases the fitting error is detected. Driven in part by previous experiments, I have given preference to causets with polynomial node-shell-growth. Thus, if a smaller error is found for the exponential fit, the individual is assigned a conveniently low value (-2, in the sequel), while if the polynomial fit is more accurate, a fitness value is assigned that corresponds to the obtained exponent b, with minor modifications meant to promote small errors and large numbers of node shells, while avoiding excessively small values for the multiplicative parameter a.

If, at some step, a loop edge is created in the trinet–a pathological situation in which rewrite rule application becomes problematic – the computation is forced into a fixpoint, and the trinet no longer evolves.

Mating the process of deriving a new generation of individuals from the current one. With generations of 100 individuals, 50 couples are selected, by using a random distribution that favors the choice of high-fitness elements. For couple (a, b), with genome strings g_a, g_b, a split point is chosen, which is an integer random variable s distributed around half the length of the genome (i.e. 343/2). A pair of children (c, d) are obtained from the couple by splitting both parental genomes at point s, and by combining the first part of g_a with the second part of g_b, and vice versa.

Random mutation the replacement of some genome elements by a random value. In our experiment, typically each element in the genome of a new born has a 4 percent probability of being replaced by a random value.

We are now ready for describing some of the preliminary results that have been obtained from the application of GA's to the evolution of ants living on trinets. In the sequel, I shall concisely refer to the latter by the term 'GA ants'.

4 Some Experiments and Preliminary Results

4.1 First Experiment

In a first experiment, the initial generation is formed by 100 GA ants with random genome codes; recall that these are tuples of 343 random integers in the range $[1, 36]$. The fitness of each ant has been computed by the discussed comparative polynomial and exponential fitting procedure, relative to a 2000-ant-step computation, yielding a 2000-node causet; 500 generations have been produced.

Three-four generations are sufficient for the average fitness to approach value 0 (from below), which corresponds to a 1-D node-shell-growth dimension, relative to the causet root.

This is not surprising, since the pervasive presence of rather uninteresting 1-D polymer-like cases was already observed in other algorithmic causet families.[4]

However, in 500 generations, and several hours of simulation, no increase of the average fitness above zero could be observed. Thus, an alternative way to stimulate the growth was explored.

4.2 Second Experiment

In a second round of experiments, special individuals have been included in the initial generation, hoping to trigger interesting evolutions. The special behaviors I have used were obtained by previous experiments with the *threshold-based Trinet Mobile Automata* mentioned above. The causets that they originate, their (*threshold*, *expandCode*, *exchangeCode*) parameters, and their fitness values are illustrated in Figure 2.

The choice of the two upper causets was motivated by their relatively high fitness values – these are approximately 2-D and 3-D causets, respectively, under the chosen node-shell-growth dimension estimator. The lower case was chosen simply because it represents one of the two exceptional pseudorandom cases that were found in the class of threshold-based automata.

The behavior of these threshold-based ants is fully defined by their three-element codes – a much smaller genome than the 343-element genome of a GA ant. However, these automata can be imported in the more elaborate GA setting because the situations recognized on the graph by a GA ant are precisely a *refinement* of those relevant to threshold-based ants. A GA ant detects the sizes (up to size 8) of *three* neighboring trinet faces, and reacts differently for each such triple, while a threshold-based ant only looks at *two* of them, and reacts only depending on how their sizes compare with the fixed threshold (for the refinement to hold, no threshold above 8 should be considered). On the other hand, the set of possible reactions – determining which rule to apply and where to move next – is the same in the two cases. As a consequence, any behavior of a threshold-based ant can be expressed by a GA ant genome, one in which many situations are regarded as equivalent and share the same reaction.

The inclusion of just one instance of ant (6, 10, 2) in an initial set of 100 otherwise random individuals proves insufficient for triggering anything of interest: the special individual disappears in the second generation without leaving trace of its high fitness, and the subsequent evolution is equivalent to the one obtained from the totally random initial generation.

A better result is obtained by associating 10 copies of ant (6, 10, 2) with 90 random individuals. In this case, within the first 7 generations several variants of the initial 'seed' appear, as shown in Figure 3.

Fitness values are also provided in that figure, relative to 1000 and 10,000 ant-step computations. Increasing the number of ant-steps improves the approximation to the expected fitness = 1 but, given the size and number of gener-

[4] In [4], for example, we show that 4084 out of 4096 elementary Turing machines yield 1-D, polymeric causets, when their tape is initially filled with 0's.

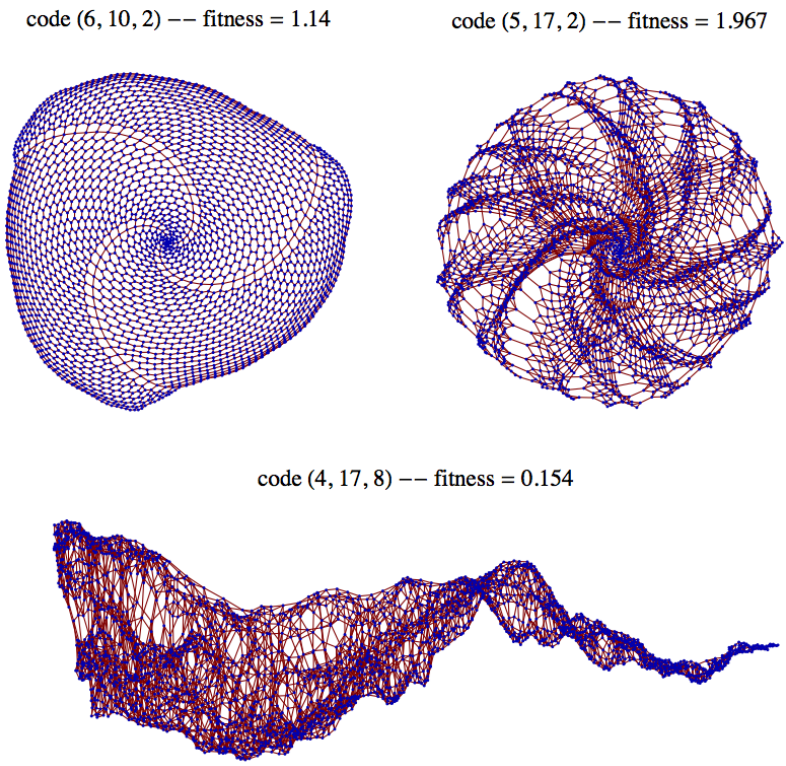


Fig. 2. The causets obtained from 3000-ant-step computations of three threshold-based Trinet Mobile Automata, their codes and their fitness values.

ations considered, letting the ants run for more than 1000-2000 steps is quite unpractical.

4.3 Third Experiment

In the third round of experiments, the random individuals have been completely removed from the initial generation. As a first choice, 50 copies of ant (6, 10, 2) and 50 copies of ant (4, 17, 8) (see Figure 2) have been used in the initial population, and the evolution of 200 generations has been simulated, taking over 24 hours to complete.

The individuals in the final generation are partitioned into two classes. We find 62 elements with fitness close to 0, or smaller, that correspond to the usual, uninteresting, polymer-like causets. The remaining 38 elements are of the type shown at the l.h.s. of Figure 4.

Subsequently, 10 instances of this newly found individual have been used for composing a new initial generation, where 40 instances of ant (4, 17, 8), and 50 instances of ant (5, 17, 2) (see Figure 2) have been added. The latter was

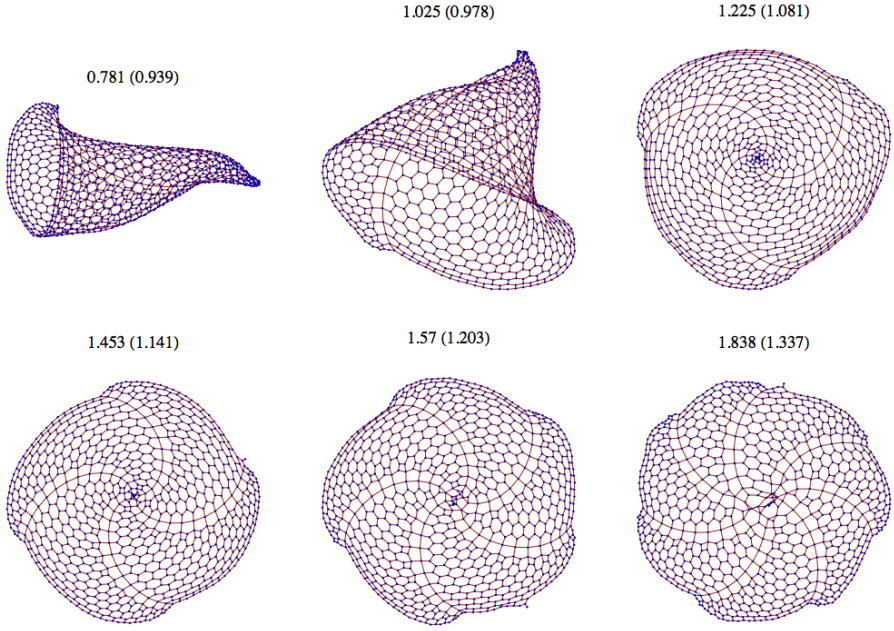


Fig. 3. Some of the causets obtained in 7 generations of Trinet Mobile Automata, when the initial generation is composed by 10 instances of ant (6, 10, 2) (see Figure 2) and 90 random individuals. The causets exhibit, respectively, 1, 2, 3, 4, 5 and 7 spiraling 'particles', formed by pentagonal faces in a hexagonal grid. Fitness values are shown, relative to 1000 and (in parentheses) 10,000 ant-step computations.

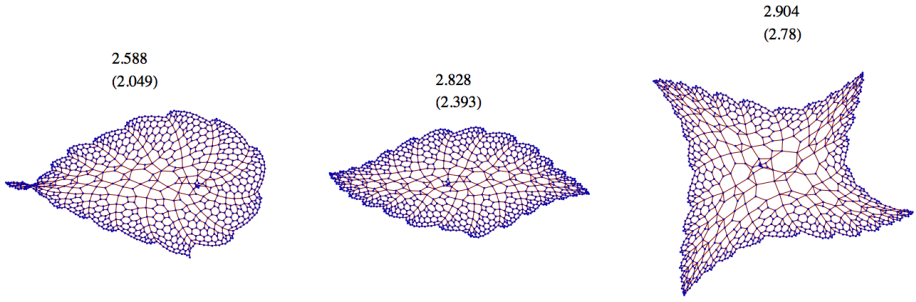


Fig. 4. Causets obtained from the evolution of Trinet Mobile Automata. The first was obtained from an initial generation composed by 50 copies of ant (6, 10, 2) and 50 copies of ant (4, 17, 8), and was used, jointly with copies of ants (4, 17, 8) and (5, 17, 2) (see Figure 2) for obtaining the second and third. Fitness values are shown, relative to 1000 and (in parentheses) 10,000 ant-step computations.

chosen because it builds the causet with fastest, polynomial node-shell-growth from the root, namely $O(n^2)$, that could be found among those derived from TMA computations.

Three identical experiments were conducted, each computing a sequence of 100 generations of 100 individuals, each individual running for 1000 steps. Due to the randomness involved in mating and mutations, we may expect different results for each simulation. Indeed, causets of a familiar type were obtained in two cases, with relatively low fitness figures. But in one case the final generation contains individuals that produce the causets shown in Figure 4. These causets appear as one-fold, two-fold, and four-fold realizations of the same basic pattern, and in this respect bear analogies with those in Figure 3. However, they achieve considerably higher fitness figures.

Although for applications to physics we should be in general more interested in pseudo-random computations, or in cases exhibiting a mix of order and disorder, inspecting regular cases, such as those in Figure 4, may reveal interesting features. The interest for these *planar and layered* causets comes on the fact that, in spite of their planarity, they achieve and even overtake the $O(x^2)$ node-shell growth rate of a *3-D* cubic lattice! Note that, although the ants work on planar trinets, and the adopted rewrite rules preserve planarity, the derived causets need not be planar; but in this case, they are.

This causet type was never observed among the (over 1000) cases that I have derived from threshold-based Trinet Mobile Automata. The conclusion is that the evolving ant, due to its rich and flexible genome, has entered a region that is out of the reach of the more primitive, threshold-based ant. On the other hand, similar planar, layered causets with $O(x^2)$ node-shell growth from the root, were found with other models of computation, namely with Turing machines, string rewrite systems, and cyclic tag systems; these are described in [4], where it is also proved that all causets from those models have to be planar.

One of the open problems of the computational universe conjecture is concerned with the choice among various candidate models of computation, or with questioning the actual relevance of such a choice, given the fact that all (Turing-complete) models are equivalent, at least in a formal sense. Causets may represent the appropriate abstraction for looking at these different models, and finding commonalities among different causet classes, as the one just exposed, may shed further light on the relevance or irrelevance of the above choice.

5 Further Experiments and Ongoing Work

Further experiments with Genetic Algorithms for Trinet Mobile Automata have been conducted, and more are under way. Most of the creativity in setting them up is concentrated in two aspects: (i) the definition of the *situation* to which the ant reacts, and (ii) the definition of the fitness function. The former may take advantage from additional structure introduced in trinets, e.g. node and/or edge labels.

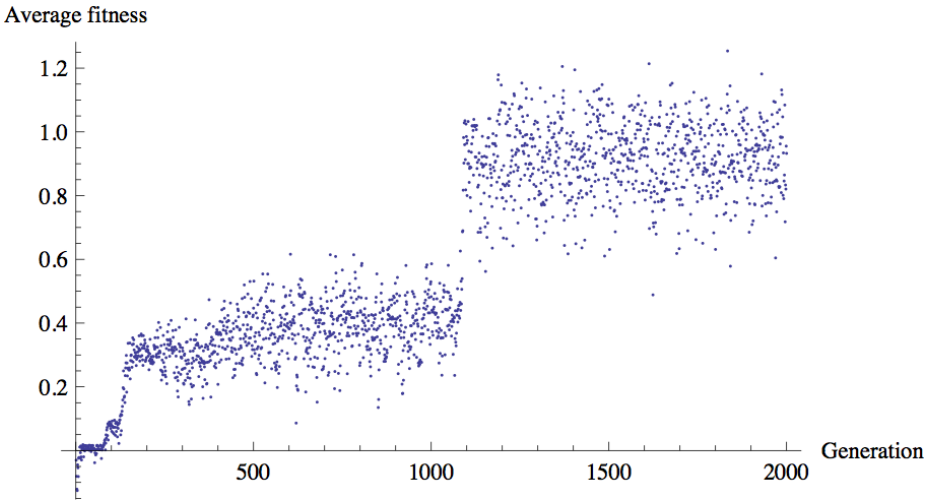


Fig. 5. The growth of the average fitness for a sequence of 2000 generations, each consisting of 100 individuals. The fitness is defined as in the previously described experiments, but the TMA algorithm is now based on trinets with three-colored edges. The ant reactions depend here on the detected color patterns.

For example, I have run a series of experiments with three-colored trinets, while keeping the same fitness function used in the previous experiments, meant to maximize node-shell-growth dimension from the causet root. A *Tait coloring* of a trivalent graph is a labeling of its edges made with only three colors, such that at each vertex the colors of the three incident edges are different (I am are grateful to Richard Southwell for stimulating interest in this approach). For an ant living on such a graph, the *situation* can be defined in terms of the color pattern that it detects around her.

One of the reasons for looking at three-colored trinets was to relax the trinet planarity condition. Graph rewrite rule *Exchange* has two forms, planar and nonplanar. In the previous experiments the first form was always chosen. But when looking only at edge colors, we lose control on this choice, and the trinet may soon become non planar. Following the intuition that nonplanar trinets could achieve higher (node-shell-growth) dimension than planar ones, and that the higher the dimension of the support, the higher the dimension of the causet derived from computing on it, [5] one could expect to hit relatively high fitness values with this type of TMA.

Figure 5 is a plot of the average fitness achieved, generation by generation, in a sequence of 2000 of them. The fitness function is the same of the previous experiments. Unfortunately, the values achieved so far are relatively low; however, the plot indicates that sudden increases in these values may take place after considerably long stationary phases.

[5] One can think of this increment as the contribution of the time dimension.

In another series of experiments on nonplanar Trinet Mobile Automata, the *situation* has been conceived based on the *age* of edges, that is, on their order of appearance in the growing graph. Assume that the ant is positioned on edge e_1, facing node n_1, and let e_2 and e_3 be the other edges incident to n, where e_2 is younger than e_3. The *situation* is then defined as the ordered pair $(L_{1,2}, L_{1,3})$ of lengths of the two shortest cycles containing, respectively, edge pair (e_1, e_2) and edge pair (e_1, e_3). In a way, these cycle lengths are the nonplanar analogous of the polygonal face sizes used for defining the *situation* in planar trinets. Three cycle lengths have been distinguished: 3 (parallel edges are excluded), 4, and 5 or higher. Thus, there are $3^2 = 9$ possible *situations*.

The *reaction* consists in choosing a rewrite rule in the set {*Expand*, *Exchange-planar*, *Exchange-nonplanar*} and then using a numeric code in the range 1-6

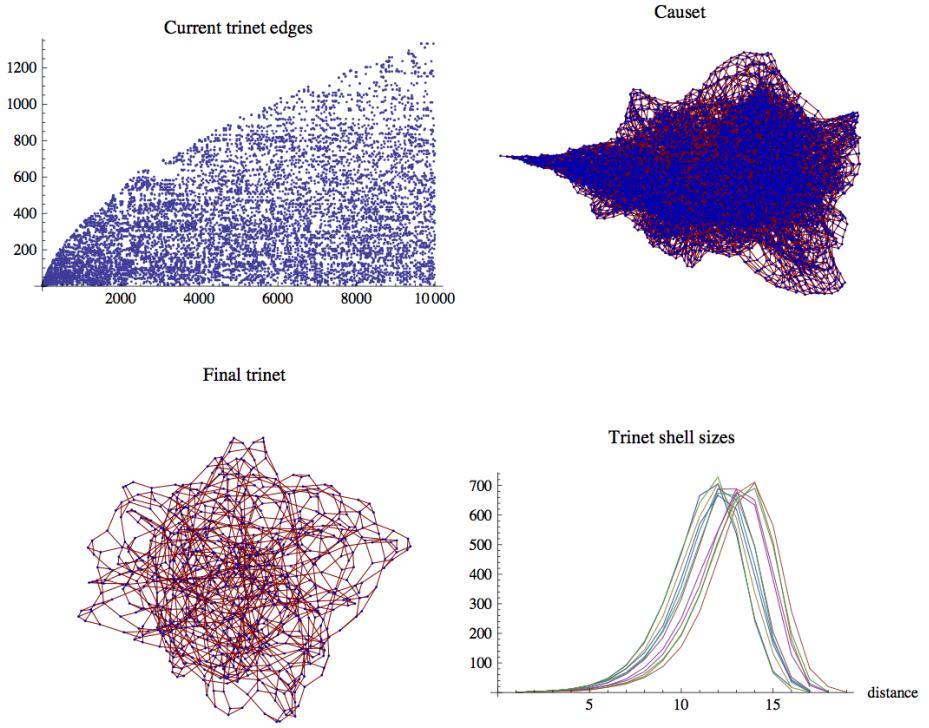


Fig. 6. The remarkable pseudo-random, 10,000-step computation of a nonplanar Trinet Mobile Automaton. Upper-left: sequence of edges visited and revisited by the ant. Upper-right: 10,000-node causet. Lower-left: final trinet. Lower-right: node-shell sizes at progressive distance from a trinet node, for 12 nodes selected at random in the graph, showing the uniformity of the latter (these 12 plots are actually based on a trinet with about 3500 nodes, larger than the one shown, and obtained by a 80,000-step computation).

to move to one of six possible next locations, defined relative to the current, oriented ant position. Thus, there are $3 * 6 = 18$ possible *reactions*.

Since the genome is a function from *situations* to *reactions*, there are a total of $18^9 = 198{,}359{,}290{,}368$ possible genetic codes, much less than with the previous TMA models.

Even with this new, nonplanar trinet setting, no improvement of fitness values can be reported yet. However, the search through the generations has led, among others, to the interesting individual illustrated in Figure 6.[6]

This is the third case of pseudorandom TMA that I have found, after the two cases described in [2]; but it is the first yielding a nonplanar trinet. The upper-left diagram in Figure 6 closely resembles the 'gem' found in Figure 1 – the thumbnail with parameters (17, 8) – and shares with it an $O(x^{1/2})$ growth rate, but only for an initial phase; subsequently, it appears to grow roughly linear, at least as far as I could experimentally test. The trinet produced by this computation appears completely uniform, in the sense that all trinet nodes seem to share the same view at the node-shell-growth rate, as illustrated by the lower-right plots.

6 Conclusions

During one of his late-evening talks at some NKS Summer School edition, Stephen Wolfram has indicated two types of achievement that can be expected from the NKS endeavor: breakthroughs in fundamental physics, and new insights in various fields of technology.

In the second case, the exploration of the computational universe, intended as the space of simple programs, may indeed help in identifying unorthodox solutions to computing problems that would be unaccessible by traditional engineering techniques; the development of a complex software system such as the Wolfram Alpha knowledge engine, for example, seems to have benefitted from this approach.

In this paper, however, we have concentrated on the first goal: we have formulated some visions on the desirable evolutionary steps that systems of artificial particles, similar to those in Wolfram's ECA 110, might take for building systems of increasing complexity, and, hopefully, increasing similarity with the physical universe; and we have shown some applications of Genetic Algorithms to the challenging activity of 'universe hunting'.

Under the picture of a physical universe organized as a layered architecture of emergence, 'universe hunting' primarily refers to the simulation of computational Big Bangs, looking at the bottom layers of the architecture in search for structures, objects, behaviors, laws, relevant at those levels. There is no guarantee that what we find there has a *direct* counterpart in (let alone numerical correspondence with) the repertoire of objects and laws from current physical theories, that are to be probably located several layers above. But, following

[6] According to our implemented conventions for reaction coding, this individual has genetic code {5, 7, 17, 8, 3, 3, 13, 1, 11}.

Einstein, we can believe that the universe is, surprisingly, understandable; additionally, we could count on the fact that its layers show imaginative variations of a few recurrent patterns. Thus, in simulating the lower layers, we can hope to detect at least analogies with what we already know about higher layers – those that we can directly explore, say, by particle accelerators.

For qualifying as a scientific theory, any computational Big Bang conjecture will eventually need a stringent validation in the form of an accurate numerical prediction of some known physical constant, or of the outcome of some 'real' experiment. After Galileo, there is no escape from this requirement. In fact, I do not consider this event as totally unconceivable. However, given the gap that separates, say, the 'artificial' localized structures of some TMA computation from the mass of a 'real' electron, much work still remains to be done, at both ends of the gap.

In particular, we still need plenty of creative experiments for simulating, bottom up, the the lower levels of the architecture. The fact that most of the work done at these levels proceeds somewhat blindly, with the primary objective to see what might possibly happen, makes it quite reasonable to call this approach 'a new kind of science'.

Acknowledgments. I would like to thank Hector Zenil for his kind invitation to contribute to this volume, celebrating the tenth anniversary of Wolfram's NKS book.

References

[1] Agha, G.: ACTORS: A Model of Concurrent Computation in Distributed Systems. The MIT Press, Cambridge (1990)
[2] Bolognesi, T.: Planar trinet dynamics with two rewrite rules. Complex Systems 18(1), 1–41 (2008)
[3] Bolognesi, T.: A pseudo-random network mobile automaton with linear growth. Inf. Process. Lett. 109(13), 668–674 (2009)
[4] Bolognesi, T.: Causal sets from simple models of computation. Int. Journ. of Unconventional Computing 6(6), 489–524 (2010)
[5] Bolognesi, T.: Causal sets from simple models of computation (April 2010), `http://arxiv.org/abs/1004.3128v1` [physics.comp-ph]
[6] Bolognesi, T.: Algorithmic causets. In: Proceedings of DICE 2010: Space, Time, Matter. Journal of Physics: Conference Series, vol. 306(1). IOP Science (2011), doi:10.1088/1742-6596/306/1/012042
[7] Bolognesi, T.: Reality is ultimately digital, and its program is still undebugged. FQXi Essay Contest 2011 (2011), `http://fqxi.org/community/essay/winners/2011.1`
[8] Bolognesi, T.: Algorithmic causal sets for a computational spacetime. In: Zenil, H. (ed.) A Computable Universe - Understanding Computation and Exploring Nature as Computation. World Scientific (2012)
[9] Bombelli, L., Lee, J., Meyer, D., Sorkin, R.D.: Space-time as a causal set. Phys. Rev. Lett. 59(5), 521–524 (1987)

[10] Dowker, F.: Causal sets and the deep structure of spacetime. In: Ashtekar, A. (ed.) 100 Years of Relativity - Space-time Structure: Einstein and Beyond, pp. 445–464. World Scientific (2005) arXiv:gr-qc/0508109v1
[11] Hewitt, C.: What is computation? Actor Model versus Turing's Model. In: Zenil, H. (ed.) A Computable Universe - Understanding Computation and Exploring Nature as Computation. World Scientific (2012)
[12] Hewitt, C., Baker, H.: Laws for communicating parallel processes. Information Processing (1977)
[13] Hoare, C.A.R.: Communicating sequential processes. Commun. ACM 21(8), 666–677 (1978)
[14] Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press (1975)
[15] Kauffman, S.A.: At home in the universe: the search for laws of self-organization and complexity. Oxford University Press, Oxford paperbacks (1995)
[16] Markopoulou, F.: Dual formulation of spin network evolution (1997) arXiv:gr-qc/9704013v1
[17] Milner, R.: Communication and concurrency. Prentice-Hall, Inc., Upper Saddle River (1989)
[18] Mitchell, M.: Complexity: a guided tour. Oxford University Press (2009)
[19] Nowotny, T., Requardt, M.: Dimension Theory of Graphs and Networks. J. Phys. 31, 2447–2463 (1998) arXiv:hep-th/9707082, doi:10.1088/0305-4470
[20] Pegg, E.: Turmite. In: Weisstein, E.W. (ed.) MathWorld–A Wolfram Web Resource, `http://mathworld.wolfram.com/Turmite.html`
[21] Pegg, E.: Turmites. The Wolfram Demonstration Project, `http://demonstrations.wolfram.com/Turmites/`
[22] Pegg, E., Bryant, J., Gray, T.W.: Voronoi diagrams. The Wolfram Demonstration Project, `http://demonstrations.wolfram.com/VoronoiDiagrams/`
[23] Rideout, D.P., Sorkin, R.D.: Classical sequential growth dynamics for causal sets. Phys. Rev. D 61, 024002 (1999) arXiv:gr-qc/9904062v3
[24] Sorkin, R.D.: Causal sets: Discrete gravity, Notes for the Valdivia Summer School (2003) arXiv:gr-qc/0309009
[25] Wolfram, S.: A New Kind of Science. Wolfram Media, Inc. (2002)

Chapter 13
Artificial Cosmogenesis: A New Kind of Cosmology

Clément Vidal

Center Leo Apostel, Global Brain Institute
Evolution, Complexity and Cognition Research Group
Vrije Universiteit Brussel (Free University of Brussels), Belgium
clement.vidal@philosophons.com
http://clement.vidal.philosophons.com

Abstract. This paper introduces foundations for a new kind of cosmology. We advocate that computer simulations are needed to address two key cosmological issues. First, the *robustness* of the emergence of complexity, which boils down to ask: "what would remain the same if the tape of the universe were replayed?" Second, the much debated *fine-tuning* issue, which requires to answer the question: "are complex universes rare or common in the space of possible universes?" We argue that computer simulations are indispensable tools to address those two issues scientifically. We first discuss definitions of *possible universes* and of *possible cosmic outcomes*—such as atoms, stars, life or intelligence. This leads us to introduce a generalized Drake-like equation, the *Cosmic Evolution Equation*. It is a modular and conceptual framework to define research agendas in computational cosmology. We outline some studies of alternative complex universes. However, such studies are still in their infancy, and they can be fruitfully developed within a new kind of cosmology, heavily supported by computer simulations,

Artificial Cosmogenesis. The appendix [A] provides argumentative maps of the paper's main thesis.

Keywords: artificial cosmogenesis, cosmic evolution, computational cosmology, digital physics, Drake equation, Cosmic Evolution Equation, robustness, fine-tuning, multiverse.

What I cannot create I do not understand
On Richard Feynman's blackboard
at time of death in 1988, as reported in [29]

1 Introduction

I am fond of both computer science and cosmology. However, the methods, concepts and tools used in those two disciplines are very different. Is it possible to

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 157–182.
DOI: 10.1007/978-3-642-35482-3_13

unite this dual passion? This essay outlines foundations for such a new kind of cosmology, *Artificial Cosmogenesis*.

Broadly speaking, we can distinguish three kinds of science: *deterministic*, *probabilistic* and *computational*. Deterministic science can roughly be characterized by the science Newton practiced. He used physical laws and initial conditions to predict the future and explain the past. The predictions are of an amazing accuracy and the tools used are mathematical equations which are relatively easy to solve. Because of its successes, it is often implicitly considered the typical model of hard science.

However, when there are too many particles in a system, their sheer number and interactions make the newtonian approach weak. In fact, even with only three gravitational bodies the newtonian theory of gravitation fails to make practically useful predictions. The main insight of the founders of statistical physics was to average out the interactions of particles to derive *statistical* laws of behavior, such as the laws of thermodynamics or quantum mechanics.

In recent years, Laurent Nottale generalized this statistical predictability to all scales in nature, by unifying relativity theories with microphysics (see e.g. [41, 42, 43]). This *scale relativity* theory constitutes a revolution in progress in the domain of theoretical physics, since its leads to fundamental theoretical results as well as highly precise and validated predictions (see also [63], p96-97).

But what if our statistical methods also fail or are absent? What if we do not know any way to predict the behavior of a very complex system? An even more general approach is needed. This can be done in a computational view of nature, by theorizing and experimenting with algorithms (see e.g. [71, 70]). The field of Artificial Life constitutes a remarkable application of this view, when it attempts to decipher the most general laws of life, and then to implement and experiment with them in computers. Stephen Wolfram [69] argued at length how important this new kind of science based on computer simulations is. He advocated a wide exploration of simple programs, to study their behavior and properties. He argued that such a new approach is unavoidable if we want to understand complex dynamics. As a matter of fact, the study of complex dynamical systems will in most cases *not* be predictable with simple equations. Wolfram [68, 69] further conjectured that most systems in nature are *computationally irreducible*. This means that to study complex systems, there is no shorter way than to run step by step the model, and study how it behaves (see also [72] for a general formal definition of irreducible computation). Such a kind of science can still make predictions because simulations can be run faster than reality. Studying complex systems, equations won't help, simulations will.

Of course, when possible, it is best to aim for absolute and precise predictions such as in Newtonian science. When this fails, statistical laws are the second best option. But most real and complex systems may not be predictable in these two ways. A broader general computational exploration promises to be the way to understand the rise and evolution of complexity.

My aim in this paper is to propose a computational approach to progress on two arduous cosmological issues. First, the *robustness of the emergence of complexity* in our universe; second, the question of how *fine-tuned* our universe is.

The question of the robustness of the emergence of complexity can simply be illustrated by a thought experiment. *What would remain the same if we would replay the tape of the universe?* To address this issue, we introduce the *Cosmic Evolution Equation* (CEE). It is a modular conceptual framework to discuss possible universes, possible cosmic outcomes, the robustness of the universe and fine-tuning. To define it, we build on Drake's [19] equation in the Search for Extraterrestrial Intelligence (SETI) and on the thoughtful discussion of possible universes by Ellis, Kirchner and Stoeger [23].

The fine-tuning issue is much debated and intricate. The problem is that if we vary one by one a number of parameters, both in cosmological and standard particle models, no life or no complexity of any sort emerges (see e.g. [35, 44, 16]). The issue is mined with logical and probabilistic fallacies (e.g. [39, 14]) as well as physical fallacies (see e.g. [64, 67, 53]). It is also commonly confused with other related issues such as free parameters, parameter sensitivity, metaphysical issues, anthropic principles, observational selection effects, teleology and God's existence [67].

Additionally, different discussions of fine-tuning focus on very different *cosmic outcomes*. We see fine-tuning discussions regarding the dimensionality of space [44], the production of carbon atoms in stars [30], the existence of long-lived stars [1]; the number of black holes [49]; biochemistry [5]; but also complexity of any sort [20]. A key question to clarify the issue is thus to explicitly ask: *fine-tuning for what?* Which cosmic outcome are we interested in? In particular, we will see that most fine-tuning arguments are poor, since they vary parameters one by one, which is a fallacy resulting in exploring only 0,00000000000000456 % of the parameter space under consideration!

To remedy this situation, we generalize the CEE. The Drake equation estimates the number of communicative intelligent civilizations in our galaxy. By extension, one application of the generalized CEE is to estimate the likelihood of our particular universe in the space of possible universes. In other words, if Drake's equation allows to estimate the probability of life existing "somewhere in the galaxy"; one application of the CEE is to estimate the more general probability of life existing "*anywhere* in the space of possible universes". *Artificial Cosmogenesis*—ACosm for short—is the study of alternative cosmic evolutions and allows in principle to assess how fine-tuned our universe is.

We first discuss the issues of possible universes and possible cosmic outcomes (sections 2 and 3). Then we introduce the CEE to discuss the robustness issue (section 4) and generalize the CEE to address the fine-tuning issue (sections 5-6). By bridging the gap between computer science and cosmology, I hope this framework will fruitfully pave the way for resolving these two fundamental cosmological issues.

2 Possible Universes

What are the possible universes? How can we describe the space of possible universes? These questions raise enormous logical, metaphysical, philosophical,

and scientific problems. Although possible universes or possible worlds have been discussed centrally in the history of philosophy (see e.g. [2, 36], see also [18] for a wider historical perspective), our aim here is to formulate the issue of possible universes so that it can progressively exit metaphysics and enter the realm of operational science.

We now follow Ellis', Kirchner's and Stoeger's [23] definition of the class of all possible universes. Let M be a structural and dynamical space of all possible universes m. Each universe m is described by a set of states s in a state space S. Each universe m is characterized by a set P of distinguishing parameters p, which are coordinates on S. Such parameters will be logical, physical or dynamical. How will they dynamically evolve? The three authors elaborate:

> Each universe m will evolve from its initial state to some final state according to the dynamics operative, with some or all of its parameters varying as it does so. The course of this evolution of states will be represented by a path in the state space S, depending on the parametrisation of S. Thus, each such path (in degenerate cases a point) is a representation of one of the universes m in M. The coordinates in S will be directly related to the parameters specifying members of M.

In such a procedure, we face a first major issue:

> **Possibility space issue:** *What delimits the set of possibilities? What is the meta-law or meta-cause which determines M?*

As the three authors argue, we can't avoid the meta-law issue, because otherwise we have no basis to set up a consistent description of M. We need to have a logic which describes M. There are other difficult issues related to identifying which different representations represent the same universe models—*the equivalence problem*— and the problem of dealing with an *infinite space of possible universes*. I refer the reader to the three authors' paper for more in depth discussions of these issues.

More directly related to the fine-tuning issue is the remark of Jean-Philippe Uzan that "the larger the possibility space considered, the more fine-tuned the actual universe appears to be" (in [23], p923). Indeed, we can easily increase the unlikelihood of our universe simply by allowing the parameter space to grow. You could ask for example, did you explore if universes with 42 dimensions generate life? Do we really want to capture the radical idea of "all that can happen, happens"? There is much variation in the space of possibility we want to delimit. Ellis ([21], p1261) distinguishes four levels of variation, *weak*, *moderate*, *strong* and *extreme*:

> • "*Weak variation*: e.g. only the values of the constants of physics are allowed to vary? This is an interesting exercise but is certainly not an implementation of the idea 'all that can happen, happens'. It is an extremely constrained set of variations.
> • *Moderate variation*: different symmetry groups, or numbers of dimensions, etc. We might for example consider the possibility landscapes of

string theory [24] as realistic indications of what may rule multiverses [24, 55, 56]. But that is very far indeed from 'all that is possible', for that should certainly include spacetimes not ruled by string theory.
• *Strong variation*: different numbers and kinds of forces, universes without quantum theory or in which relativity is untrue (e.g. there is an aether), some in which string theory is a good theory for quantum gravity and others where it is not, some with quite different bases for the laws of physics (e.g. no variational principles).
• *Extreme variation*: universes where physics is not well described by mathematics; with different logic; universes ruled by local deities; allowing magic as in the Harry Potter series of books; with no laws of physics at all? Without even mathematics or logic?"

We indeed need to make a choice between theoretical physics and magic... or anything in between.

Do we need to assume an actual multiverse? No we do not. To study the fine-tuning issue, we need only*possible* or *virtual* universes, not actually realized ones. This interpretation still allows us to use the vast multiverse literature to define and explore possible universes, without making strong and problematic ontological claims regarding their actual existence.

3 Possible Cosmic Outcomes

Once we settle on a framework to define possible universes, a second major issue is to specify the parameters which differentiate possible universes:

> **Cosmic outcomes issue**: *What are the cosmic outcomes? What are the milestones of cosmic evolution? What parameters differentiate possible universes? How do we find those parameters?*

As the three authors mention, the values of the parameters may not be known initially. They may emerge out of *transitions* from one regime to another. For example, sociologists do not explore alternative sociological structures by varying the mass of elementary particles. They start from different, less fundamental parameters, such as the influence of population density, the climate or the media. *The challenge to understand complexity transitions in cosmic evolution is of upmost importance and difficulty.* For example, how did atoms emerge out of the big bang era? How did planets form out of stars and stardust? How did life originate out of molecules? How did consciousness emerge from biological organisms? Etc.

The ideal of reducing such parameters is a major goal of science. The objective is to build a consistent theory and narrative of cosmic evolution, which explains a maximum of cosmic outcomes with a minimum of parameters. Scientific progress is achieved when new theories capture previously free and unexplained parameters (see e.g. [64] for an illustration in physics). We could now extend this attitude to attempt a reduction of other higher parameters (such as life)

to fundamental physics and cosmic parameters. However, since we are still very far from such a feat, in our description of possible universes we must assume explicitly higher parameters. Typically, when researchers tackle the issue of the origin of life, they don't start from big bang nucleosynthesis, but they assume the existence of molecules.

Ellis, Kirchner and Stoeger categorize the parameters from the most basic ones to the most complex ones. They distinguish different categories of parameters p_j, with $j = 1$ - 2 describing basic physics; $j = 3$ - 5 describing cosmology and a category of parameters $j = 6$ - 7 related to the emergence of life and higher complexity.

Each category p_j is composed of different parameters i. For example, $p_1(i)$ are basic physics parameters, such that the fine-structure constant; masses, charges and spins of particles, as well as other dimensionless parameters. I refer the reader to the detailed description of the parameters given by the three authors.

However, in each parameter category I would like to add explicitly some random, chance or noise parameters. For example, these could include for $j = 1$ - 5 quantum effects in the early universe; or nonlinear chaotic dynamics which might trigger catastrophic events, such as meteorites impacting planets for $j = 7$. This would certainly complicate the dynamics, but would also make it much more realistic. A dynamical argument can even be advanced that such random events might be essential to the open-ended growth of complexity. An illustration can be found in engineering with the heuristic of *simulated annealing*. It starts by adding important noise into the system, and then gradually reduces it. The purpose of the noise is to shake the system to reach a maximally stable configuration.

Now, how do we decide which cosmic outcomes to keep, and which ones to leave out? At first, we can aim at including a maximum of parameters. Then, we would progressively reduce the number of parameters, as we get better and better insights on how they emerge from more fundamental principles and theories; i.e. from previous parameters. Robert Aunger ([3], p1142-1144) did compile from many authors a list of more than 100 different cosmic outcomes. This is the most comprehensive review I am aware of, ranging from the big bang, the formation of atoms, stars, solar systems, life, DNA, multicellularity, sexual reproduction, fishes, to mammals, agriculture, modern science and space exploration.

However, we can already anticipate a fallacy lurking when considering a large list of cosmic outcomes. Similarly to Uzan's remark for the space of possible universes, we can note that the more cosmic outcomes we have, the more unlikely they will seem. The extreme case is to consider one single object as a cosmic outcome. For example, in intelligent design discussions, they consider a complex object (like a living organism or an airplane) and try to assess the likelihood that it arose by chance. Of course this will be very unlikely! Additionally, as Dawkins [17] argues, natural selection would still constitute a much better candidate explanation than design. A scientist will look for possible mechanisms, theories, which can explain the emergence of complexity. The *a posteriori* probability of a single object isolated from its evolutionary or human context is of weak scientific interest.

To avoid such an error, we need to advance *theoretical reasons* to select certain cosmic outcomes and not others. This is rarely attempted. Most authors propose an arbitrary list without strong theoretical justification. Ellis, Kirchner and Stoeger did not justify their choice of distinguishing parameters; although it is clear that they included a lot of cosmological parameters necessary for their subsequent study of alternative universes with different geometries.

The most promising avenue of research is to focus on thermodynamics (see e.g. [47]). Indeed, all systems need to process energy, which is therefore a universal concept, applicable from the beginning of the universe to our energy hungry technological society. Robert Aunger [3, 4] built on a thermodynamical theory to select cosmic outcomes, *non-equilibrium steady-state transitions*. Each transition involves first an energy innovation, then a structural adjustment and finally a new control mechanism. He thus constructed a consistent selection of cosmic outcomes and evolutionary transitions.

Which cosmic outcomes are contingent and evolutionary? Which ones are necessary and developmental? Are there attractors in the dynamic of cosmic evolutionary development? To answer these issues, we need to explore the *robustness* of the emergence of complexity. Stated otherwise, if we would re-run the tape of the universe, would galaxies, stars, biology and technology arise again and again? The straightforward way to answer those questions, in parallel to a theoretical rationale like Aunger's, is indeed to re-run the tape of the universe. Let us now examine how we can conceptualize and do that.

4 Robustness in Cosmic Evolution

what would remain the same if the tape of life were replayed?
Stephen Jay Gould [25]
what would remain the same if the tape of the universe were replayed?
Paraphrasing Gould's question to the universe [62]

Answering this latter question, Paul Davies ([15], p317) wrote that if "the universe were re-run a second time, there would be no solar system, no Earth and no people. But the emergence of life and consciousness somewhere and somewhen in the cosmos is, I believe, assured by the underlying laws of nature." Those claims, as Davies acknowledges, are only informed intuitions. How can we test this intuition or different ones scientifically? This is the issue of the *robustness of the emergence of complexity in cosmic evolution*.

A first analysis of the tape metaphor shows its limits. Indeed, if the tape and its player were perfect, we should get exactly the same results when re-running the tape. So, the thought experiment would be trivial. Yet if our universe self-constructs, one question is whether small fluctuations, chance events, noise or random perturbations would lead to slightly different outcomes, or very different ones. This makes the issue of robustness in cosmic evolution highly stimulating.

It is very hard to tackle because it is linked to a great weakness of cosmology as a science: it has only one object of study, our unique universe. More precisely, we can distinguish two fundamental limitations that Ellis ([21], 1216) pointed out:

> **Thesis A1: The universe itself cannot be subjected to physical experimentation**. *We cannot re-run the universe with the same or altered conditions to see what would happen if they were different, so we cannot carry out scientific experiments on the universe itself.* Furthermore,
>
> **Thesis A2: The universe cannot be observationally compared with other universes**. *We cannot compare the universe with any similar object, nor can we test our hypotheses about it by observations determining statistical properties of a known class of physically existing universes.*

Our thesis is that it is possible to address those limitations and the issue of robustness by running computer simulations of our universe. It is important to note that if we replay the tape of *our* universe, we don't aim to actually explore the full space of possible universes. Here, we only aim to assess the robustness of the emergence of the different cosmic outcomes. We thus vary *only* nondeterministic dynamical parameters we discussed above (quantum mechanical effects, random perturbations, nonlinear chaotic dynamics, etc.). An open question is also how we vary the random parameters. How often? How strong is the variation? Various distributions can be tested, from gaussian distributions, where most random variations are of an average strength, few are weak or strong; to power-law distributions, where there are few very strong variations, some medium variations, and most of the time weak random variations.

Because of the inclusion of such parameters, it makes sense to re-run the same universe simulation. By running a multitude of times the simulation, it will be possible to make statistics on the emergence of complexity. An even more straightforward way to make such statistics would be to drastically intensify astrobiology—the search for extraterrestrials. If or when we will find extraterrestrials, we would be able to progressively study the "natural re-runs" of complexity. Additionally, searching for extraterrestrials more complex than us would force us to break with the implicit anthropocentric assumption that life and humans on Earth are the highest development in cosmic evolution. This invites us to speculate on the existence of higher cosmic outcomes, and this opens the way to test our theories of the general evolution of cosmic complexity (see e.g. [10, 65] for modern views on the search for advanced extraterrestrials).

An example of ambitious simulations of *our* universe are the Millennium run simulations [50, 9, 27]. The authors studied the formation, evolution and clustering of galaxies and quasars within the standard (or concordance) model of cosmology. Although they did not run the same simulation in its full complexity many times, the volume space explored is large enough to extract meaningful statistical properties on the evolution of the distribution of matter.

Replaying the tape of our entire universe is still a much more ambitious project, which at present remains unrealistic. We should remain aware that our current models and their associated free parameters are most likely not the ultimate ones. Of course, new theories need to be developed to know what the key parameters of our universe are. In the meantime, a way to progress is to break down the issue into smaller solvable problems. For example, if we want to tackle the robustness up to the emergence of intelligent life, we can write a generalized Drake equation ([23], p925) that we call the *Cosmic Evolution Equation*:

$$N_{life}(m^*) = N_g \cdot N_S \cdot f_S \cdot f_p \cdot n_e \cdot f_l \cdot f_i$$

where $N_{life}(m^*)$ is the number of planets with intelligent life in our particular universe m^*; and

- N_g is the number of galaxies in the model
- N_S is the average number of stars per galaxy
- f_S is the fraction of stars suitable for life
- f_p is the fraction of such stars with planetary systems
- n_e is the mean number of planets which are suitable habitats for life
- f_l is the fraction of planets on which life originates
- f_i is the fraction of life bearing planets with intelligent life.

There are many implicit assumptions in such a framework, for example that life-supporting stars will be Sun-like; or that life starts necessarily on planets and not on more exotic places. We also implicitly assume that the parameters are independent. To deal with dependent parameters, one would need to introduce a bayesian probability framework. Additionally, we may have clear definitions of what stars or galaxies are, but the issues of defining higher cosmic outcomes such as life or intelligence remain of huge scientific debate.

The factors N_g and N_S can nowadays be estimated, while the recent explosion of exoplanets discoveries is allowing us to estimate more and more precisely the factors $f_S \cdot f_p \cdot n_e$. However, huge uncertainties remain regarding the last two factors $f_l \cdot f_i$.

The main interest of such a framework—whether we consider these seven factors to be most relevant or others—is that we can in a first approximation estimate the factors independently. Additionally, *the more we progress in our knowledge of the universe, the larger the distance between factors we can assess.* For example, assessing the number of planets with intelligent life knowing only the number of galaxies seems very hard. But shorter distances between factors are easier to assess. For example, Miller's [40] famous experiment tells us that the probability to have amino acids out of a primordial soup and some energy source is high. Which is indeed an important insight to evaluate $n_e \cdot f_l$.

Let us now imagine that we run multiple times a model of our entire universe m^*. We would be able to interpret the results of the multiple runs of the simulation as a set of *virtual* universes. We would end up with a distribution function $f(m^*)$ combining the probability distributions obtained for each factor.

However, we need to further specify a *possibility space*, which in this case is M^* resulting from the variation of random parameters only; and a measure π^* on M^*. Such a virtual ensemble of simulated universes V would thus be defined as:

$$V = \{M^*, \pi^*, f(m^*)\}$$

The number of planets with intelligent life would then be:

$$N_{life}(m^*) = \int N_g \cdot \ N_S \cdot \ f_S \cdot \ f_p \cdot \ n_e \cdot \ f_l \cdot \ f_i \cdot \pi^*$$

Note that the integral is necessary to normalize the result according to the measure π^* and distribution function $f(m^*)$.

There are important and subtle issues to make this normalization sound and possible (see again[23]).

Let us give some more concrete possible results such simulation studies would bring. We might conclude that our universe is robust for galaxy-formation, i.e. most simulation runs lead to galaxy formation.

But still, it might turn out that our universe is not robust for intelligent life, i.e. most simulations *do not* lead to the emergence of intelligent life.

We can now take a fresh eye on our question: are cosmic outcomes necessary or contingent? We can define a cosmic outcome as *necessary* if it appears again and again as we re-run the same universe simulation, as *contingent* otherwise. For example, let us take the DNA code in biology: is it necessary that there is a unique DNA code for terrestrial or extraterrestrial biology? In a similar fashion, in economy, is it a necessity in civilizational development that monetary systems converge to a common currency?

We can also compare the cosmic outcome selections. On the one hand we would have the ones resulting from "simulation experiments" (see e.g. [32] for a discussion); and on the other hand the theoretical approaches (such as Aunger's). *Simulation experiments* in cosmology can play the role that *empirical experiments* play in other sciences. This approach can be called "cosmology in silico" or "computational cosmology". In fact, these endeavors are already developing quickly, as illustrated by the Virgo Consortium for Cosmological Supercomputer Simulations.

We have just begun to explore how robust the emergence of complexity in our universe is. If we want to understand it better, we need to perform computer simulations and use existing conceptual, mathematical and statistical tools to design simulation experiments and to assess the results.

However interesting and important this enterprise is, it does not tackle the fine-tuning issue. Indeed, in studying the robustness of our universe, we try to understand the emergence of complexity in *our universe*, whereas to address fine-tuning we must study the place of our particular universe in *the space of possible universes*.

5 Artificial Cosmogenesis or the Study of Alternative Cosmic Evolutions

Now, we create a considerable problem. For we are tempted to make statements of comparative reference regarding the properties of our observable Universe with respect to the alternative universes we can imagine possessing different values of their fundamental constants. But there is only one Universe; where do we find the other possible universes against which to compare our own in order to decide how fortunate it is that all these remarkable coincidences that are necessary for our own evolution actually exist?

Barrow and Tipler ([6], p6)

you might end up having a future subject which is "comparative universality"—we have all these laws for the universe that cannot be eliminated as ours and you study them, you talk about them, you compare them, this could be a future subject. Students would be required to pass exams on their ten possible favorite universes ...

Gregory Chaitin ([1], p339)

This first quote by Barrow and Tipler summarizes the core problem of fine-tuning. The second quote by Chaitin illustrates a core idea towards its resolution. With the robustness issue, we have focused on *our* universe. To assess in how far our universe is fine-tuned, we must study the place of our universe in the *space of possible universes*. We call this space the *virtual multiverse*.

Fine-tuning arguments vary just one parameter, a fallacy which is nearly *always* committed. The underlying assumption is that parameters are independent. As Stenger ([53], p70) remarks, this is "both dubious and scientifically shoddy". If the history of physics learned us something, it is that phenomena which where thought to be widely independent, turned out to have common underlying causes and principles. For example, our common sense fails to see a connection between the fall of an apple and the tides; magnetism and electricity; and even less between space, time and the speed of light. But all these phenomena have been unified thanks to physical theories.

Additionally, varying several parameters without care can lead to what is known as the *one-factor-at-a-time* (OAT) paradox in sensitivity analysis. The problem with the OAT method is that it is non-explorative. Let us see why. At first sight, it seems logical and rigorous, since it varies factors one-at-a-time while keeping the others constant. It seems consistent because the output from a change can be attributed unambiguously to the change of one factor. It also never detects non-influential factors as relevant. However, by construction, this method is non-explorative, with exploration decreasing rapidly with the number of factors. For a simple example, consider Figure 1, which shows clearly that OAT explores only 5 points forming a cross, out of 9 points in total.

Let us now generalize this example with a geometrical interpretation of the parameter space. In n-dimensions, the n-cross will necessarily be inscribed in the n-sphere. The problem is that this n-sphere represents a small percentage

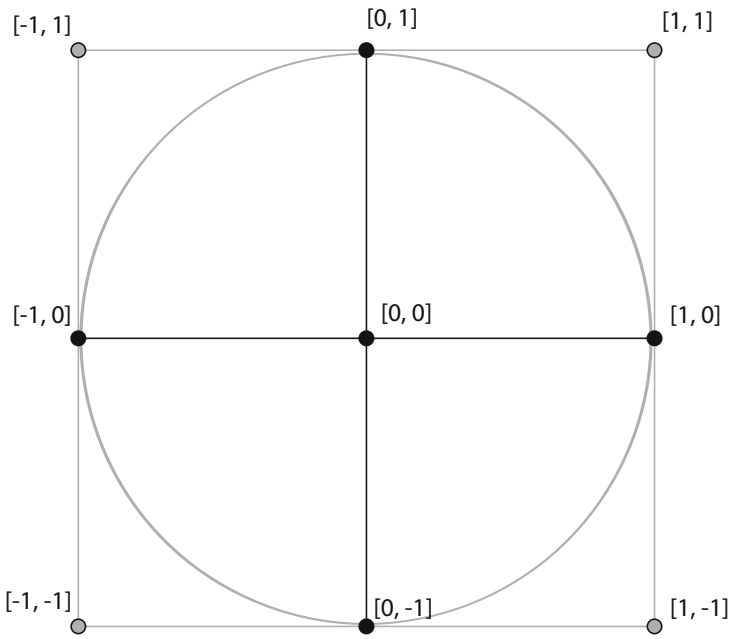


Fig. 1. The one-factor-at-a-time method can only reach points on the cross. In this simple two-dimensional parameter space, each discrete factors can only take values 0, 1 or -1. OAT can reach [0, 0], [0, 1], [0, -1] (points on the vertical line); and [-1, 0], [1, 0] (points on the horizontal line). The points explored are thus on a cross. The points not explored are the corners [-1, 1], [-1, -1], [1, 1], [1, -1]. In a geometrical interpretation, note that the cross is by construction inscribed in the circle. But OAT actually restricts the exploration to points on the cross, not inside the circle because exploring points inside the circle would imply varying two parameters at the same time. Now, that cross itself is inscribed in the circle. In sum, OAT restricts the exploration to the cross, not the circle, but the cross is inscribed in the circle. And this circle is inscribed in the square (2-cube), which is why OAT can't reach the corners of the square.

of the total parameter space defined by the n-cube. This is illustrated in Figure 1, where the cross explored is inscribed in the circle of center [0, 0] and radius 1. In this 2-dimensional example, the ratio of the partially explored to the total area—i.e. the square minus the circle—is $r \approx 0{,}78$. The problem gets quickly worse as we increase the number of dimensions. In 3 dimensions, r ≈ 0,52 and in 12 dimensions, r ≈ 0,000326 (see [46] for those calculations, as well as critiques and alternatives to OAT).

Fine-tuning arguments typically vary one parameter at a time. So, they use the OAT method to explore the space of alternative universes by varying one by one some of the 31 fundamental physics and cosmic parameters. They actually

explore only r $\approx 4,56.10^{-15}$ of the parameter space. We conclude that such fine-tuning arguments have restricted their exploration to 0,00000000000000456 % of the relevant parameter space![1] Can we hope to explore more of this space? How can we proceed?

Let us first call a *fecund universe* a universe generating at least as much complexity as our own. *Are fecund universes rare or common in the multiverse?* This is the core issue of fine-tuning. To answer it demands to explore this virtual multiverse. Milan Ćirković [13] and I both converged on this conclusion. Ćirković used the metaphor of sailing the archipelago of possible universes; I proposed to perform simulations of possible universes, an endeavor called *Artificial Cosmogenesis* (or ACosm, see [62];[64]; and also [60]; [61] for critiques; and [66] for replies). Such simulations would enable us not only to understand our own universe (with "real-world modelling", or processes-as-we-know-them) but also other *possible* universes (with "artificial-world modelling", or processes-as-they-could-be). We thus need to develop methods, concepts and simulation tools to explore the space of possible universes (the "cosmic landscape" as Leonard Susskind [55] calls it in the framework of string theory). In [62], I proposed to call this new field of research *Artificial Cosmogenesis* because it sets forth a "general cosmology", in analogy with Artificial Life (ALife) which appeared with the help of computer simulations to enquiry about a "general biology". However, recent work on the EvoGrid[2] simulation project suggests that the growth of complexity is more likely to remain open-ended if stochastic, non-deterministic processing is used at the bottom, instead of deterministic rules, like in ALife.

Now that we have a framework to define possible universes, we will need to generalize the "Cosmic Evolution Equation" we used to assess the robustness of our universe to explore not only our universe m^*, but also all universes m element of the wider class of possible universes M. This constitutes a rigorous approach to assess how fine-tuned our universe is. However, it is important to understand that the results of such studies would not *ipso facto* provide an *explanation* of fine-tuning. Only if it turns out that our kind of complex universe is common, then an explanation of fine-tuning would be a principle of *fecundity*: "there is no fine-tuning, because intelligent life of some form will emerge under extremely varied circumstances" ([57], p4).

Most fine-tuning arguments change just one parameter at a time and conclude that the resulting universe is not fit for developing complexity. This leads to the "one-factor-at-a-time" paradox. What if we would change *several* parameters at the same time? Systematically exploring the multiple variation of parameters seems like a very cumbersome enterprise. As Gribbin and Rees wrote ([26], p269):

[1] I used the formulae in ([46], 1510) for this calculation. Note that this assumes that we can put upper and lower boundaries on each of the parameters, which is not at all warranted for physics and cosmic parameters. Note also that this is a very generous estimate, since the actual exploration of OAT will only be a tiny n-cross within the volume of the n-sphere, which itself represents only $4,56.10^{-15}$ of the full parameter space defined by the n-cube.

[2] http://www.evogrid.org

> If we modify the value of one of the fundamental constants, something invariably goes wrong, leading to a universe that is inhospitable to life as we know it. When we adjust a second constant in an attempt to fix the problem(s), the result, generally, is to create three new problems for every one that we "solve". The conditions in our universe really do seem to be uniquely suitable for life forms like ourselves, and perhaps even for any form of organic complexity.

Back in 1991, it indeed seemed very difficult to explore and find alternative universe. However, a way to overcome this problem is to use *computer simulations* to test systematical modifications of parameters' values. In varying just one parameter, parameter sensitivity arguments have only begun to explore possible universes, like a baby wetting his toes for the first time on the seashore. Surely, we had to start somewhere. But it is truly a tiny exploration. Furthermore, maybe there is a deep link between the different constants and physical laws, such that it makes no sense to change just one parameter at a time. Changing a parameter would automatically perturb other parameters (see [11], p1581). Fortunately, more recent research have gone much further than these one-parameter variations.

What happens when we vary multiple parameters? Let us first generalize the Cosmic Evolution Equation, which this time includes other possible cosmic *evolutions*—notice the plural! Let us imagine that we run multiple times simulations of different models of universes m. We interpret the results of the multiple runs of the simulations as a set of *virtual* universes. We end up with a distribution function $f(m)$ combining the probability distributions obtained for each factor of the CEE. Let us mention that, based on modern developments in computer science, there is another more theoretical way to study and choose distribution functions for possible universes (see the remarkable study of Schmidhuber [48]).

The *possibility space* is the huge M resulting from the definition of possible universes; and we add a measure π on M. The resulting ensemble of simulated universes E would thus be defined as:

$$E = \{M, \pi, f(m)\}$$

The number of planets with intelligent life would then be:

$$N_{life}(m) = \int N_g \cdot \ N_S \cdot \ f_S \cdot \ f_p \cdot \ n_e \cdot \ f_l \cdot \ f_i \cdot \pi$$

We are now talking about cosmic outcomes in other universes. The topic becomes quite speculative, because it is not clear at all *which* cosmic outcomes are the most relevant to assess. The factors in the equation above might be totally irrelevant. What if other possible universes do not generate objects like galaxies, stars and planets, but completely different kinds of complex structures? Nothing *that we know* may evolve anymore... but other things might! We now see the fundamental importance to define cosmic outcomes and the emergence of complexity in a very general manner, so they can also apply to other possible universes. Bradford [11] proposed such a framework when he wrote about

sequences of entropy reduction. Aunger's [3] systems theoretical approach in terms of energy innovation, organization and control is also a higher-level approach. Valentin Turchin [58] also proposed a cybernetic theory of complexity transitions with the central concept of *metasystem transition*. Theoretical computer science measures such as *algorithmic complexity* (see e.g. [22]) or *logical depth* [8] are also precious tools to assess the complexity of systems in a universal manner. But these are just a few examples of frameworks to tackle the general, fascinating and fundamental problems of the evolution and measure of complexity (see also [7] for a discussion in the context of Artificial Life).

We already saw that higher outcomes $f_l \cdot f_i$ are harder to assess. This is precisely where computer simulations can be very helpful. Typically, there are so many local interactions in the evolution of complex organisms that it is hard to analyze them analytically with a deterministic science approach. For example, there is not one single equation which allows to predict the development of an embryo.

Let us now outline some remarkable alternative complex universes that researchers recently studied. Gordon McCabe studied variations on the standard model of particles, by changing the geometrical structure of space-time. The result is not the end of any complexity, but just the beginning of a new set of elementary particles. McCabe ([38], 2:38) elaborates:

> Universes of a different dimension and/or geometrical signature, will possess a different local symmetry group, and will therefore possess different sets of possible elementary particles. Moreover, even universes of the same dimension and geometrical signature will not necessarily possess the same sets of possible particles. To reiterate, the dimension and geometrical signature merely determines the largest possible local symmetry group, and universes with different gauge fields, and different couplings between the gauge fields and matter fields, will possess different local symmetry groups, and, perforce, will possess different sets of possible particles.

It thus seems that we can vary basic physics parameters without compromising all kinds of cosmic evolution. Who knows what kind of complexity can emerge from this new set of particles?

As an illustration of their framework to define the multiverse, Ellis, Kirchner and Stoeger [23] did examine some parameter variations in Friedmann-Lemaître-Robertson-Walker (FLRW)

models. They found life-allowing regions in a phase space described by the evolution of FLRW models. The fact that they found *regions* and not a *single point* in the phase space shows that there is room for some variation. So it seems that we can vary fundamental geometrical cosmological parameters without precluding the apparition of life.

Harnik, Kribs and Perez [28] constructed a universe without electroweak interactions called the Weakless Universe. They show that by adjusting standard model and cosmological parameters, they are able to obtain:

> a universe that is remarkably similar to our own. This "Weakless Universe" has big-bang nucleosynthesis, structure formation, star formation, stellar burning with a wide range of timescales, stellar nucleosynthesis up to iron and slightly beyond, and mechanisms to disperse heavy elements through type Ia supernovae and stellar mergers.

This is a truly remarkable result because the cosmic outcomes are numerous, relatively high and non trivial. Three factors in the CEE are addressed more or less directly: $N_g \cdot N_S \cdot f_S$. Maybe strong living creatures could live in the weakless universe? This remains to be investigated.

Anthony Aguirre [2] did study a class of cosmological models "in which some or all of the cosmological parameters differ by orders of magnitude from the values they assume in the standard hot big-bang cosmology, without precluding in any obvious way the existence of intelligent life." This study also shows that it is possible to vary parameters widely without obviously harming the emergence of complexity as we know it.

Robert Jaffe, Alejandro Jenkins and Itamar Kimchi [31] pursued a detailed study of possible universes with modified quark masses. They define *congenial* worlds the ones in which the quark masses allow organic chemistry. Again, they found comfortable regions of congeniality.

Fred C. Adams [1] has conducted a parametric survey of stellar stability. He found that a wide region of the parameter space provides stellar objects with nuclear fusion. He concludes that the "set of parameters necessary to support stars are not particularly rare."

An early attempt to explore alternative universes with simulations has been proposed by Victor Stenger [51, 52]. He has performed a remarkable simulation of possible universes. He considers four fundamental constants, the strength of electromagnetism α; the strong nuclear force α_s, and the masses of the electron and the proton. He then analysed "100 universes in which the values of the four parameters were generated randomly from a range five orders of magnitude above to five orders of magnitude below their values in our universe, that is, over a total range of ten orders of magnitude" [52]. The distribution of stellar lifetimes in those universes shows that most universes have stars that live long enough to allow stellar evolution and heavy elements nucleosynthesis. Stenger's initial motivation was to refute fine-tuning arguments, which is why he ironically baptised his simulation "MonkeyGod". The implicit idea is that even a stupid monkey playing with cosmic parameters can create as much complexity as God.

In conclusion, other possible universes are also fine-tuned for some sort of complexity! Those remarkable studies show consistently that *alternative complex universes are possible*. One might object that such explorations do not yet assess the higher complexity factors in the CEE. They do not answer the following key questions: would other interesting complex structures like planetary systems, life, intelligence or technology evolve in those other universes? However, these are only early attempts in conceptualizing and simulating other possible universes, and the enterprise is certainly worth pursuing. The fine-tuning issue could then be seriously tackled, because we would know more and more precisely

the likelihood of having our universe as it is, by comparing it to other possible universes. Such pioneering studies are just a beginning, and certainly new studies will come up with more and more complex alternative universes.

6 Summary

Let us now summarize the three main steps necessary to assess how fine-tuned our universe is.

(1) *Define* a space M of possible universes
(2) *Explore* this space
(3) *Assess* the place of our universe in M

Let us review step (1). Our analysis of the historical trends regarding free parameters [64] invites us to start with a *weak variation*, i.e. varying free parameters in physical and cosmological models. Why not vary the laws of physics themselves? It seems a very cumbersome enterprise, because we do not even know how to make them vary (see [59]). It can also be dubious to do so, since the distinction between laws and initial or boundary conditions is fuzzy in cosmology [21].

This suggestion to focus on weak variation makes most sense for the following reasons. First, it is concrete and operational, and has a clear meaning with well established physics. Second, we assume supernatural miracles happening in the middle of cosmic evolution to be—by definition—impossible. We assume there is a consistency and continuity in cosmic evolution. We hypothesize that higher level parameters are ultimately reducible to these physics and cosmic ones. The emergent higher levels occur naturalistically. Of course, this remains to be shown, and for practical purposes we might assume as given such higher level parameters in our studies and simulations. New levels of emergence, new levels of complexity did historically emerge from lower levels, even if complicated top-down causation occurs (see e.g. [22]). Take for example an economic law like the law of supply and demand. It did not and could not exist before the apparition of organized human civilizations. It emerged out of such new organizations. It seems that what we call "natural laws" are simply the result of more and more regular interactions. For example, as the universe cools down, new organizations emerge. Again, it is clear that a few billion years ago, there was no economic laws.

We also need to be more specific to apply probabilities to the ensemble of possible universes, and avoid probabilistic fallacies. For example, we must decide, arbitrarily or not, parameter's upper and lower bounds. This is necessary for all practical purposes, because we can not explore the parameter space of all parameters varying from $-\infty$ to $+\infty$. We thus need to define the maximum deviation allowed for each parameter.

We must beware of one-factor-at-a-time limitations and paradox. We must also define a probability measure on the parameter space. I refer the reader to [33] and [23] for detailed arguments that measure-theoretical grounds can be specified to assess fine-tuning. It is also crucial to define *cosmic outcomes*

to specify the object of fine-tuning we aim to address. Do we talk about fine-tuning for nucleosynthesis? atoms? Stars? Life? Intelligence? Or a more general complexity emergence?

Step (2) requires to explore this space. The simplest exploration is to re-run the tape of *our* universe. But this only tackles the issue of the *robustness* of the universe. If we want to address the fine-tuning issue we must also run and re-run tapes of *other possible universes*. This will bring us insights into how our and other universes are parameter sensitive, and generate complex outcomes. Although we always need good theoretical models to start with, it is necessary to use computer simulations to explore the huge parameter landscape we are talking about. That landscape is not just very big, but really huge. Because we don't want to and do not have the resources to explore the space blindly, it also makes most sense to use simulations to test particular hypotheses and theories. As an application, if we take Lee Smolin's [49] cosmological natural selection theory, and find alternative universes with more black holes (the cosmic outcome under consideration) by tweaking parameters, it is a way to falsify the theory.

The last step (3) is to compare the distribution functions of the cosmic outcomes obtained through simulations, to the space M of possible universes. In other words, we assess the probability to find a universe with outcome O. Note that this is the crucial difference between tackling the robustness and the fine-tuning issue. In robustness analysis, we run multiple times the *same* universe simulation changing only the random dynamical parameters. We compare multiple runs of the same universe. In fine-tuning analysis, we run multiple *different* universe simulations, changing a wide number of parameters. We compare our universe to the set of possible universes. How typical or atypical is our universe in the space of possible universes? The results of such simulation experiments will enable us to answer this question. Ideally, we will be in a position to assess the likelihood or unlikelihood of complexity emergence in the space of possible universes. Even better than assessing specific cosmic outcomes, which might bias us to a universe-centric perspective, we can aim to assess the probability to find universes which display open-ended evolutionary mechanisms leading to ever increasingly complex cosmic outcomes.

To the traditionally trained cosmologist, this enterprise might seem totally unconventional. And it is, because it is a new kind of computational science. This is why we can call it *Artificial Cosmogenesis*. It might also seem out of reach. As I argued elsewhere, since the sheer computational resources grow more than exponentially, this allows us in principle to increase accordingly the complexity and richness of our computer simulations [62].

Additionally, engineers and professional model makers have developed a wide variety of tools to test multiple variables, rarely used in cosmological contexts. Let us just mention of few of them. A starting point is to use the tools of global sensitivity analysis (see e.g. [45]).

These include advanced statistical approaches such as latin hypercube sampling, multivariate stratified sampling or Montecarlo simulations for finding dynamic confidence intervals. Systems dynamics and engineering have also many

tools to offer such as phase portraits or probabilistic designs. The classic book by John D. Sterman [54] remains a reference and quite comprehensive introductory book on complex systems modeling and simulations.

Let us now be scrupulous. What is a proof of fine-tuning? Let n be the number of free parameters. We have a logical and statistical version of what a proof of fine-tuning would be:

> **Logical proof of fine-tuning:** *If you vary one parameter, there exists no possible universe generating outcome O by adjusting the* $(n-1)$ *other parameters.*
> Which is equivalent to:
> *if you vary one parameter, there is no way whatsoever that all other possible universes can generate outcome O.*

> **Probabilistic proof of fine-tuning:** *If you vary one parameter, adjusting the* $(n-1)$ *other parameters will not make outcome O more likely.*
> Which is equivalent to:
> *if you vary one parameter, there is no way whatsoever that all other possible universes can generate outcome O with a higher probability.*

In sum, you need to have explored the relevant parameter space of possible universes to make serious claims about fine-tuning. Pretty hard to prove! This is even harder for outcomes as advanced as life or intelligence.

Our conclusion is that *fine-tuning for life or intelligence remains a conjecture.* Like in mathematics, we have strong reasons to believe the conjecture is true, but a proof is out of reach and certainly requires a huge amount of work. As a matter of fact, the challenge of simulating possible universes and comparing them is overwhelming. This is why the concept of the cosmic outcome is so important to ease the process. Indeed, we can break down the problem and progress by tackling higher and higher outcomes, with more and more connection between outcomes. We don't need nor can assess all outcomes at once in the CEE. As our understanding, modeling capacities and computational resources increase, we can be more ambitious in simulating more and more as well as higher and higher outcomes in cosmic evolution. I am well aware of the highly ambitious research program that ACosm proposes. However, the good news is that there is work for many generations of scientists. Tomorrow's cosmology is not restricted to empirical observations or highly theoretical models. It is also the science of simulating and experimenting with alternative universes.

7 Conclusion

Up to now, discussions about possible universes were chiefly a metaphysical recreation. We advanced conceptual foundations to study possible universes scientifically, with the help of computer simulations. This approach is needed if we take seriously the thesis of computational irreducibility, namely that most complex systems are theoretically impossible to predict in a deterministic or

statistical manner. A more general computational kind of science is needed. We applied this new kind of science to cosmology, to address two key cosmological issues: the robustness of the emergence of complexity, and the fine-tuning of the universe.

We first formulated the issues of defining possible universes, and possible cosmic outcomes (sections 2 and 3).

Based on previous work, we defined a modular "Cosmic Evolution Equation" (CEE). This equation can have many applications to define research agendas in computational cosmology. In particular, to tackle our two issues, we adjusted the CEE by varying the space of possible universes it acts upon, to study either the robustness (section 4) or the fine-tuning issue (5).

Importantly, we considered only a *virtual multiverse*, that we define within our concrete models and simulations. This is in sharp contrast with speculations about an actual multiverse, an idea quite common in modern cosmology, yet often criticized for being hard or impossible to test scientifically.

To address the delicate fine-tuning issue, we further argued that studies and simulations of alternative possible universes are demanded, a research field called *Artificial Cosmogenesis* (ACosm, sections 5-6). This field is actually not new, since we outlined quite some research which have examined alternative possible universes. Yet these studies are really just beginning to explore possible universes, and ACosm holds great promise to further investigate whether and how our universe and others generate increasing complexity.

Acknowledgments. I thank Rick Bradford, Bruce Damer, Jean-Paul Delahaye, Francis Heylighen, Tomás Igor Veloz González, Stanley Salthe, and William Stoeger for thoughtful comments and criticisms.

A Appendix - Argumentative Maps

Fig. 2 maps the problem described in introduction, while Fig. 3 maps the core argument presented in the paper. Please read in a top-down direction. More details on argumentation mapping can be found in [62].

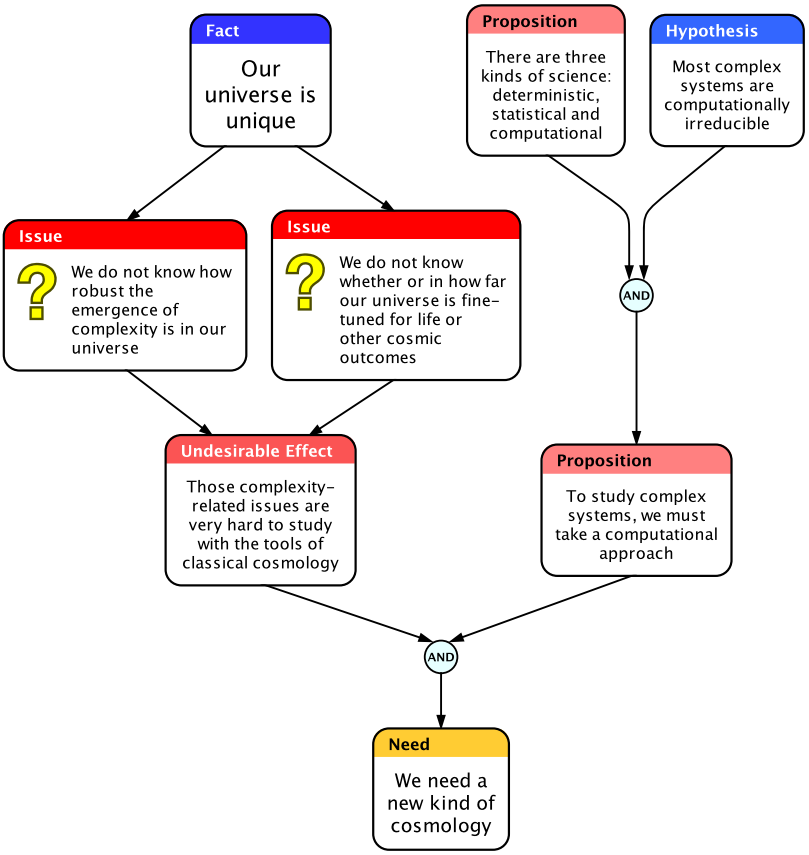


Fig. 2. The core problem

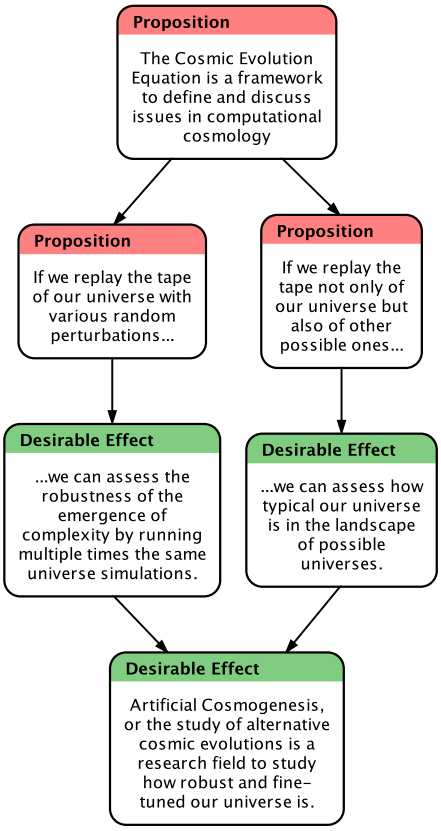


Fig. 3. The proposed solution

References

[1] Adams, F.C.: Stars in Other Universes: Stellar Structure with Different Fundamental Constants. Journal of Cosmology and Astroparticle Physics (08) (August 7, 2008), http://arxiv.org/abs/0807.3697, doi:10.1088/1475-7516

[2] Aguirre, A.: Cold Big-Bang Cosmology as a Counterexample to Several Anthropic Arguments. Physical Review D 64(8), 83508 (2001), http://arxiv.org/abs/astro-ph/0106143

[3] Aunger, R.: Major Transitions in 'Big' History. Technological Forecasting and Social Change 74(8), 1137–1163 (2007a), doi:10.1016/j.techfore.2007.01.006

[4] Aunger, R.: A Rigorous Periodization of 'Big' History. Technological Forecasting and Social Change 74(8), 1164–1178 (2007b), doi:10.1016/j.techfore.2007.01.007

[5] Barrow, J.D., Morris, S.C., Freeland, S., Harper, C.: Fitness of the Cosmos for Life: Biochemistry and Fine-Tuning. Cambridge University Press (2008)

[6] Barrow, J.D., Tipler, F.J.: The Anthropic Cosmological Principle. Oxford University Press (1986)

[7] Bedau, M.A.: The Evolution of Complexity. In: Mapping the Future of Biology. Studies in the Philosophy of Science, vol. 266, pp. 111–130. Springer, Netherlands (2009)
[8] Bennett, C.H.: Logical Depth and Physical Complexity. In: Herken, R. (ed.) The Universal Turing Machine: A Half-Century Survey, pp. 227–257. Oxford University Press (1988), http://www.research.ibm.com/people/b/bennetc/UTMX.pdf
[9] Boylan-Kolchin, M., Springel, V., White, S.D.M., Jenkins, A., Lemson, G.: Resolving Cosmic Structure Formation with the Millennium-II Simulation. Monthly Notices of the Royal Astronomical Society 398(3), 1150–1164 (2009), http://arxiv.org/abs/0903.3041, doi:10.1111/j.1365-2966.2009.15191.x
[10] Bradbury, R.J., Ćirković, M.M., Dvorsky, G.: Dysonian Approach to Seti: A Fruitful Middle Ground? Journal of the British Interplanetary Society 64, 156–165 (2011)
[11] Bradford, R.A.W.: The Inevitability of Fine Tuning in a Complex Universe. International Journal of Theoretical Physics 50(5), 1577–1601 (2011), http://rickbradford.co.uk/InevitabilityofFineTuningJun.pdf, doi:10.1007/s10773-011-0669-2
[12] Chaitin, G.J., Calude, C., Casti, J., Davies, P.C.W., Svozil, K., Wolfram, S.: Is the Universe Random? In: Zenil, H. (ed.) Randomness Through Computation: Some Answers, More Questions, pp. 311–352. World Scientific Publishing Company (2011)
[13] Cirkovic, M.M.: Sailing the Archipelago. In: Veal, D. (ed.) Collapse: V. 5: Philosophical Research and Development - The Copernican Imperative. Urbanomic (2009), http://ieet.org/archive/archipelago.pdf
[14] Colyvan, M., Garfield, J.L., Priest, G.: Problems With the Argument From Fine Tuning. Synthese 145(3), 325–338 (2005), doi:10.1007/s11229-005-6195-0
[15] Davies, P.C.W.: Our Place in the Universe. In: Leslie, J. (ed.) Modern Cosmology & Philosophy, pp. 311–318. Prometheus Books, Amherst (1998)
[16] Davies, P.C.W.: The Goldilocks Engima: Why Is the Universe Just Right for Life? Mariner Books (2008)
[17] Dawkins, R.: The God Delusion. 1st Mariner Books ed. Houghton Mifflin Co., Boston (2008)
[18] Dick, S.J.: Plurality of Worlds: The Extraterrestrial Life Debate from Democritus to Kant. New Ed. Cambridge University Press (1982)
[19] Drake, F.: The Radio Search for Intelligent Extraterrestrial Life. In: Mamikunian, G., Briggs, M.H. (eds.) Current Aspects of Exobiology, pp. 323–345. Pergamon, New York (1965)
[20] Ellis, G.F.R.: Multiverses: Description, Uniqueness and Testing. In: Carr, B. (ed.) Universe or Multiverse? pp. 387–410. Cambridge University Press, Cambridge (2007a)
[21] Ellis, G.F.R.: Issues in the Philosophy of Cosmology. In: Butterfield, J., Earman, J. (eds.) Handbook in Philosophy of Physics, pp. 1183–1285. Elsevier (2007), http://arxiv.org/abs/astro-ph/0602280
[22] Ellis, G.F.R.: On the Nature of Causation in Complex Systems. Transactions of the Royal Society of South Africa 63(1), 69–84 (2008), http://www.mth.uct.ac.za/~{}ellis/Top-down%20Ellis.pdf, doi:10.1080/00359190809519211
[23] Ellis, G.F.R., Kirchner, U., Stoeger, W.R.: Multiverses and Physical Cosmology. Monthly Notices of the Royal Astronomical Society 347(3), 921–936 (2004), http://arxiv.org/abs/astro-ph/0407329, doi:10.1111/j.1365-2966.2004.07261.x

[24] Freivogel, B., Kleban, M., Martnez, M.R., Susskind, L.: Observational Consequences of a Landscape. Journal of High Energy Physics (03), 039–039 (March 9, 2006), doi:10.1088/1126-6708/2006/03/039
[25] Gould, S.J.: Wonderful Life: The Burgess Shale and the Nature of History. WW Norton & Company (1990)
[26] Gribbin, J., Rees, M.J.: Cosmic Coincidences, Dark Matter, Mankind, and Anthropic Cosmology. Black Swan (1991)
[27] Guo, Q., White, S., Boylan-Kolchin, M., De Lucia, G., Kauffmann, G., Lemson, G., Li, C., Springel, V., Weinmann, S.: From Dwarf Spheroidals to cD Galaxies: Simulating the Galaxy Population in a ΛCDM Cosmology. Monthly Notices of the Royal Astronomical Society 413, 101–131 (2011), http://arxiv.org/abs/1006.0106
[28] Harnik, R., Kribs, G.D., Perez, G.: A Universe Without Weak Interactions. Physical Review D 74(3), 035006 (2006), http://arxiv.org/abs/hep-ph/0604027v1, doi:10.1103/PhysRevD.74.035006
[29] Hawking, S.W.: The Universe in a Nutshell. Bantam Books (2001)
[30] Hoyle, F., Dunbar, D.N.F., Wenzel, W.A., Whaling, W.: A State in C12 Predicted from Astrophysical Evidence. Physical Review 92, 1095 (1953)
[31] Jaffe, R.L., Jenkins, A., Kimchi, I.: Quark Masses: An Environmental Impact Statement. 0809.1647 (September 10, 2008), http://arxiv.org/abs/0809.1647, doi:10.1103/PhysRevD.79.065014
[32] Kleijnen, J.P.C., Sanchez, S.M., Lucas, T.W., Cioppa, T.M.: A User's Guide to the Brave New World of Designing Simulation Experiments. INFORMS Journal on Computing 17(3), 263–289 (2005), http://www.tilburguniversity.edu/research/institutes-and-research-groups/center/staff/kleijnen/informs_joc.pdf
[33] Koperski, J.: Should We Care About Fine-Tuning? The British Journal for the Philosophy of Science 56(2), 303–319 (2005), doi:10.1093/bjps/axi118
[34] Leibniz, G.W.: 1710. Essais De Théodicée Sur La Bonté De Dieu, La Liberté De L'homme Et L'origine Du Mal. Garnier-Flammarion, Paris (1969)
[35] Leslie, J.: Universes. Routledge (1989)
[36] Lewis, D.K.: On the Plurality of Worlds. B. Blackwell, Oxford (1986)
[37] Li, M., Vitányi, P.M.B.: An Introduction to Kolmogorov Complexity and Its Applications. Springer (1997)
[38] McCabe, G.: The Structure and Interpretation of the Standard Model, 1st edn., vol. 2. Elsevier Science (2007)
[39] McGrew, T., McGrew, L., Vestrup, E.: Probabilities and the Fine-Tuning Argument. Mind 110(440) (2001), http://commonsenseatheism.com/wp-content/uploads/2010/05/McGrew-Mcgrew-Vestrup-Probabilities-and-the-Fine-Tuning-Argument-a-skeptical-view.pdf
[40] Miller, S.L.: A Production of Amino Acids Under Possible Primitive Earth Conditions. Science 117(3046), 528–529 (1953), doi:10.1126/science.117.3046.528
[41] Nottale, L.: Fractal Space-Time and Microphysics: Towards a Theory of Scale Relativity. World Scientific (1993)
[42] Nottale, L.: Scale Relativity and Fractal Space-Time: Theory and Applications. Foundations of Science 15(2), 101–152 (2010), http://www.arxiv.org/abs/0912.5508, doi:10.1007/s10699-010-9170-2
[43] Nottale, L.: Scale Relativity and Fractal Space-Time: A New Approach to Unifying Relativity and Quantum Mechanics. World Scientific Publishing Company (2011)
[44] Rees, M.: Just Six Numbers: The Deep Forces That Shape the Universe. Weidenfeld and Nicholson, London (1999)

[45] Saltelli, A., Ratto, M., Andres, T.: Global Sensitivity Analysis: The Primer. John Wiley & Sons (2008)
[46] Saltelli, A., Annoni, P.: How to Avoid a Perfunctory Sensitivity Analysis. Environmental Modelling & Software 25(12), 1508–1517 (2010), doi:10.1016/j.envsoft.2010.04.012
[47] Salthe, S.N.: The Natural Philosophy of Entropy. Seed 2(2) (2002)
[48] Schmidhuber, J.: Algorithmic Theories of Everything. Technical Report IDSIA-20-00, Lugano, Switzerland (2000), http://arxiv.org/abs/quant-ph/0011122
[49] Smolin, L.: Did the Universe Evolve? Classical and Quantum Gravity 9(1), 173–191 (1992)
[50] Springel, V., White, S.D.M., Jenkins, A., Frenk, C.S., Yoshida, N., Gao, L., Navarro, J., Thacker, R., Croton, D., Helly, J.: Simulations of the Formation, Evolution and Clustering of Galaxies and Quasars. Nature 435, 629–636 (2005), http://astronomy.sussex.ac.uk/~{}petert/archive/svirgo05.pdf.
[51] Stenger, V.J.: The Unconscious Quantum Metaphysics in Modern Physics and Cosmology. Prometheus Books, Amherst (1995)
[52] Stenger, V.J.: Natural Explanations for the Anthropic Coincidences. Philo. 3(2), 50–67 (2000)
[53] Stenger, V.J.: The Fallacy of Fine-Tuning: Why the Universe Is Not Designed for Us. Prometheus Books (2011)
[54] Sterman, J.D.: Business Dynamics: Systems Thinking and Modeling for a Complex World. McGraw-Hill, Irwin (2000)
[55] Susskind, L.: The Cosmic Landscape: String Theory and the Illusion of Intelligent Design. Little Brown, New York (2005)
[56] Susskind, L.: The Anthropic Landscape of String Theory. In: Carr, B. (ed.) Universe or Multiverse? pp. 247–266. Cambridge University Press, Cambridge (2007), http://arxiv.org/abs/hep-th/0302219
[57] Tegmark, M., Aguirre, A., Rees, M.J., Wilczek, F.: Dimensionless Constants, Cosmology, and Other Dark Matters. Physical Review D 73(2), 23505 (2006)
[58] Turchin, V.F.: The Phenomenon of Science. Columbia University Press, New York (1977), http://pespmc1.vub.ac.be/POS/TurPOS.pdf
[59] Vaas, R.: Is There a Darwinian Evolution of the Cosmos? - Some Comments on Lee Smolin's Theory of the Origin of Universes by Means of Natural Selection. In: MicroCosmos - MacroCosmos Conference, Aachen, Germany, September 2-5 (1998), http://arxiv.org/abs/gr-qc/0205119
[60] Vaas, R.: Life, the Universe, and Almost Everything: Signs of Cosmic Design (2010), http://arxiv.org/abs/0910.5579
[61] Vaas, R.: Cosmological Artificial Selection: Creation Out of Something? Foundations of Science 17(1), 25–28 (2012), http://arxiv.org/abs/0912.5508, doi:10.1007/s10699-010-9218-3
[62] Vidal, C.: The Future of Scientific Simulations: From Artificial Life to Artificial Cosmogenesis. In: Tandy, C. (ed.) Death and Anti-Death. Thirty Years After Kurt Gödel (1906-1978), vol. 6, pp. 285–318. Ria University Press (2008), http://arxiv.org/abs/0803.1087
[63] Vidal, C.: Introduction to the Special Issue on the Evolution and Development of the Universe. Foundations of Science 15(2), 95–99 (2010a), http://www.arxiv.org/abs/0912.5508, doi:10.1007/s10699-010-9176-9
[64] Vidal, C.: Computational and Biological Analogies for Understanding Fine-Tuned Parameters in Physics. Foundations of Science 15(4), 375–393 (2010b), http://arxiv.org/abs/0912.5508, doi:10.1007/s10699-010-9183-x

[65] Vidal, C.: Black Holes: Attractors for Intelligence? Presented at the Kavli Royal Society International Centre. Towards a scientific and societal agenda on extra-terrestrial life, October 4-5 (2010), http://arxiv.org/abs/1104.4362
[66] Vidal, C.: Fine-tuning, Quantum Mechanics and Cosmological Artificial Selection. Foundations of Science 17(1), 29–38 (2012a), http://arxiv.org/abs/0912.5508, doi:10.1007/s10699-010-9219-2
[67] Vidal, C.: The Beginning and the End: The Meaning of Life in a Cosmological Perspective (PhD Thesis, to appear). Vrije Universiteit Brussel, Brussels (2012b)
[68] Wolfram, S.: Undecidability and Intractability in Theoretical Physics. Physical Review Letters 54(8), 735–738 (1985), doi:10.1103/PhysRevLett.54.735
[69] Wolfram, S.: A New Kind of Science. Wolfram Media Inc., Champaign (2002)
[70] Zenil, H., Delahaye, J.-P.: On the Algorithmic Nature of the World. In: Burgin, M., Dodig-Crnkovic, G. (eds.) Information and Computation. World Scientific (2010), http://arxiv.org/abs/0906.3554
[71] Zuse, K.: Calculating Space. Reedited (Project MAC MIT Translation). In: Zenil, H.A. (ed.) Computable Universe: Understanding Computation & Exploring Nature As Computation. World Scientific (1970/2012), ftp://ftp.idsia.ch/pub/juergen/zuserechnenderraum.pdf
[72] Zwirn, H., Delahaye, J.-P.: Unpredictability and Computational Irreducibility. In: Zenil, H. (ed.) Irreducibility & Computational Equivalence Ten Years After Wolfram's a New Kind of Science (2012)

Part V

The Behavior of Systems and the Notion of Computation

Chapter 14
An Incompleteness Theorem for the Natural World

Rudy Rucker

Department of Computer Science
San Jose State University, USA
rudy@rudyrucker.com

1 Introduction

The philosopher Gottfried Wilhelm von Leibniz is perhaps best known for the fierce controversy that arose between him and Sir Isaac Newton over the invention of calculus. The S-like integral sign that we use to this day is in fact a notation invented by Leibniz.

When Leibniz was a youth of nineteen, he wrote a paper called "De Arte Combinatorica", in which he tried to formulate a universal algebra for reasoning, in the hope that human thought might some day be reducible to mathematical calculations, with symbols or characters standing for thoughts.

> But to return to the expression of thoughts by means of characters, I thus think that controversies can never be resolved, nor sectarian disputes be silenced, unless we renounce complicated chains of reasoning in favor of simple calculations, and vague terms of uncertain meaning in favor of determinate characters. In other words, it must be brought about that every fallacy becomes nothing other than a calculating error, and every sophism expressed in this new type of notation becomes in fact nothing other than a grammatical or linguistic error, easily proved to be such by the very laws of this philosophical grammar.
>
> Once this has been achieved, when controversies arise, there will be no more need for a disputation between two philosophers than there would be between two accountants. It would be enough for them to pick up their pens and sit at their abacuses, and say to each other (perhaps having summoned a mutual friend): "Let us calculate."[1]

Let's refer to this notion as Leibniz's dream – the dream of finding a logical system to decide all of the things that people might ever disagree about. Could the dream ever work?

[1] The quote is from Leibniz and Gerhardt (1978), volume VII, p. 200. The passage is translated by the British philosopher George MacDonald Ross and can be found on his website.

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 185–198.
DOI: 10.1007/978-3-642-35482-3_14

Even if the dream were theoretically possible (which it isn't), as a practical matter it wouldn't work anyway. If a universal algebra for reasoning had come into existence, would, for instance, Leibniz have been able to avoid his big arguments with Newton? Not likely. People don't actually care all that much about logic, not even Leibniz. We just pretend to like logic when it happens to be on our side – otherwise we very often abandon logic and turn to emotional appeals.

This said, there's a powerful attraction to Leibniz's dream. People like the idea of finding an ultimate set of rules to decide everything. Physicists, for instance, dream of a Theory of Everything. At a less exalted level, newspapers and TV are filled with miracle diets – simple rules for regulating your weight as easily as turning a knob on a radio. On the ethical front, each religion has its own compact set of central teachings. And books meant to help their readers lead happier lives offer a simple list of rules to follow.

But, as I hinted above, achieving Leibniz's dream is logically impossible.

In order to truly refute Leibniz's dream, we need to find a precise way to formulate it. As it happens, formal versions of Leibniz's dream were first developed early in the Twentieth century.

An early milestone occurred in 1910, when the philosophers Bertrand Russell and Alfred North Whitehead published their monumental *Principia Mathematica*, intended to provide a formal logical system that could account for all of mathematics. And, as we'll be discussing below, hand in hand with the notion of a formal system came an exact description of what is meant by a logical proof. There were some problems with the Russell-Whitehead system, but by 1920, the mathematician David Hilbert was confident enough to propose what came to be known as *Hilbert's program*.

1. We will discover a complete formal system, capable of deciding all the questions of mathematics.
2. We will prove that this system is free of any possible contradiction.

As Hilbert put it, "The conviction of the solvability of every mathematical problem is a powerful incentive to the worker. We hear within us the perpetual call: There is the problem. Seek its solution. You can find it by pure reason, for in mathematics there is no *ignorabimus*."

For a decade, scientists could dream that Hilbert's program might come true. And meanwhile mathematics and much of physics were being recast as formal systems. Scientific theories could now be viewed as deterministic processes for determining the truth of theorems. Leibniz's dream was nearly at hand! But, then, in 1931, the logician Kurt Gödel proved his celebrated Incompleteness Theorem.

Gödel's Incompleteness Theorem. If F is a consistent formal system as powerful as arithmetic, then there are infinitely many sentences which are undecidable for F.

This means there can never be formal system of mathematics of the kind sought by Hilbert's program. Every formal system F about mathematics is incomplete in the sense that there are sentences G such that F fails to prove G or $\sim G$, where $\sim G$ is the negation of G.

Gödel's sentences G take the form of statements that certain algebraic formulas have no solutions in the natural numbers. Normally these sentences include at least one very large numerical parameter that in some sense codes up the entire theory F. And a typical Gödelian sentence G is in some sense asserting "I am not provable from the theory F." Wolfram ([11] p. 790) has suggested that there might be some much simpler undecidable Gödelian sentences, and proposes that the following sentence might be undecidable: "For all m and n, $m^2 \neq n^5 + 6n + 3$."

Philosophers of science have wondered if there is something like an Incompleteness Theorem for theories about the natural world. One somewhat awkward approach might be to argue that if the natural world happens to be infinite, then we can in some sense represent the system of natural numbers as a list of objects within the world and then go on to claim that the usual undecidable Gödel statements about arithmetic are also statements about the natural world.

But, as I discuss in ([3] p. 290), this isn't a satisfying approach. If we wanted to have number theory be a subset of a theory W about the physical world, we'd need for W to single out an infinite set of objects to play the role of the numbers, and W would also need to define relations the correspond to numerical addition and multiplication.

What we really want is a proof–or at least a plausibility argument–for a Natural Incompleteness Theorem that asserts the existence of undecidable sentences that are about natural physical processes–as opposed to being about the natural numbers in disguise.

Wolfram's analysis of computation in *A New Kind of Science* opens a path. The first step is to accept the idea that natural processes can be thought of as computations. And the second step is to argue for some form of Wolfram's Principle of Computational Equivalence.

Wolfram's Principle of Computational Equivalence (PCE): Almost all processes that are not obviously simple can be viewed as computations of equivalent sophistication.

In this essay I'll show that, starting from Wolfram's two steps, we can prove a Natural Incompleteness Theorem. My method will be to make use of Alan Turing's 1936 work on what he called unsolvable halting problems. And rather than using the full strength of Wolfram's somewhat controversial Principle of Computational Equivalence, I'll base my argument on a weaker assumption, which I call the Halting Problem Hypothesis. And we'll end up with the following Natural Incompleteness Theorem.

Natural Incompleteness Theorem. For most naturally occurring complex processes and for any correct formal system for science, there will be sentences about the process which are undecidable by the given formal system.

This is, I believe, a clean statement of new result–and may be of real importance to the philosophy of science. Although Wolfram ([11] p. 1138) gives some specific examples of undecidable statements about natural processes, he fails to state the general Natural Incompleteness Theorem.

2 The Halting Problem Hypothesis

It's traditional to ask if a computation comes to an end, or if it halts. We can extend our language a bit and speak of a natural process as halting it happens to reach or to pass through some particular designated state. The established results about the narrow sense of halting apply to this generalized sense as well.

In many situations we value processes that halt in our more general sense. Suppose you feed a set of equations into some computer algebra software, and that you ask the software to solve the equations. What you want is for the resulting process to halt in the sense of displaying an answer on the screen. It doesn't halt in the more dramatic and narrow sense of going dead or freezing up the machine.

In many situations, we like to have computations or processes that don't halt. When we simulate, say, the life of some artificially alive creature, or the evolution of a species, we aren't aiming towards a specific kind of result, and still less do we want to see a fixed state or periodic behavior. In this situation we prefer a non-halting computation that continues to produce novel effects.

The distinction between halting and not halting leads to Turing's Theorem of 1936.

Definition. The computation P is said to have a *solvable halting problem* if and only if there is an algorithm for deciding in advance which inputs will cause P eventually to reach a halted target state, and which inputs will cause P to run endlessly without ever reaching a halted target state.

Definition. A computation is *universal* if it can emulate any other computation.

Emulating a particular computation C means that you can feed a certain code into your universal computation U that will cause U to produce the same input-output behavior as C.

As it happens, universal computations are in fact very common. Any personal computer, for instance, embodies a universal computation. Indeed, even as

simple a computation as the one-dimensional cellular automaton with rule-code 110 is universal [11].

Putting all our new concepts together, we arrive at the following.

Turing's Theorem. If U is a universal computation, then U has an unsolvable halting problem.

This means that if a computation is of a sufficiently rich and general nature, then there is no simple algorithm for predicting which inputs will make U run forever, and which inputs will make U end up in some desired target state, such as the state of coming to a halt.

Let's switch focus now, and discuss how the notion of halting problems can be used to formulate a weaker form of Wolfram's Principle of Computational Equivalence. For convenience, here is a statement of the PCE again.

Wolfram's Principle of Computational Equivalence (PCE): Almost all processes that are not obviously simple can be viewed as computations of equivalent sophistication.

I'll now ring the PCE through three changes, hit a snag, formulate an alternate form of the PCE, and then suggest a still-weaker hypothesis that I'll call the Halting Problem Hypothesis (HPH).

Suppose that we speak of computations rather than processes, and that we speak of computations that are "complex" rather than "not obviously simple." In this case the PCE becomes:

(1) Almost all complex computations are of equivalent sophistication.

What might Wolfram mean by saying that two computations are "of equivalent sophistication"? Suppose we take this to mean that the computations can emulate each other or that, more technically, they have the same degree of unsolvability. So now the PCE becomes:

(2) Almost all complex computations can emulate each other.

Now certainly Turing's universal computation is complex. So, given that a computation which emulates a universal computation is itself universal, the PCE becomes:

(3) Almost all complex computations are universal.

But mathematical logicians have proved:

(Snag) *There are very many complex computations which are not universal.*

The "almost all" in the PCE gives us some wiggle room.[2] But at this point we'd do well to back off. Suppose we weaken the range of application of the PCE. Rather than saying it applies to "almost all" complex computations, suppose we say it applies to "Most naturally occurring" complex computations. And this gives us a weakened formulation of the PCE.

(4) Most naturally occurring complex computations are universal.

This statement may still be too strong. Rather than insisting upon it, let's consider what we plan to use the PCE *for*. As I mentioned in the introductory section, I plan to use something like the PCE as a stepping stone to a Natural Incompleteness Theorem. And for this, all I need is the following *Halting Problem Hypothesis (HPH)*.

(HPH) Halting Problem Hypothesis: Most naturally occurring complex computations have unsolvable halting problems relative to some simple notion of halting.

Think of a computation as an ongoing process, for example your life, or society, or a plant growing, or the weather. As I mentioned in the previous section, relative to a given computation we can formulate the notion of a *target state* as being some special status or behavior that the computation might eventually reach. The *halting problem* in this context is the problem of deciding whether a

[2] When Wolfram formulated his PCE, he was well aware of the problem that there are infinitely many degrees of unsolvability. Therefore he phrased his PCE so that it has two loopholes. ([11], p. 734 and pp. 1130-1131.) The loopholes are to be found in, respectively, the very first and very last phrases of the PCE, which I italicize here: *Almost all* processes that are not obviously simple can be viewed as computations *of equivalent sophistication*. Regarding the first loophole, Wolfram is saying that complex non-universal Turing machines "almost never" occur in natural contexts. This is an interesting aspect of the PCE, in that it seems to say something about the kinds of processes that actually occur in the real world. Keep in mind that Wolfram's work is *empirical*. Unlike physical experiments, computer science experiments are exactly reproducible, and thus have a touch of the mathematical or theoretical. But really his inspiration came from looking at exceedingly many computations in action. And to reduce experimenter bias, he made a point of conducting *exhaustive* surveys of various classes of rudimentary computations such as Turing machines and cellular automata. To exploit the second loophole we might interpret "computations of equivalent sophistication" more broadly than "computations that can emulate each other." Wolfram feels that the processes by which logicians construct complex but non-universal computations have always depended so essentially on the use of an underlying universal computation that the constructed computations are in some as-yet-to-be-defined-sense "as sophisticated as" the universal computations. Now, so far as I know, all the existing constructions of these complex non-universal computations *do* use a universal computation. But it seems capricious to conclude that therefore *every* complex non-universal computation in some way relies upon a construction involving a universal Turing machine. Indeed it seems plausible that there may in fact be naturally occurring processes of intermediate degree. It's tempting to speculate that the one-dimensional CA Rule 30 itself is such a computation. And in this case, the PCE would be false, but the HPH that I describe would be true.

given input will eventually send your computation into one of the target states. And, once again, a halting problem is *unsolvable* if there's no computation, algorithm, or rule-of-thumb to detect which inputs won't ever produce one of these specified target state.

The HPH says that if you have some naturally occurring computation that isn't obviously simple, then there will probably be some simple notion of a target state that leads to an unsolvable halting problem.

Note that the PCE implies the HPH. Going in the other direction, the HPH does *not* imply the PCE. The HPH claims only that certain computations have unsolvable halting problems, and does *not* claim that these computations are universal. The good thing about the HPH is that, unlike the PCE, the HPH has no difficulties with the many non-universal computations that have unsolvable halting problems. The HPH has a better chance of being true, and is easier to defend against those who doubt the validity of Wolfram's analysis of computation.

It's worth noting that it may be possible to drop the two-fold qualifier "mast naturally occurring" from the HPH and to get a Strong Halting Problem Hypothesis as stated below.

Strong Halting Problem Hypothesis: All complex computations have unsolvable halting problems relative to some notion of halting.

This says that *all* complex computations have associated with them some unsolvable halting problem. If this is indeed the case, then the Strong Halting Problem Hypothesis clarifies what we mean by a "complex computation."

Table 1. Unsolvable Halting Problems In Everyday Life

Computation	**Target States**	**Unsolvable Halting Problem**
The motions of the bodies in our solar system.	Something rams into Earth.	Which possible adjustments to Earth's orbit can make us safe forever?
The evolution of our species as we spread from world to world.	Extinction.	Which possible tweaks to our genetics might allow our race survive indefinitely?
The growth and aging of your body.	Developing cancer.	Which people will never get cancer?
Economics and finance.	Becoming wealthy.	Which people will never get rich?
Crime and punishment.	Going to jail.	Which kinds of careers allow a person to avoid incarceration forever?
Writing a book.	It's obviously finished.	Which projects are doomed from the outset never to be finished?
Working to improve one's mental outlook.	Serenity, tranquility, peace.	When is a person definitely on the wrong path?
Finding a mate.	Knowing that this is the one.	Who is doomed never to find true love?
Inventing something.	Eureka!	Which research programs are utterly hopeless?

Getting back to the weaker HPH, let me clarify its import by giving some fanciful examples. Table 1 lists a variety of real world computations. In each row, I suggest a computation, a notion of "target state", and a relevant question that has the form of a halting problem–where we try to detect initial states that produce endlessly running computations that never reach the specified target state. (I'm idealizing here, and temporarily setting aside the issue that none of the physical processes that I mention can in fact run for infinitely many years.)

Assuming that the HPH applies to these computations with these particular definitions of target state, we're faced with unsolvability, which means that none of the questions in the third column can be answered by a finding a simple way to detect which inputs will set off a process that never reaches the target states.

3 A Natural Incompleteness Theorem

Let's begin by defining what I mean by a formal system. A *formal system* F can be characterized as having four components: A set of symbols, a rule for recognizing which finite strings of symbols are grammatical sentences, a rule for deciding which sentences are to be regarded as the axioms of the system, and some inference rules for deducing sentences from other sentences.

A *proof* of a sentence S from the formal system F is a sequence of sentences, with the last sentence of the sequence being the targeted sentence S. Each preceding sentence must either be an axiom or be a sentence which is arrived at by combining still earlier sentences according to the inference rules. If a sentence is provable from F, we call it a theorem of F.

Combined with the notion of proof, a formal system becomes the source of a potentially endless number of theorems. Aided by a formal system, we mentally reach out into the unknown and produce facts about entirely new situations.

Now let's think of a formal system as a computation. There are several ways one might do this, but what's going to be most useful here is to work with a computation *FProvable* that captures the key aspect of a formal system: it finds theorems. Our *FProvable* will try to detect – so far as possible – which strings of symbols are theorems of F. That is, for any proposed provable sentence S, the computation *FProvable(S)* will carry out the following computation.

1. If S fails to be a grammatical sentence *FProvable(S)* returns False.
2. Otherwise *FProvable* starts mechanically generating proofs from the formal system F in order of proof size, and if S appears at the end of a proof, *FProvable(S)* returns True.
3. If S is a grammatical sentence but no proof of S is ever found, then *FProvable(S)* fails to halt.

As it turns out, if F is a powerful enough formal system to prove the basic facts of arithmetic, then FProvable will be universal. And then, by Turing's Theorem, FProvable has an unsolvable halting problem.[3]

Let's come back to Leibniz's dream. Suppose we could formulate some wonderfully rich and inclusive formal system F that includes mathematics, physics, biology, human psychology, and even the laws of human society. And then, just as Leibniz said, whenever we're asked if some statement S about the world were true, we'd set the computation FProvable(S) in motion, and the computation would eventually return True – provided that S is provable as well as true.

One cloud on the horizon is that, if S isn't provable, then FProvable(S) is going to run forever. And, due to the unsolvability of the halting problem, there's no way to filter out in advance those sentences S that are in fact unprovable sentences.

To delve deeper, we need two more definitions. As I mentioned before, we'll use $\sim$ to represent negation. So if S is a sentence, $\sim S$ means "not S". That is, S is false if and only if $\sim S$ is true. Using this notion of negation, we can formulate the notion of consistency.

Definition. F is consistent if and only if there is no sentence S such that F proves S and F proves $\sim$ S.

According to the usual rules of logic, if a theory proves even one contradiction, then it will go ahead and prove everything possible. So an inconsistent theory is useless for distinguishing between true and false statements about the world. We can reasonably suppose that our proposed Leibniz's-dream-type theory F is consistent.

What if *neither* S *nor* $\sim S$ are provable from F? As it turns out, the neither-nor case *does* happen. A lot! The reason has to do with, once again, the unsovability of the halting problem for FProvable.

Definition. If F is a formal system and S is a particular statement such that F proves neither S nor $\sim S$, we say S *is undecidable for* F.

A priori, we can see that there are four possible situations regarding the behavior of the "Is S provable?" computation, as shown in Table 2.

In their optimism, the early mathematical logicians such as David Hilbert hoped to find a formal system F such that the undecidable and inconsistent cases would never arise. As I mentioned earlier, Hilbert's program proposed finding a provably consistent formal system F that could decide all mathematical questions. But Hilbert's hopes were in vain. For, as I already mentioned, we

[3] Turing's work showed that arithmetic is strong enough to emulate the running of Turing machines. More specifically, he showed that for any F as strong as arithmetic, we can set things up so that FProvable emulates any given machine M. This means that FProvable is a universal computation, so Turing's Theorem applies, and FProvable has an unsolvable halting problem.

Table 2. Four Kinds of Provability and Unprovability

	FProvable($\sim S$) returns True	**FProvable**($\sim S$) doesn't halt
FProvable(S) returns True	F proves both S and $\sim S$, meaning F is inconsistent.	F proves S.
FProvable(S) **doesn't halt**	F proves $\sim$ S.	F proves neither S nor $\sim S$, meaning that S is undecidable for F.

have *Gödel's Incompleteness Theorem*, which tells us that any formal system designed along the lines of Leibniz's dream or Hilbert's program will leave infinitely many sentences undecidable. Let's look at the details.

Gödel's Incompleteness Theorem. If F is a consistent formal system as powerful as arithmetic, then there are infinitely many sentences which are undecidable for F.

What are these undecidable sentences like? As I mentioned in the introduction, one simple kind of undecidable sentence, call it G, might be characterized in terms of some algebraic property $g[n]$ that a number n might have. It might look like this, where $g[n]$ can be thought of as being a simple algebraic formula with the parameter n:

(G) For all n, $g[n]$ isn't true.

It's interesting, though a bit dizzying, to compare and contrast two related ways of talking about a sentence S. On the one hand, we can ask if S is true or false in the real world of numbers, and on the other hand we can ask if S or $\sim S$ happens to be provable from F . In the case where the sentence G has the form mentioned above, only three possibilities can occur. In order to illuminate the notion of undecidability, let's take a quick look at the three case.

1. *G is false, and $\sim G$ is provable.* If G is false, this means there is a specific n such that $g[n]$ holds in the world of numbers. F will be able to prove the instance $g[n]$ simply by checking the arithmetic. Therefore, F will be able to prove $\sim G$.
2. *G is true, and G is provable.* If the G sentence is true in the world of numbers, then $g[n]$ is false for every n. Now in some situations, there may be a clever proof of this general fact from F. I call such a proof "clever" because it somehow has to prove in a finite number of symbols that that $g[n]$ is impossible for *every* n. A general proof doesn't get bogged down at looking at every possible value of n. It has to use some kind of tricky reasoning to cover infinitely many cases at once.
3. *G is true, and G is not provable.* In these cases, there is no clever proof. The only way F could prove G would be to look at every possible number n and show that $g[n]$ isn't true – but this would take forever. In a case like this it's

> almost as if G only *happens* to be true. At least as far as F can see, there's no overarching reason *why* $g[n]$ is impossible for every n. It's just that, as chance would have it, in the real world there *aren't* any such n. And thus G is undecidable by F.

The computer scientist Gregory Chaitin suggests that in a case like the third, we think of G as a *random truth*. It's not true for any deep, theoretical reason. It's just something that turns out to be so. [4] Note that there's an endless supply of undecidable sentences S beyond the simple kinds of sentences G that I've been discussing. Some initial examples of the next level of complexity might be "For each m there is an n such that $g[m, n]$" or "There is an m such that for all n, $g[m, n]$." Most mathematicians would feel that, in the real world of mathematics, any of these sentences is definitely true or false, regardless of F's inability to prove either of the alternatives. And the true but unprovable statements are brute facts that hold for no particular reason.

So far, we've only been talking about number theory. How do we get to undecidable sentences about the *natural* world? If we accept the HPH, and if we assume that any natural process can be regarded as a computation, then we can find undecidability in any complex natural process!

The path leads through the following lemma, proved by Turing in 1936.

Unsolvability and Undecidability Lemma. If P is a computation with an unsolvable halting problem, and F is a correct formal theory, then there will be infinitely many sentences about P which are undecidable for F.

In this Lemma, by the way, I'm using the phrase "correct formal theory" to mean a formal theory that doesn't prove things which are false. I won't go into the somewhat technical details of the proof of this lemma, but the general idea is that there have to be lots of sentences about P that are undecidable for F, for otherwise F could solve P's unsolvable halting problem.

So now we come to the pay-off. Naturally occurring processes can be thought of as computations. If we accept the Halting Problem Hypothesis, then each naturally occurring process will have an unsolvable halting problem. And then, by applying Turing's Unsolvability and Undecidability Lemma, we get the following.

Natural Incompleteness Theorem. For most naturally occurring complex processes, and any correct formal system for science, there will be sentences about the process that are undecidable by the given formal system.

What makes the Natural Incompleteness Theorem attractive is that the undecidable sentences are not just about arithmetic. They're about the behavior of actual real-world processes.

No matter how thoroughly you try and figure the world out, there are infinitely many things you can't prove. Here are some examples of potentially undecidable sentences. Each of them may be, in principle, true or false, but only in a random

[4] You can find more details in [1], and in the papers on Chaitin's home page.

kind of way, in that they're not proved or disproved by any of our formal theories about the world.

Potentially undecidable statements about the natural world:

- Nobody will ever manage to bounce a golf ball a thousand times in a row off a putter head.
- There are an endless number of planets in our universe.
- There are an endless number of planets with people indistinguishable from you.
- No human will ever be born with six functioning arms.
- No cow's spots will ever spell out your first name in big puffy letters.
- The left wing will dominate American politics more often than the right wing does.
- Mankind will evolve into higher forms of life.
- The majority of times that you move to a different line in a supermarket, the new line goes slower than one of the lines you didn't pick.
- New races of intelligent beings will emerge over and over for the rest of time.
- The time of our cosmos extends forever.

Do note that, as with our examples about natural halting problems, we need some analysis of how to take into account the issue that so few of our natural systems can in fact be viewed as potentially eternal. But I'll leave the fine points of issue for other investigators to work out.

4 Undecidability Everywhere

It often happens in the history of science that some odd-ball new category is discovered. At first nobody's sure if any phenomena of this kind exist, but then there's some kind of logical argument why these odd-ball things have to occur. And then, as time goes on, more and more of the curious entities are discovered until finally they're perceived to be quite run of the mill. And I think this is what will happen with the notion of undecidable sentences about the natural world.

To dramatize this notion, I'll present a sustained analogy between the spread of undecidability and the rise of transcendental numbers in mathematics. Brian Silverman suggested this analogy to me in an email.

Transcendental Numbers. 300 BC. The Greeks worked primarily with real numbers that can be expressed either as the fraction of two whole numbers, or which can be obtained by the process of taking square roots. By the time of the Renaissance, mathematicians had learned to work with roots of all kinds, that is, with the full class of algebraic numbers – where an algebraic number can be expressed as the solution to some polynomial algebraic equation formulated in terms of whole numbers. The non-algebraic numbers were dubbed the

transcendental numbers. And, for a time, nobody was sure if any transcendental numbers existed.

Undecidable Sentences. 1920. In David Hilbert's time, it seemed possible that, at least in mathematics, every problem could be decided on the basis of a reasonable formal system. This was the inspiration for Hilbert's program.

Transcendental Numbers. 1884. The first constructions of transcendental real numbers were carried out by Joseph Liouville. Liouville's numbers were, however, quite artificial, such as the so-called Liouvillian number 0.11000100000000000000010000... which has a 1 in the decimal positions $n!$ and 0 in all the other places. Someone might readily say that a number like this is unlikely to occur in any real context. ($n!$ stands for "n factorial" which is the product $1 \times 2 \times \cdots \times n$ of all the integers from 1 to n.)

Undecidable Sentences. 1931. Kurt Gödel proved the existence of some particular undecidable algebraic sentences. These sentences were somewhat unnatural. Relative to a given formal system F, they had the form "This sentence is not provable from F," or the alternate form, "The contradiction $0 = 1$ is not provable from the formal system F."

Transcendental Numbers. 1874. Georg Cantor developed his set theory, and showed there are an infinite number of transcendental numbers. Someone could say that Cantor's transcendental numbers aren't numbers that would naturally occur, that they are artificial, and that they depend in an essential way upon higher-order concepts such as treating an infinite enumeration of reals as a completed object.

Undecidable Sentences. 1936. Building on Gödel's work, Alan Turing proved his theorem on the unsolvability of the halting problem. He immediately derived the corollary that there are infinitely many undecidable sentences of mathematics, and that these sentences come in quite arbitrary forms. Even so, the specific examples of such sentences that he could give were still odd and somewhat self-referential, like Gödel's undecidable sentences.

Transcendental Numbers. 1873. Charles Hermite proved that the relatively non-artificial number e is transcendental.

Undecidable Sentences. 1965. On an entirely different front, Paul J. Cohen proved that an important question about infinite sets called the continuum hypothesis is undecidable from the known axioms of mathematics. (Cohen's proof built on an earlier result proved by Kurt Gödel in 1946.) And in 1970, back in the realm of unsolvable halting problems, Julia Robinson, Martin Davis, Yuri Matiyasevich showed that among the sentences undecidable for any formal theory we'll find an infinite number of polynomial Diophantine equations which don't have any whole number solutions, but for which we can't prove this fact. This means there is a very large range of ordinary mathematical sentences which are undecidable.

Transcendental Numbers. 1882. Ferdinand Lindemann proved that the garden variety number π is transcendental.

Undecidable Sentences. 2002. Wolfram pointed out that we should be able to find numerous examples of undecidability in the natural world.

And now we have a Natural Incompleteness Theorem telling us that every possible complex natural process is going to have undecidable sentences associated with it! Undecidability is everywhere, and all of our theories about nature must remain incomplete.

References

[1] Chaitin, G.J.: The Unknowable. Springer, New York (1999)
[2] Leibniz, G.W., Gerhardt, C.I. (eds.): Die philosophischen Schriften von Gottfried Wilhelm Leibniz. Georg Olms Verlag, Hildesheim (1978)
[3] Rucker, R.: Infinity and the Mind. Birkhäuser, Boston (1982)
[4] Rucker, R.: The Lifebox, the Seashell, and the Soul. Thunder's Mouth Press, New York (2005)
[5] Wolfram, S.: A New Kind of Science. Wolfram Media, Champaign (2002)

Chapter 15
Pervasiveness of Universalities of Cellular Automata: Fascinating Life-Like Behaviours

Emmanuel Sapin

Laboratory MTG UMR CNRS IDEES, France, and
College of Engineering, Mathematics and Physical Sciences,
University of Exeter, UK
emmanuelsapin@hotmail.com, e.sapin@exeter.ac.uk

Abstract. We aim to construct an automatic system for the discovery of Turing-universal cellular automata. In this chapter, steps towards this automatic system are presented. These steps will lead to the discovery of thousands of computing patterns in cellular automata.

1 Introduction

One of the most fascinating phenomena in nature is the case of systems in which the whole exhibits more than the sum of its parts. Such systems are known as complex systems [25]. Some relevant examples of such systems are ant colonies and brains. One of the biggest challenges in science is to find out principles behind complex systems [1], indeed this is one of the greatest mysteries of the natural world [28].

Promising environments in which to study complex systems are the universes of cellular automata [12, 18] which are the simplest mathematical representations of complex systems [7], and important modelling paradigms in natural sciences. They are also uniform frameworks in which cells evolve through time on the basis of a local functions, called the transition rules [26].

Complex phenomena in cellular automata have different forms of which one of the most widely studied is the emergence of computation tasks. Some have studied specific computation like density and synchronization tasks [11, 13, 24], and pattern recognition [29]. While others have considered *Turing-universal automata* [3, 10, 9] i.e. automata encompassing the whole computational power of the class of Turing machines. Wolfram [27] and others have asked the question about the pervasiveness of universal cellular automata, this problem is tackled here.

In order to find universal automata, we aim to construct an automatic system which can discover Turing-universal cellular automata. Possible steps towards this automatic system are presented in this chapter. The Game of Life of Conway *et al.* [4] and some generalities about cellular automata are the subjects of Section 2 while, in Section 3, a search for traveling patterns is presented. A 2D 2-state automaton, called R, is described and it is proven to be Turing-universal in Section 4. In Section 5, a search for computing patterns is presented while a

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 199–210.
DOI: 10.1007/978-3-642-35482-3_15

search for simulations of logic gates is the subject of Section 6. To finish, the last section summarizes the results which have been put forward, and discusses directions for future research.

2 Previous Work

2.1 Game of Life

In 1970, Conway discovered a special automaton [25] he gave it the name *Game of Life* because of its resemblance to real-life processes [6]. In the Game of Life, cells on a regular grid can be born, die, or survive generation after generation. Cells follow local rules that consider only their eight closest neighbours: Cells stay alive if and only if they have two or three living neighbours and dead cells come alife if and only if they have three living neighbours. From these simple local rules and a random population of cells emerges a society of living organisms [6]. Patterns that emerge from random configurations of cells are shown in figure 1.

Pattern	Name	Average number of appearences
	Blinker	10
	Block	64
	Beehive	45
	Boat	5
	Tub	3

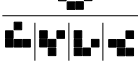
Pattern	Name	Average number of appearences
	Loaf	6
	Ship	7
	Long barge	0
	Pond	3
	Glider	16

Fig. 1. Patterns that emerge most often after 1000 generations in the Game of Life from four random configurations of cells in a 50×50 square with equal probability of each cell being either dead or alive

Stable and periodic patterns are common but the most remarkable discovery is an element that recovers its shape after shifting in space. Such mobile self-localized patterns of non-resting states called *spaceships* are very common in the Game of Life with various shapes and *periods* (i.e. number of generations before recovering their original shapes). Conway et al. called the five cell spaceship shown in figure 1 a *glider*. Gliders and glider generators called *guns* which, when evolving alone, periodically recover their original shape after emitting a number of spaceships led to the universality of the Game of Life [4].

2.2 Cellular Automata

The Game of Life is in the space $\mathcal{I}$ of isotropic two dimensional two-state automata using the eight closest neighbours of a cell to determine its state in the

next generation. The eight closest neighbours of a cell are defined as a *neighbourhood state* (hereafter referred to as a N-state). There are 2^8=256 different N-states [2] and the state of the central cell of a N-state in the next generation is determined by its state in the current generation and its N-state. At each N-state is associated a first number that is the state in which the central will be in the next generation if its state is 0 and a second number if its state is 1. In order to maintain the isotropy, these two numbers have to be the same for some N-states [20]. The 51 sets of such N-states are named the *isotropic N-state subsets* (hereafter referred to as INS subsets).

The convention that is used here is the one of the freeware *Life32* that can be downloaded at [5]. Fig. 2 shows one element from each INS subset that is referred to by the number of cells from 0 to 8 and a letter. Conventionally, the INS subset that allows the central cell to survive is noticed in the first part after the letter *s* of the cellular automaton notation and the INS subset that allows the central cell to be born is noticed in the second part after the letter *b*. For example, the notation *s*2*iv*3*a*8/*b*4*i*8 means cells survive for the INS subsets 2*i*, 2*v*, 3*a* and 8.

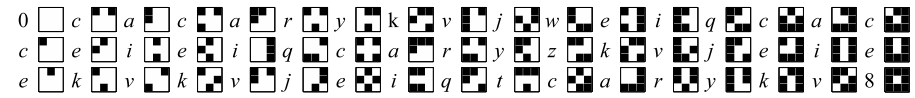


Fig. 2. An element from each INS subset that is referred to by the number of cells from 0 to 8 and a letter

3 Search for Spaceships

As the glider of the Game of Life is one of the key elements to demonstrate its universality, a search for spaceships is one of the possible first steps to search for universal cellular automata and is presented here.

3.1 Method

Various stochastic algorithms have been tried to search for spaceships and Monte Carlo method and evolutionary algorithms are described here:

- The Monte Carlo method consists solely of generating random solutions and testing them.
- Evolutionary algorithms incorporate aspects of natural selection or survival of the fittest. It maintains a population of structures (usually initially generated at random) that evolves according to rules of selection, recombination, mutation, and survival referred to as genetic operators. A shared environment is used to determine the fitness or performance of each individual in the population. The fittest individuals are more likely to be selected for

reproduction through recombination and mutation, in order to obtain potentially superior ones.

Throughout our experiment, no spaceship was found with a Monte Carlo method in one million randomly generated cellular automata. Then, an evolutionary approach was tried in which the fitness of each individual is determined using the number of spaceships and periodic patterns that appear from a random configuration of cells.

3.2 Results

Our algorithm was able to find tens of thousands of spaceships that can move orthogonally or diagonally [17]. When a spaceship is found, this spaceship may or may not be a new discovery. Considering the number of spaceships the algorithm can find, an automatic system to determine whether or not a spaceship was unknown was required. This system has been based on the transition rules of the automata that accept the considered spaceship [19]. Spaceships of different periods were found and figure 2 shows the distribution of the periods of some orthogonal and diagonal spaceships.

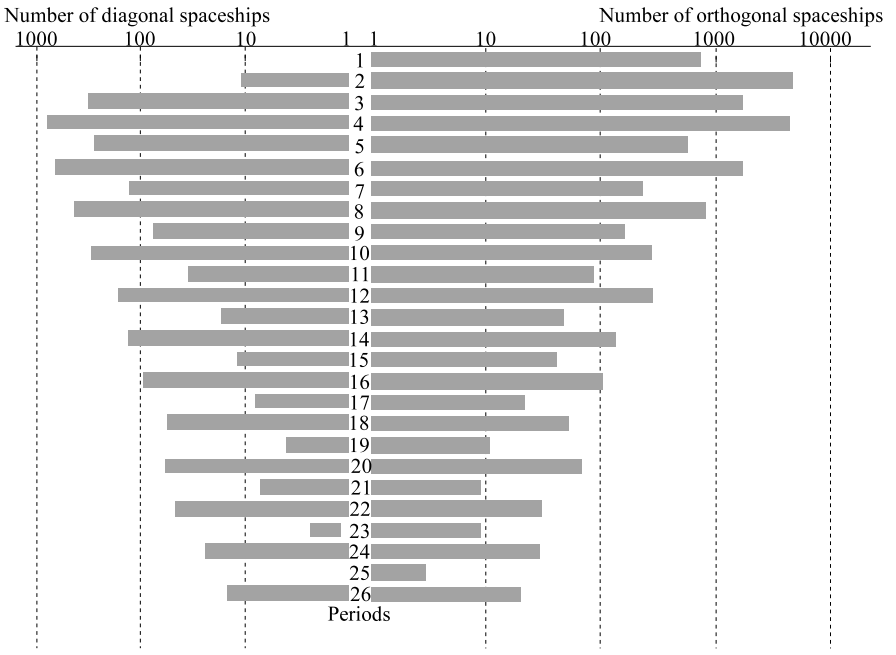


Fig. 3. Distribution of the period of the discovered orthogonal and diagonal spaceships for the first 20000 spaceships found in an experiment

The even periods are more common than odd. That can be explained by the fact some of the spaceships that were discovered recover a symmetry of their original shape after half of their period has elapsed as shown in figure 4 for a spaceship.

Fig. 4. A spaceship of period 4 in generations 0, 1, 2, 3 and 4 evolving with the transition rule $s/b2ca$

The spaceships with low periods are the ones for which the average numbers of appearances from random configurations of cells are the highest. In the experiment, the spaceship with the highest number of appearances from a random configuration of cells when it was discovered is a surprising result and will be the subject of the next section.

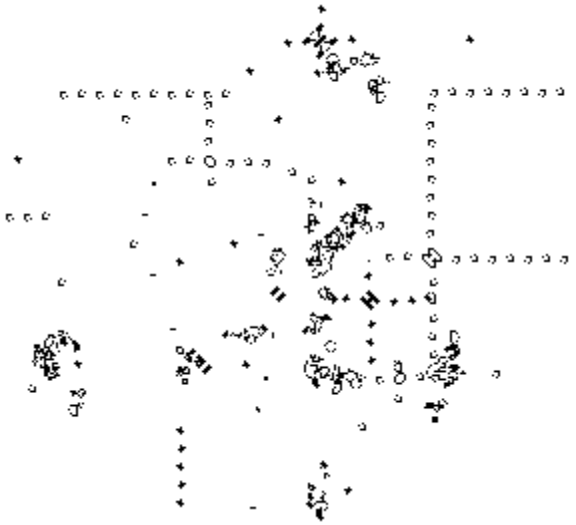


Fig. 5. Evolution of cells by the automaton R

4 Automaton R

This section is devoted to a cellular automaton we called R that was discovered from the search for spaceships. This automaton is described by $s2eki3caiv$ $rqjy4cvqjz/b3caivqy4cekaivrqytwz5ckairj6cekaiv8$ in the convention of Life32.

A visual analysis of the evolution of cells by R, shown in figure 5, leads to the conclusion that most of the spaceships that appeared were emitted by guns that spontaneously appear from the evolution of cells.

As for the Game of Life, patterns that emerge from random configurations of cells are shown in figure 6.

Pattern	Type	Average number of appearances
	Blinker	7
	Block	3
	Periodic	2
	Gun and emitted spaceships	4
	Stable	1

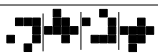
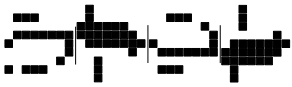
Pattern	Type	Average number of appearences
	Periodic	1
	Spaceship	185
	Spaceship	6
	Spaceship	4

Fig. 6. Patterns that emerge the most often after 1000 generations in the automaton R from 4 random configurations of cells in a 50×50 squares with equal probability to be dead and alive for each cell

Guns can appear in the Game of Life from collisions of gliders but such a frequent appearance of guns from random configurations of cells is a new and very surprising result. Also the automaton R is chosen on which attempts are made to demonstrate its universality.

The demonstration of universality of R is inspired by the demonstration of universality of the Game of Life in which streams of spaceships are used to carry information where spaceships represent 1s and missing spaceships represent 0s.

In this demonstration of universality of the Game of Life and the automaton R, a collision, Conway called *vanishing reaction* and shown in figure 7 with spaceships in R, is used to simulate `and` gates. Collisions of streams of spaceships and a stream emitted by a gun is used in order to turn at right angle streams and to realize `and` gates. The result of a collision of a stream of spaceships A and a stream from a gun is a stream orthogonal to the stream A and carrying the information not(A). This stream crashes into the stream B and each spaceship

Fig. 7. Vanishing collision of two spaceships generation after generation in the automaton R

of stream not(A) when it exists destroys the corresponding spaceships of B when it exists. Then the stream B is turned into a stream $A\&B$.

In the Game of Life, a very tricky mechanism using a particular collision of gliders Conway called *Kickback reaction* is used in order to perform duplication of streams and a `not` gate. This collision does not exist in R but a very simple way to duplicate a stream is described then another method to simulate a `not` gate is presented. The `not` gate and the duplication of streams are used in order to demonstrate the universality of R.

4.1 Duplication

The duplication of streams is based on a specific collision of spaceships we called *duplicative collision* and shown in figure 8.

Fig. 8. Duplicative collision of two spaceships generation after generation

In order to duplicate a stream A, an orthogonal stream B emitted by a gun crashes into it. All spaceships of stream A survive from this collision whereas each spaceship of stream B is destroyed by the corresponding spaceship of stream A when it exists. Then the stream B carries the information not(A) and is orthogonal to stream A. Another stream emitted by a gun can collide with stream B in order to make it parallel to stream A and to carry the information A.

4.2 `Not` Gate

The simulation of a `not` gate in cellular automata is a genuine challenge. A gun can complement a stream of spaceships but turns it at a right angle at the same time. To perform a `not` gate complementing without turning a stream must be done or, that is equivalent, by turning without complementing. The latter has been chosen in the automaton R and done thanks to a particular frontal collision of two streams that creates a spaceship at a right angle for any spaceship that was in the stream [21]. The scheme of the `not` gate, using this frontal collision and the duplication of streams, is shown in figure 9.

4.3 Universality of R

In order to demonstrate the universality of the automaton R, the idea is to simulate the Game of Life with R. The Game of Life being universal, R will also be demonstrated as universal. Using simulations of `and` gates, `not` gates

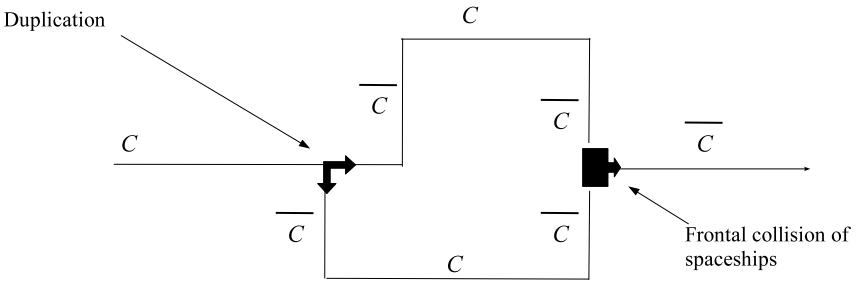


Fig. 9. A scheme of the `not` gate in the automaton R with the input stream C

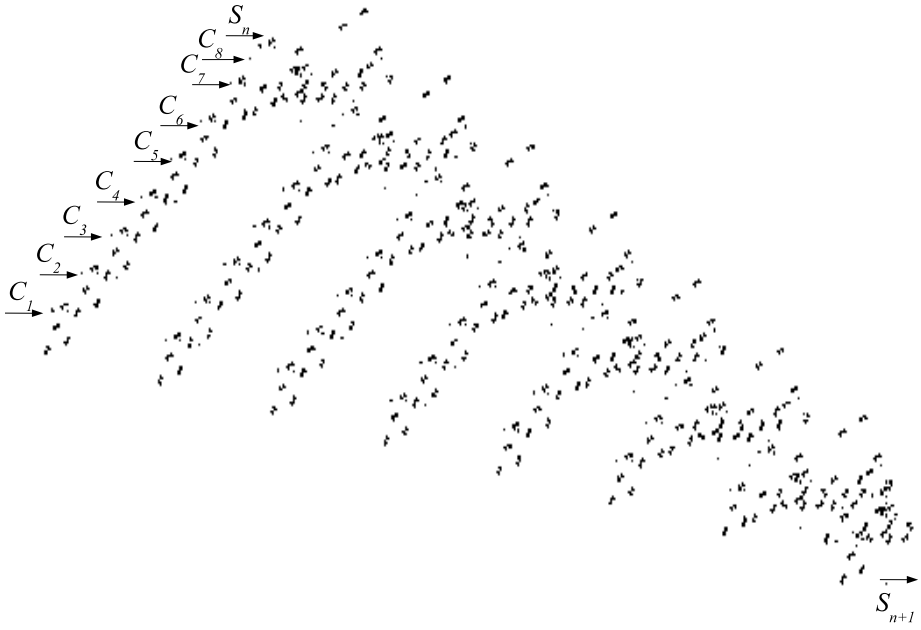


Fig. 10. Pattern that represents a cell S of the Game of Life in the automaton R. The nine input streams represent the state of the eight direct neighbours of S and the state of this cell in a generation n. The output stream is the state of S in generation $n + 1$.

and duplications of streams, a cell of the Game of Life can be simulated [16], as shown in figure 10.

The simulation of the Game of Life is then shown by carrying out a tilling of a surface with the identical and interconnected simulations of cells as a proof that R is universal in the Turing sense [22]. The automaton R is thus the first 2D two-state dynamical automaton, other than the Game of Life, that has been demonstrated to be universal in the Turing sense.

5 Search for Guns

As guns are one of the key elements to demonstrate the universalities of the automaton R and the Game of Life, a search for guns is one of the possible steps to search for universal cellular automata and is presented here.

Inspired by the search for spaceships, an evolutionary approach is used to find guns. The idea is to focus on a spaceship and to find a gun emitting this spaceship. As the gun of the automaton R was found by optimising the number of appearances of spaceships from a random configuration of cells, this number has been optimised by the evolutionary process in order to find guns. The results of this research are described in the next subsections.

5.1 Guns for Various Spaceships

Guns are searched for from a sample of one hundred spaceships [17]. For each spaceship, ten runs of the algorithm have been realized for each of the one hundred spaceships. These runs allow us to find guns for 55 spaceships, also fewer guns were found for spaceships of high periods and fewer guns were also found for diagonal spaceships.

5.2 Guns for a Specific Spaceship

Ten guns over ten runs of the algorithm were found for a specific spaceship shown in figure 11. These ten guns were all different. Then, a hundred runs of the algorithm have been realized and for each run a new gun was found. An extensive search found the quantity of more than 20000 guns for this spaceship [19]. All the discovered guns have emerged spontaneously from the evolution of a random configuration of cells.

The emergence of this number of guns is a very surprising result as Resnick et al. claimed: 'You would be shocked to see something as large and organized as a gun emerge spontaneously.' [14]. The existence of so many different guns shows that they are more easy to find than Wolfram claimed "There are however indications that the required initial configuration is quite large, and is very difficult to find." [26]. Thus the discovery of so many different guns represents a significant contribution to the theory of cellular automata and complex systems that considers computational theory.

6 Search for Logic Gates

The universality of the Game of Life and the automaton R is demonstrated using simulations of logic gates, they are then searched for in the automata with a spaceship and a gun that were discovered by the search that is described in the previous section.

6.1 Search for and Gates

`And` gates are simulated in the Game of Life and R using vanishing reactions shown in figure 7 then simulations of this collision by new automata are searched for. The idea is to search for vanishing reactions that can be simulated by the discovered automata that already simulate guns and spaceships. An evolutionary algorithm has been elaborated to search for vanishing reactions.

Simulations of `and` gates have been discovered for 15 percent of the automata with a spaceship and a gun [23].

6.2 Search for not Gates

The kickback reaction that allows the simulation of `not` gates in the Game of Life and the frontal collision of spaceships that allows to simulate `not` gate in the automaton R are searched for in the automata able to simulate `and` gates but they have not been found. Another way to simulate `not` gates was then searched for. A pattern, called *90-degree reflector* or *reflector* that allows a stream to be redirected without changing its value is searched for.

In 2 percent of the automata that simulate `and` gates a reflector is found [15]. A reflector that turns a spaceship at a right angle that was emitted by a gun is shown in figure 11 generation after generation.

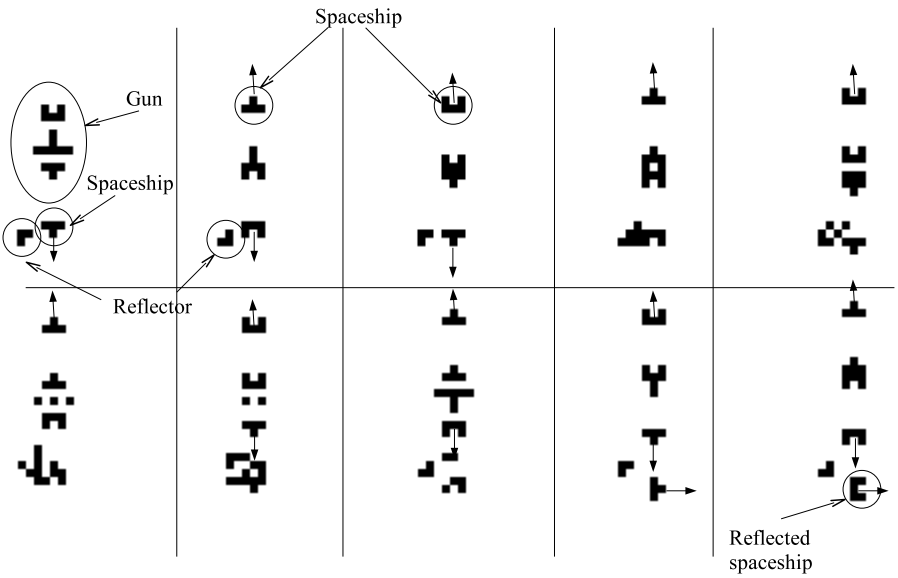


Fig. 11. A reflector that turn at right angle a spaceship that was emitted by a gun generation after generation with the transition rule *s*2*kaiv*3*ce kvj*4*caqr*5*cekrqy*6*civ*7*c*8/*b*2*ci*3*cekayj*4*kavyq*5*ekivy*6*cekai*8 in the convention of Life32

7 Synthesis and Perspectives

This paper deals with the emergence of computation in complex systems with local interactions. A contribution to the well established problem of universality in cellular automata is presented. Steps toward an automatic system for the discovery of Turing-universal cellular automata are presented. These steps are a search for spaceships, a demonstration of universality of a discovered automaton, a search for guns and simulations of logic gates.

Tens of thousands of cellular automata that allow spaceships to appear from random configurations of cells were discovered. One of them called R also allows guns to appear and is demonstrated to be universal. Guns are searched for in some of these automata. For more than half of the one in which guns are searched for, guns were found. Simulations of logic gates were searched for and found. All these discoveries suggest that universality is common in cellular automata.

Next steps towards an automatic system for the discovery of Turing-universal cellular automata are to find another ways to simulate logic gates. A further step will be to find an automatic way to search for duplication of streams and the possibility to combine logic gates. Future work could also be to search for sets of conditions that are sufficient for a cellular automaton to be universal.

Additionally, another domain that seems worth exploring is to study how spaceships appear from a random configuration of cells and which part of the transition rule makes them move throughout generations. Also this research opens the possibility of an evaluation of all discovered universal automata and calculate for each one some rule-based parameters, e.g., Langton's lamda [8]. All universal automata may have similar values for these parameters that could lead to an answer to the question 'Where are the edges of computational universality?' and may therefore lead to a better understanding of the emergence of computation in complex systems with local interactions.

References

[1] Bubak, M., Sloot, P.M.A., Hoekstra, A.G., Portegies Zwart, S.F.: Towards Distributed Petascale Computing. Springer, Heidelberg (2007)
[2] Adamatzky, A., De Lacy Costello, B., Asai, T. (eds.): Reaction-diffusion computers. Elsevier (2005)
[3] Banks, E.R.: Information and transmission in cellular automata. PhD thesis, MIT (1971)
[4] Berlekamp, E., Conway, J.H., Guy, R.: Winning ways for your mathematical plays. Academic Press, New York (1982)
[5] Bontes, J. (2009), `http://psoup.math.wisc.edu/life32.html`
[6] Gardner, M.: Wheels, Life, and Other Mathematical Amusements. Freeman & Co. (1983) ISBN 0-7167-1589-9
[7] Ilachinski, A.: Cellular Automata. World Scientific (1992)
[8] Langton, C.L.: Computation at the edge of chaos. Physica D 42 (1990)
[9] Lindgren, K., Nordahl, M.: Universal computation in simple one dimensional cellular automata. Complex Systems 4, 299–318 (1990)
[10] Margolus, N.: Physics-like models of computation. Physica D 10, 81–95 (1984)

[11] Mitchell, M., Crutchfield, J.P., Hraber, P.T.: Evolving cellular automata to perform computations: Mechanisms and impediments. Physica D 75, 361–391 (1994)
[12] Von Neumann, J.: Theory of Self-Reproducing Automata. University of Illinois Press, Urbana (1966)
[13] Packard, N.H.: Adaptation toward the edge of chaos. In: Kelso, J.A.S., Mandell, A.J., Shlesinger, M.F. (eds.) Dynamic Patterns in Complex Systems, pp. 293–301 (1988)
[14] Resnick, M., Silverman, B.: Exploring emergence (1996), `http://llk.media.mit.edu/projects/emergence/`
[15] Sapin, E.: Evolutionary search for cellular automata simulating nand gate. In: EA 2011. LNCS (2011)
[16] Sapin, E.: From Gliders to Universality of Cellular Automata: Another 2D 2-state Universal Automaton. Nova Science Publishers, NY (2011) ISBN: 978-1-61761-592-4
[17] Sapin, E.: Gliders and glider guns discovery in cellular automata. In: Adamatzky, A. (ed.) Game of Life Cellular Automata. Springer (page in press)
[18] Sapin, E., Adamatzky, A., Bull, L.: Genetic approaches to search for computing patterns in cellular automata. IEEE Computational Intelligence Magazine, 4
[19] Sapin, E., Adamatzky, A., Collet, P., Bull, L.: Stochastic automated search methods in cellular automata: The discovery of tens of thousands glider guns. Natural Computing (in press)
[20] Sapin, E., Bailleux, O., Chabrier, J.J.: Research of a Cellular Automaton Simulating Logic Gates by Evolutionary Algorithms. In: Ryan, C., Soule, T., Keijzer, M., Tsang, E.P.K., Poli, R., Costa, E. (eds.) EuroGP 2003. LNCS, vol. 2610, pp. 414–423. Springer, Heidelberg (2003)
[21] Sapin, E., Bailleux, O., Chabrier, J.-J., Collet, P.: A New Universal Cellular Automaton Discovered by Evolutionary Algorithms. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 175–187. Springer, Heidelberg (2004)
[22] Sapin, E., Bailleux, O., Chabrier, J.J., Collet, P.: Demonstration of the universality of a new cellular automaton. IJUC 2(3) (2006)
[23] Sapin, E., Bull, L.: Evolutionary search for cellular automata logic gates with collision based computing. Complex Systems (in press)
[24] Sipper, M.: Evolution of parallel cellular machines. In: Stauffer, D. (ed.) Annual Reviews of Computational Physics, vol. V, pp. 243–285. World Scientific (1997)
[25] Waldrop, M.M.: Complexity: The Emerging Science at the Edge of Chaos. Simon and Schuster, New York (1992)
[26] Wolfram, S.: Universality and complexity in cellular automata. Physica D 10, 1–35 (1984)
[27] Wolfram, S.: Twenty problems in the theory of cellular automata. Physica Scripta, 170–183 (1985)
[28] Wolfram, S.: A New Kind of Science. Wolfram Media, Inc., Illinois (2002)
[29] Wolz, D., de Oliveira, P.B.: Very effective evolutionary techniques for searching cellular automata rule spaces. Journal of Cellular Automata 3(4), 289–312 (2008)

Chapter 16
A Spectral Portrait of the Elementary Cellular Automata Rule Space

Eurico L.P. Ruivo[2] and Pedro P.B. de Oliveira[1,2]

Universidade Presbiteriana Mackenzie
[1] Faculdade de Computação e Informática &
[2] Pós-Graduação em Engenharia Elétrica, São Paulo, SP – Brazil
euricoruivo@gmail.com, pedrob@mackenzie.br

Abstract. Fourier spectra of cellular automata rules give a quantitative characterisation of the frequency of bit patterns present in the limit configurations generated out of their time evolution. By means of the similarity among the spectrum of each rule, the elementary cellular automata rule space is partitioned, both for periodic and non-periodic boundary conditions, thus giving rise to rule classes of spectral equivalence. The partitions generated are formally explained in terms of the dynamical behaviour of the rules, and maps of the elementary space are given in terms of graphs that represent the spectral similarity among the rules, thus displaying a spectral portrait of how the classes relate to each other.

Keywords: Fourier spectrum, discrete Fourier transform, elementary cellular automata, discrete dynamical systems, dynamical behaviour of cellular automata.

1 Introduction

The analysis of the Fourier spectra of cellular automata rules arises as an approach to investigate their dynamical behaviour, the existence of patterns and attractors generated by their temporal evolution ([13], [2] and [7]), and provides a summary of the types of blocks that make up the limit configurations that can be generated. Moreover, the Fourier transforms, or the spectra obtained by means of it, give statistical information about the behaviour of cellular automata ([12]).

The Fourier spectrum of a cellular automaton (CA) rule is obtained by computing the *discrete Fourier transform* (DFT) for a set of limit configurations generated through the time evolution of the rule, thus giving a *frequency* domain representation for the average limit configuration of the CA. This means that, in the particular case of binary cellular automata, such a frequency becomes a measure of how often every possible bit change pattern is present in all possible binary strings that make up the limit configurations entailed by a rule.

Not many analyses of cellular automata based on the Fourier spectrum itself—or on its *power spectrum*, which is obtained by squaring the Fourier spectrum coordinate-wise—can be found in the literature. For instance, in [2] the

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 211–235.
DOI: 10.1007/978-3-642-35482-3_16

elementary cellular automata (ECA) is studied along this perspective. Also, [5] analyses the particular ECA rule 110, showing that it displays $1/f$ noise power spectrum, and a genetic algorithm is used in [6] to search for rules that present that type of power spectrum, based on the conjecture that there might be a relation between computational universality and $1/f$ noise power spectrum. It is worth noticing that in [5], [6] and [7], the spectra are computed along the time evolution of a rule iterated from a single initial configuration, rather than computing the spectra over sets of final configurations obtained at the end of the time evolution of a rule, as done here.

It is possible to classify the ECA rule space according to the spectra obtained by the DFT of each particular rule. In [7], for instance, the space is partitioned into four broad groups of rules, according to the general characteristics of their power spectra.

But it is tempting to consider a finer-grained, Fourier transform based classification of a CA rule space, by taking into account the absolute similarity among the rules' spectra, rather than their general characteristics. This is the direction we take here.

Accordingly, a partition of the ECA rule space in disjoint sets with similar Fourier spectra is given and the meaning of such spectral similarity is studied based on the dynamical behaviour of the rules; this allows us to refer to *spectral classes* in the ECA. Moreover, graphs representing a proximity structure between the different spectral classes obtained are derived. The similarity between rule spectra implies similarity between the blocks that constitute the final configurations obtained out of the time evolution of the rules (possibly, except for reflection, conjugation or conjugated reflection). Not only the similarity of blocks of limit configurations is represented in resembling spectra, but also the similarity among the densities at which every possible bit change pattern is present in the limit configurations.

The key point of this paper is to provide a portrait or map of the ECA rule space, from the standpoint of their spectra, both for cyclic and acyclic lattices. This is achieved through a graph structure representing a notion of similarity among classes of spectral equivalence, a notion that is also derived herein. Bits and pieces of the text are drawn from our recent, much shorter account on the subject [9], where the presentation is limited to cyclic configurations, and none of the actual maps are displayed.

This paper is organised as follows. In the next section, the notion of the Fourier spectrum of a cellular automaton rule is presented, as well as an explanation to a rule spectrum. The spectral similarity among ECA rules under periodic boundary condition (PBC), and the reasons behind it, are explained in Section 3; also, the spectral classes obtained out of partitioning the ECA rule space based upon this similarity are listed in Section 4 and these classes are organised in a graph in Section 5, giving a mapping of the ECA rule space in terms of spectral similarity. Then, another partition of the ECA rule space is given and another graph is built, now with configurations under non-periodic boundary conditions in Section 5.1. Concluding remarks are made in Section 6.

2 Fourier Spectrum of a Cellular Automaton

Before presenting a way to compute the spectrum of a cellular automaton rule, let us first consider the *discrete Fourier transform* (DFT), upon which the computation of the spectrum is based.

Given a complex vector $u = (u_1, \cdots, u_n)$, the DFT of u is denoted by $DFT(u)$ and corresponds to the complex vector $v = (v_1, \cdots, v_n)$ given by

$$v_k = \frac{1}{n} \sum_{j=1}^{n} u_j e^{2\pi i(j-1)(k-1)/n} \tag{1}$$

Here, $|DFT(u)|$ will denote the absolute value of $DFT(u)$ taken coordinate-wise. That is, if $DFT(u) = (v_1, \cdots, v_n)$, then $|DFT(u)| = (|v_1|, \cdots, |v_n|)$.

The Fourier spectrum of a cellular automaton rule is obtained by computing the DFT for a set of configurations (*final configurations* - FCs) obtained after the time evolution of the rule applied to a set of initial configurations (ICs).

Formally, given an ECA local rule f, a set C of N ICs of length L and a number of iterations $t \in \mathbb{R}$, the *spectrum S_f of rule f* is given by

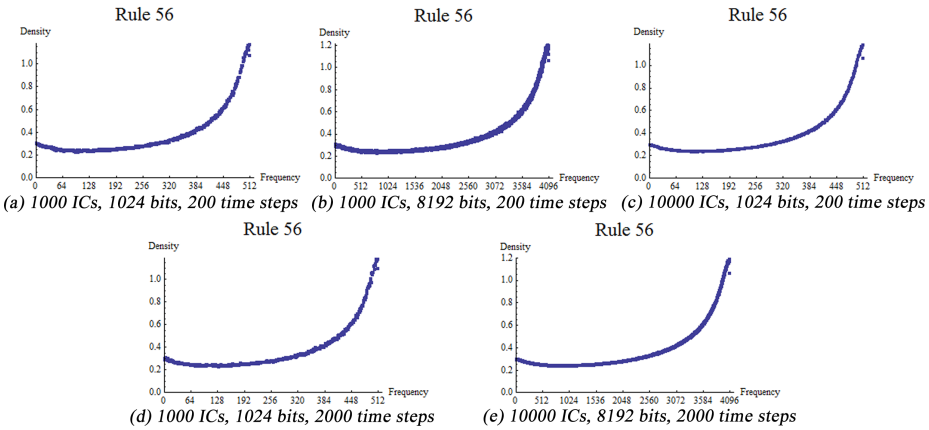
$$S_f = \frac{1}{N} \sum_{c \in C} |DFT(F^t(c))| \tag{2}$$

where F is the global rule induced by f and $F^t(c)$ denotes the configuration obtained by iterating t times F over c.

Since the computation of the spectrum of an ECA is obtained by iterating a rule a given number of times on a random sample of initial configurations of a fixed length, it may present statistical fluctuations depending on the set of ICs, their length and the number of iterations of the rule.

The power spectra computed in [2] were obtained by iterating each ECA rule from 15 to more than 100 times, over each initial configuration, in order to ensure that each computed spectrum would be beyond any transient stage. Considering this and that the sets of initial configurations had to be sufficiently large, the experiments in Sections 3 to 5 were made with 1,000 random ICs of length 1,024 and 200 iterations.

Also, in order to ensure that each spectrum represents the limit behaviour of a rule, it is worth verifying the sensibility of the spectrum with their parameters. As an example, Figure 1 shows the effects of changing the set of parameters used to compute the spectrum of rule 56. Taking larger ICs (Figure 1.*b*), the result is a thicker spectrum, since the DFT of each IC has more coordinates; with more ICs (Figure 1.*c*), the resulting spectrum is smoother, for it is obtained from a larger sampling of initial conditions; for more iterations (Figure 1.*d*), the spectrum becomes essentially the same, for most rules, as discussed below. (Figure 1.*e*) shows the spectrum of rule 56 computed for larger values of the parameters in comparison to those used on this paper. In any case, the shape of the spectrum does not change, therefore the set of parameters previously chosen (Figure 1.*a*: 1,000 ICs of 1,024 bits and 200 iterations of each rule) describes indeed a representative spectral behaviour of the rules.



(a) 1000 ICs, 1024 bits, 200 time steps (b) 1000 ICs, 8192 bits, 200 time steps (c) 10000 ICs, 1024 bits, 200 time steps

(d) 1000 ICs, 1024 bits, 2000 time steps (e) 10000 ICs, 8192 bits, 2000 time steps

Fig. 1. Rule 56 spectra for different sets of parameters. The essential aspect of the spectrum remains the same, even with larger values for the parameters.

Although increasing the number of iterations far above 200 (the average number of iterations taken to compute the spectra) does not change the spectra of most ECA rules, this is not the case for rules in Wolfram class IV (complex rules), such as rules 54 and 110, which take a longer time to stabilise. Figure 2 shows the spectra computed for 200, 1,000 and 2,000 iterations for rules 54 and 110. Qualitative changes are present in these spectra, with peaks appearing and growing for longer times. That occurs because the transient times for complex rules are much longer than those for rules in other Wolfram classes. For instance, according to [13], the transient time for rule 110 from the IC made up of a single 1 surrounded by 0s is about 2,800 iterations. Therefore, rules with spectra that change along time are more complex than those without such a feature.

Let us now consider the kind of information present in the spectra. Clearly, this has two facets: it summarizes the patterns of bit sequences that appear out of the time evolution of the rule (after a sufficiently long time) and the *frequency* of changes of values among adjacent bits in these patterns.

The amount of changes between adjacent bits in a binary sequence can be considered as the *frequency of bit changes* in it. Then, the horizontal axis of a spectrum represents the sets of patterns with the same frequency of bit changes formed out of the time evolution of the rule, with the sequences that have higher frequency of bit changes on the right-hand side of the axis. As for the vertical axis, it represents the density that each set of patterns that have the same frequency of bit changes appear in the final configurations.

An illustrative example is given by rule 184, whose action on cyclic configuration is known to be as follows: for configurations with the same amount of 0s and 1s, the time evolution leads to configurations with alternating 0s and 1s; for configurations with more 0s than 1s, the final configurations display alternating 0s and 1s, as in the previous case, and the exceeding 0s are clustered in

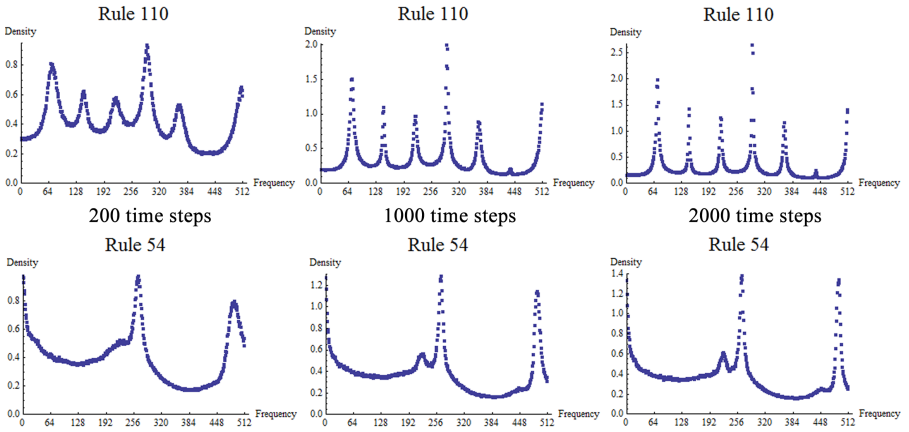


Fig. 2. Spectra of rules 54 and 110 computed with different number of time steps, with 1,000 ICs of 1,024 bits

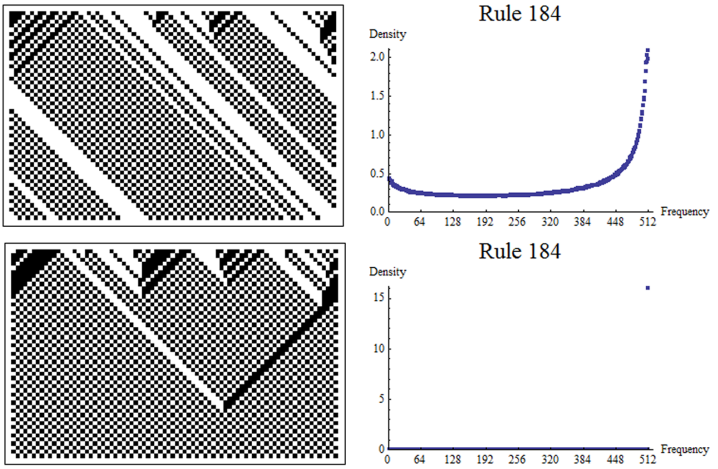


Fig. 3. Time evolution of rule 184 for an initial configuration with density of 1s equal to 0.4 (a) and 0.5 (b) and their corresponding spectra computed on a set of such initial configurations

groups; and for configurations with more 1s than 0s, the final configurations are analogous to those described in the previous case.

Figure 3.*a* shows the spectrum of rule 184 computed on a set of initial configurations with density of 1s equal to 0.4. According to the interpretation of the spectrum given above, the higher values are present in the right-hand side of the graph, depicting that the final configurations have many regions with high frequency of bit changes - that is, regions with blocks of the type $(\cdots, 0, 1, 0, 1, \cdots)$ or $(\cdots, 1, 0, 1, 0, \cdots)$. As for the left-hand side of the graph, it depicts the regions

of the groups of exceeding 0s present in the final configurations. On the other hand, by taking a set of ICs with density of 1s equal to 0.5 (same amount of 0s and 1s), all the FCs will present only the pattern $(\cdots, 1, 0, 1, 0, \cdots)$, which corresponds to the highest frequency of bit changes. The result is presented in Figure 3.*b*, with a null spectrum except for an isolated peak at the highest frequency.

Given all the latter, we can turn to the organization of the ECA rule space in terms of its spectra. First of all, it is expected that dynamically equivalent rules present similar spectra, since the FCs obtained by such rules are qualitatively the same. Since the DFT of pairs of conjugated and/or reflected vectors are the same, whenever the spectra of any two dynamically equivalent ECAs are computed with the parameter values listed above, they have very similar spectra, thus providing a fair practical approximation to their sameness.

Even though any dynamically equivalent rules have the same spectrum, the converse is not true. As shown in Figure 4 the spectra of rules 18, 146, 182 and 183 are similar, but rule 18 is only dynamically equivalent to rule 183, while rule 146 is dynamically equivalent only to rule 182.

This means that there are factors behind spectral similarity other than the dynamical equivalence. These are discussed in the next section.

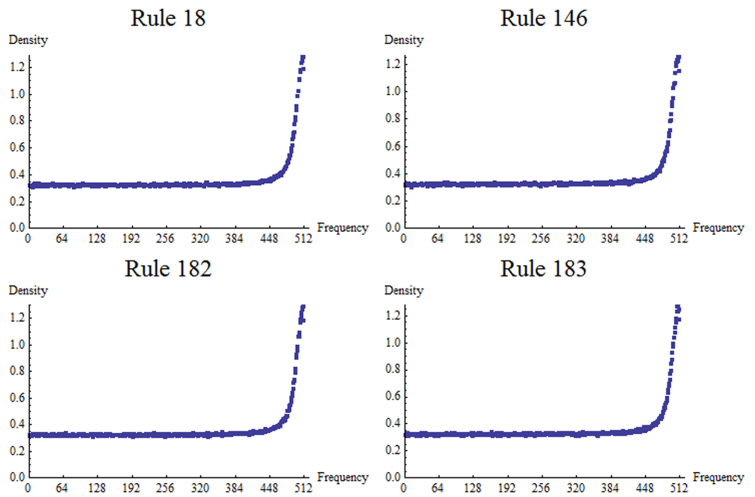


Fig. 4. Despite the fact that rules {18, 183} and {146, 182} belong pairwise to different dynamical equivalence classes, their spectra are similar

3 Spectral Similarity in the ECA Rule Space

In order to analyse the reasons behind the spectral similarity among ECA rules, the rule space was partitioned according to the spectral similarity of the rules. For this, the distance between two spectra was defined as the euclidean distance

between the vectors of the spectra. The distances between each spectrum were computed and then normalised by the maximum computed distance. Then, for each local rule f, with $\epsilon > 0$, the set $N_f^{0,\epsilon}$ of the *spectral neighbours of* f was defined as

$$N_f^{0,\epsilon} = \{\text{g is an ECA local rule} : d(f_S, g_S) < \epsilon\} \tag{3}$$

where $d(.,.)$ is the euclidean distance function.

However, with this definition, it may occur that the spectral neighbourhood of a rule $g \in N_f^{0,\epsilon}$ contains rules that are not present in the spectral neighbourhood of f, since the spectra are generated through sampling of initial configurations, what leads to statistical fluctuations. In order to sort out this issue, the sets $N_f^{k,\epsilon}$, $k \in \mathbb{N}$, $k > 0$, are defined as

$$N_f^{k,\epsilon} = \{g \in N_h^{0,\epsilon} : h \in N_f^{(k-1),\epsilon}\} \tag{4}$$

Since the ECA rule space is finite, for each local rule f there is $k_f \in \mathbb{N}$ such that $N_f^{(k+1),\epsilon} = N_f^{k,\epsilon}$, for all $k > k_f$. Therefore, it is possible to define the *spectral class of* f, S_f^ϵ, as

$$S_f^\epsilon = \bigcup_{i=0}^{k_f} N_f^{i,\epsilon} \tag{5}$$

The spectrum of rule g is said to be *similar* to that of rule f if $g \in S_f^\epsilon$

By induction on k, if $g \in N_f^{0,\epsilon}$, then for all $h \in N_f^{k,\epsilon}$, $h \in N_g^{(k+1),\epsilon}$ holds. Therefore, from Equations 3, 4 and 5 the equality $S_f^\epsilon = S_g^\epsilon$ holds for all $g \in S_f^\epsilon$, that is, if the spectrum of rule f is similar to that of rule g, then the spectrum of rule g is similar to that of rule f, as expected.

For the results presented herein, $\epsilon = 0.005$ was taken, which resulted in a partition of the ECA rule space into 59 disjoint spectral classes, which are presented in Section 4. The ϵ parameter was empirically determined in order to preserve the visual coherency of each spectral class, in the sense that only rules with visually similar spectra would end up in the same subset, and that no pair of distinct subsets would contain visually similar spectra. However, as presented below, this similarity is not merely visual, and can be justified in terms of the dynamics of the rules, making the partition formally coherent and relevant.

Since the spectrum of a rule depends upon the binary patterns present in the final configurations generated out of the time evolution of the rule, there are three main reasons underlying spectral similarity: (1) *symmetry of FCs and/or dynamical equivalence*, (2) *surjectivity* and (3) *rules with equivalent behaviour*. These reasons are more thoroughly discussed below.

3.1 Spectral Similarity due to Symmetries of Configurations and Dynamical Equivalence

Since the DFT essentially reveals periodicity in data, if two given binary vectors u and u' are such that u' can be obtained by applying the symmetry operators

of reflection, conjugation and conjugated reflection to vector u, then the discrete Fourier transform of both vectors will be the same. Moreover, under configurations taken under periodic boundary conditions, the *shift* operator can also be considered a symmetry operation. Therefore, if vector u' can be obtained by applying reflection, conjugation (and possibly a shift for the cyclic lattice case) or any composition of these operators to vector u, the DFT of both vectors will be identical.

Hence, if two ECA rules f_1 and f_2 (with inducted global rules F_1 and F_2, respectively), iterated a sufficiently large number of time steps, over any initial configuration c_0, result in two configurations c_1 and c_2, respectively, that are the same, except for symmetry, then the spectra of rules f_1 and f_2 will be identical. Formally,

$$\forall c_0, \forall t \in \mathbb{N}, F_1^t(c_0) = sym(F_2^t(c_0)) \Rightarrow |DFT(F_1^t(c_0))| = |DFT(F_2^t(c_0))| \quad (6)$$

where $sym(\cdot)$ denotes the application of reflection, conjugation, shift (the last is only a symmetry in the case of periodic boundary condition) or any composition of them.

In a similar fashion, as the dynamical equivalence classes are constructed by applying reflection, conjugation or conjugated reflection to the rule tables, and since the experiments were made over random sets of initial conditions, the spectra of dynamically equivalent rules will be similar, as described above.

Out of the 59 subsets mentioned above, 54 of them present rules with similar spectra due to symmetries in the final configurations and to dynamical equivalence, and are presented in Section 4. The remaining 5 subsets will be discussed in the next subsections.

3.2 White-Noise Spectra

A remarkable class in the partition of the ECA rule space computed is the one represented by rule 15, which exactly comprehends the *surjective* ECA rules (listed in Section 4). The spectra of those rules have a white-noise type behaviour (Figure 5), which implies that the final configurations obtained by the iterative applications of the rules to the set of random initial conditions have no noticeable periodicity, that is, the final configurations are essentially as random as the initial conditions.

Since in a surjective CA every pattern of a given size has the same number of pre-images ([1]), each configuration obtained after a fixed number of time steps has a set of initial configurations that lead to it along the time evolution of the same size. That is, any final configuration has the same probability of being obtained at the end of the time evolution, regardless of the choice of the configuration.

Therefore, in the present conditions of spectra generation, the sets of final configurations will have a white-noise type display; consequently, all surjective ECAs are in the same spectral class which, in fact, they end up defining.

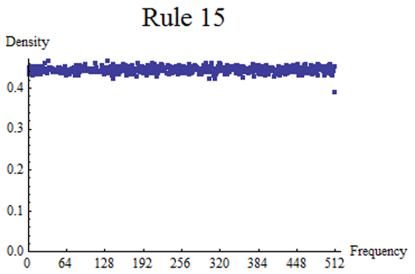


Fig. 5. The spectrum of rule 15 and the other surjective ECA rules have a white-noise type nature

3.3 Rules with Equivalent Behaviour

The rules of each of the 4 last spectral classes of the ECA space in Section 4 do not show similarity due to the reasons discussed above; however, they result in final configurations where certain binary sequences tend not to be present. Part of the rules in these classes are dynamically equivalent, which partially explains the composition of these classes. Despite the fact that not all rules in each class are dynamically equivalent, the rules show equivalent behaviour after a given number of iterations. Each particular case is explained below.

Class 13 - rules 13, 69, 78, 79, 92, 93, 141 and 197
Dynamical classes: 13 (13, 69, 79 and 93) and 78 (78, 92, 141 and 197)
Rules 78 and 79 only differ at the $(0,0,0)$ neighbourhood: for rule 78, $(0,0,0) \to 0$ and for rule 79 $(0,0,0) \to 1$.

For a set of 1,000,000 random ICs of 1024 bits, rules 78 and 79 were iterated 200 times over each IC and the sequence $(0,0,0)$ was absent in every resulting final configuration. That is, the only block upon which rules 78 and 79 differ, disappears during the time evolution of the rules.

Therefore, after a given number of time steps, rules 78 and 79 behave exactly the same, thus leading to equivalent spectra. As for the other rules in the class, the equivalence is given by dynamical equivalence to rules 78 or 79.

Class 18 - rules 18, 146, 182 and 183
Dynamical classes: 18 (18 and 183) and 146 (146 and 182)
Rules 18 and 146 only differ at the $(1,1,1)$ neighbourhood: $(1,1,1) \to 0$ for rule 18 and $(1,1,1) \to 1$ for rule 146. The set of sucessive pre-images of $(1,1,1)$ by rule 146 is given by $(1,1,1) \leftarrow (1,1,1,1,1) \leftarrow (1,1,1,1,1,1,1) \leftarrow \cdots$. Also, $(1,1,1)$ has no pre-image for rule 18.

If a configuration is not fully made of 1s, after a number of iterations of rule 146, the $(1,1,1)$ sequence will no longer be present in the configuration and rule 146 will then behave exactly as rule 18. On the other hand, if a configuration is made up of 1s, both rules (18 and 146) will lead it to a configuration with only 0s or 1s, that is, configurations that lead to null spectrum.

Hence, rules 18 and 146 display the same spectrum and the other rules in the class also present equivalent spectra due to dynamical equivalence to rules 18 or 146.

Class 28 - rules 28, 70, 156, 157, 198 and 199
Dynamical classes: 28 (28, 70, 157 and 199) and 156 (156 and 198)
Rules 156 and 157 only differ at the $(0,0,0)$ neighbourhood. For rule 56, $(0,0,0) \to 0$, and for rule 157, $(0,0,0) \to 1$. Also, there is no 5-bit sequence that has $(0,0,0)$ as its image under rule 157. Hence, any configuration containing $(0,0,0)$ is a Garden-of-Eden (GoE) for 157. Therefore, after one iteration of rule 157 over any configuration, the sequence $(0,0,0)$ is absent and, from there, rule 157 will behave just like 156. As for the converse, rule 156 may not behave as rule 157 for some particular configurations, since $(0,0,0)$ has preimages for rule 156. Nevertheless, in general, $(0,0,0)$ is eliminated by rule 156 over its time evolution. For a set of 1,000,000 random ICs of 1024 bits, rule 156 was iterated 200 times over each IC and the sequence $(0,0,0)$ was absent in every FC obtained.

The remaining equivalences are due to dynamical equivalence to rules 156 or 157.

Class 122 - rules 122, 126, 129 and 161
Dynamical classes: 122 (122 and 161) and 126 (126 and 129)
For this class, the explanation is analogous to the one for class 28. Rules 122 and 126 only differ at the neighbourhood $(0,1,0)$: $((0,1,0) \to 0$ for rule 122 and $(0,1,0) \to 1$ for rule 126). Also, $(0,1,0)$ is not an image of any 5-bit sequence for rule 126; hence, any configuration containing $(0,1,0)$ is a GoE for rule 126. Therefore, after one iteration of rule 126 over any configuration, the sequence $(0,1,0)$ is absent and rule 126 ends up behaving like rule 122. As for rule 122, it has preimages of $(0,1,0)$, hence, for some particular configurations, rules 122 and 126 may present distinct behaviour. However, in general, rule 122 excludes the sequence $(0,1,0)$ during its time evolution (for 200 iterations of rule 122 over a set of 1,000,000 random ICs of 1024 bits, $(0,1,0)$ was absent in every final configuration obtained), what justifies the spectral equivalence.

The other equivalences are due to dynamical equivalence to rules 122 or 126.

Out of the four classes described above, only one was formalised in an analytical way which has shown the spectral equivalence of the rules in this class for any case. The other three classes were partially (or completely) based upon empirical results which are useful to understand what makes these groups of rules belong to the same spectral classes. A full characterisation of the sets of initial configurations upon which the rules behave exactly the same is certainly a point to be considered in future investigations.

4 Classes of Spectral Equivalence

With the considerations made in Sections 3.1, 3.2 and 3.3, it is possible to formally define the notion of a *class of spectral equivalence* (or *spectral class*) in the

ECA rule space. Two ECA rules are said to be on the same spectral class if any of the following occur:

(a) The rules are in the same dynamical equivalence class, or the final configurations obtained by them are the same, except for conjugation, reflection and/or shift (only for the case of periodic boundary condition) or a composition of them;
(b) Both rules are surjective; and/or
(c) The rules present equivalent behaviour after a number of time steps.

Based upon the previous discussions, we obtained the spectral classes shown in Table 1, where *DE / SFC* denotes dynamical equivalence and/or symmetry of final configurations, *S* denotes surjective rules; and *EB* denotes equivalent behaviour.

Table 1. Partition of the ECA rule space in spectral classes

Rep.	Rules on spectral class	Type	Class
0	0, 8, 32, 40, 64, 96, 128, 136, 160	DE / SFC	I
-	168, 192, 224, 234, 235, 238, 239, 248	-	-
-	249, 250, 251, 252, 253, 254, 255	-	-
1	1, 127	DE / SFC	II
2	2, 16, 172, 191, 202, 216, 228, 247	DE / SFC	II
3	3, 17, 63, 119	DE / SFC SFC	II
4	4, 223	DE / SFC SFC	II
5	5, 95	DE / SFC	II
6	6, 20, 159, 215	DE / SFC	II
7	7, 21, 31, 87	DE / SFC	II
9	9, 65, 111, 125	DE / SFC	II
10	10, 80, 175, 245	DE / SFC	II
11	11, 47, 81, 117	DE / SFC	II
12	12, 34, 48, 68, 140, 187, 196	DE / SFC	II
-	206, 207, 220, 221, 243	-	-
13	13, 69, 78, 79, 92, 93, 141, 197	EB	II
14	14, 84, 143, 213	DE / SFC	II
15	15, 30, 45, 51, 60, 75, 85, 86, 89, 90	S	II/III
-	101, 102, 105, 106, 120, 135, 149	-	-
-	150, 153, 154, 165, 166, 169, 170	-	-
-	180, 195, 204, 210, 225, 240	-	-
18	18, 146, 182, 183	EB	III
19	19, 55	DE / SFC	II
22	22, 151	DE / SFC	III
23	23, 232	DE / SFC	II
24	24, 66, 189, 231	DE / SFC	II
25	25, 61, 67, 103	DE / SFC	II
26	26, 82, 167, 181	DE / SFC	II

Table 1. (*continued*)

Rep.	Rules on spectral class	Type	Class
27	27, 39, 53, 83	DE / SFC	II
28	28, 70, 156, 157, 198, 199	EB	II
29	29, 71	DE / SFC	II
33	33, 123	DE / SFC	II
35	35, 49, 59, 115	DE / SFC	II
36	36, 219	DE / SFC	II
37	37, 91	DE / SFC	II
38	38, 52, 155, 211	DE / SFC	II
41	41, 97, 107, 121	DE / SFC	II
42	42, 112, 171, 241	DE / SFC	II
43	43, 113, 142, 212	DE / SFC	II
44	44, 100, 203, 217	DE / SFC	II
46	46, 116, 139, 209	DE / SFC	II
50	50, 77, 178, 179	DE / SFC	II
54	54, 147	DE / SFC	IV
56	56, 98, 185, 227	DE / SFC	II
57	57, 99	DE / SFC	II
58	58, 114, 163, 177	DE / SFC	II
62	62, 118, 131, 145	DE / SFC	II
72	72, 237	DE / SFC	II
73	73, 109	DE / SFC	III
74	74, 88, 173, 229	DE / SFC	II
76	76, 205	DE / SFC	II
94	94, 133	DE / SFC	II
104	104, 233	DE / SFC	II
108	108, 201	DE / SFC	II
110	110, 124, 137, 193	DE / SFC	IV
122	122, 126, 129, 161	EB	III
130	130, 144, 190, 246	DE / SFC	II
132	132, 222	DE / SFC	II
134	134, 148, 158, 214	DE / SFC	II
138	138, 174, 208, 244	DE / SFC	II
152	152, 188, 194, 230	DE / SFC	II
162	162, 176, 186, 242	DE / SFC	II
164	164, 218	DE / SFC	II
184	184, 226	DE / SFC	II
200	200, 236	DE / SFC	II

Rules in the same class yield similar sets of final configurations, except for reflection, conjugation or conjugated reflection. Moreover, the spectra also depict how much the different frequencies of bit changes are represented in these configurations.

5 Similarity among Spectral Classes

Since the ECA rule space is partitioned into rule sets with equivalent spectra, the question now is how does each spectral class relate to each other in terms of spectral similarity. For instance, Figure 6 shows the spectra of rules 12 and 76. Even though these rules belong to distinct spectral classes there is a clear resemblance among their spectra. In this section we try to answer the question of how the ECA rule space can be organised in terms of the spectra of the rules; that is, a portrait of the ECA rule space is provided in these terms, depicted as a graph. The vertices of the graph are the spectra of the representatives of the spectral classes and the edges are weighted and computed as described below, representing a proximity between the spectral classes.

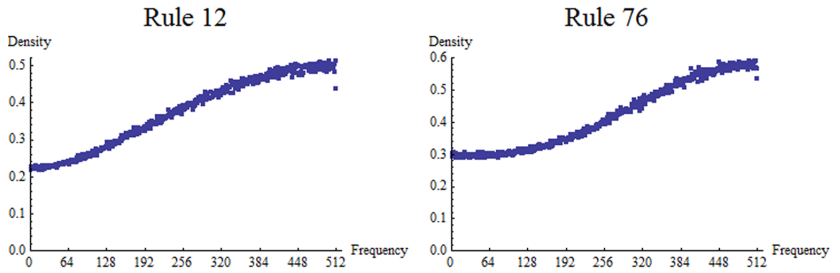


Fig. 6. Even though rules 12 and 76 belong to distinct spectral classes, there is a resemblance among their spectra

The procedure to obtain such a graph is as follows. First of all, for each spectral class, its rule with the smallest number is taken to represent it and assigned a node in the graph. Then, the spectrum of each class is computed and normalised by its highest component value. In the sequence, the euclidean distance between each pair of spectra is computed and normalised by the largest distance computed; each class is then connected (as an edge in the graph) to its closest neighbour, according to the smallest distance between them - such a procedure resulted in 19 groups of connected classes (i.e., 19 second-order type spectral classes); the process is then iterated, and each group is connected to its closest group, also given by the smallest distance between them, which resulted in 4 sets of groups. Finally, after one more iteration, the groups were connected, resulting in the graph.

The reason for not having all rules in the same group of connected classes is the generation of *cycles of connections* during the process. For instance, consider the spectral classes a, b and c. It may occur that a connects to b, b connects to c and c connects to a, giving rise to a group of connected classes apart from the other classes. The same applies to the connected components: the process yields cycles of connections between the groups of connected classes, generating more than one connected component.

Since the distance measure chosen here is the euclidean distance, the weights on the graph's edges stand for the absolute distance between the vectors of the representative spectra, rather than a visual distance between them. The result is shown below: Figure 7 shows the connection between the 19 groups (G1 to G9) and Figures 8 to 10 show the groups of rules that were obtained.

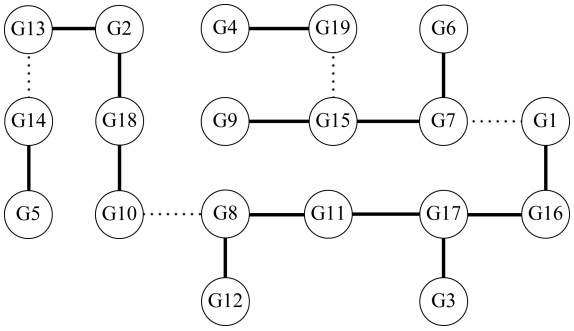


Fig. 7. Graph representing the connections of the 19 groups of spectral classes. Solid lines represent connections obtained in the second iteration of the process, while dotted lines represent connections obtained in the third and last iteration.

Since the process forces every class to be connected to another, some of the connections in the previous graph do not represent spectral similarity among different spectral classes, but rather a mapping of the pairs of closest classes (in respect to euclidean distance).

5.1 Spectral Similarity under Non-periodic Boundary Conditions

The previous results were obtained out of the time evolution of ECA rules under periodic boundary condition (PBC). In this section the partition of the ECA rule space in spectral classes is made under non-periodic boundary conditions (NPBC) and the respective graph is shown.

To compute the spectra, for each rule, a set of 1,000 random ICs of length 2,048 bits was superimposed over backgrounds of 0s or 1s (half of the number of ICs for each background) and the rule was iterated a number of times ranging from 4,000 to 8,000 time steps. Since the size of the (active) resulting final configuration (FC) obtained out of the time evolution of an IC may vary depending on the particular IC, a part of each FC was taken rather than the FC itself. Each part corresponds to a 1,024-bit substring of a FC, taken at a random point.

Under NPBC, the active part of the configuration may display significant changes over the time evolution. For instance, rule 102 yields triangular patterns appearing periodically in its time evolution under NPBC (Figure 11), thus entailing the bit patterns of the configuration to vary drastically through time.

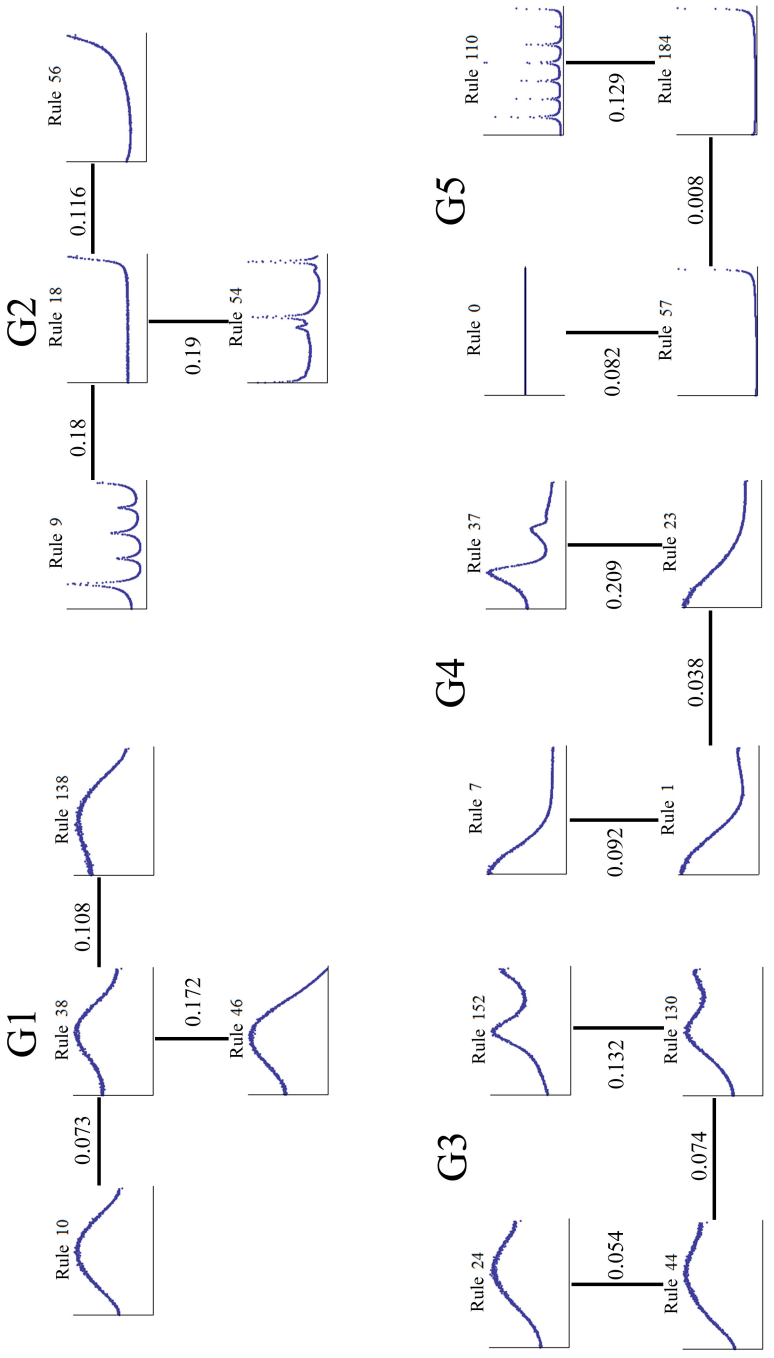


Fig. 8. Groups (1 through 5) of spectral classes, under PBC

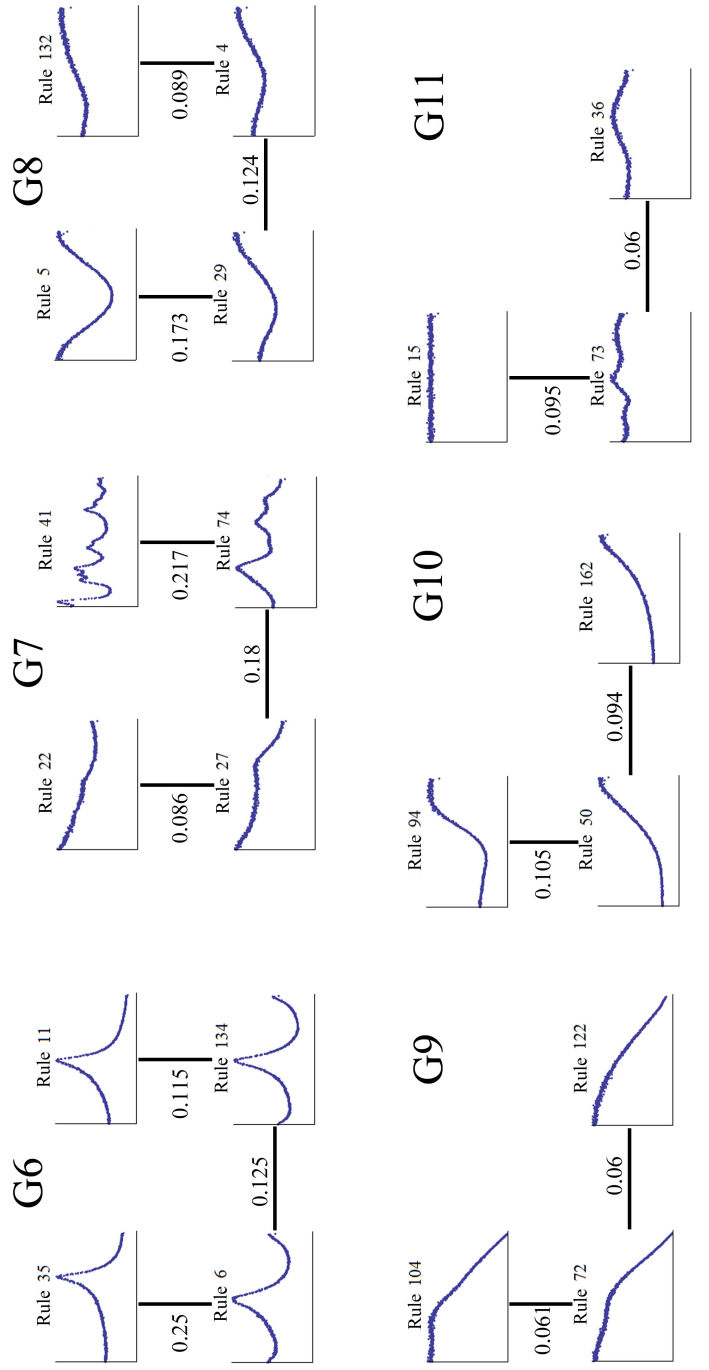


Fig. 9. Groups (6 through 11) of spectral classes, under PBC

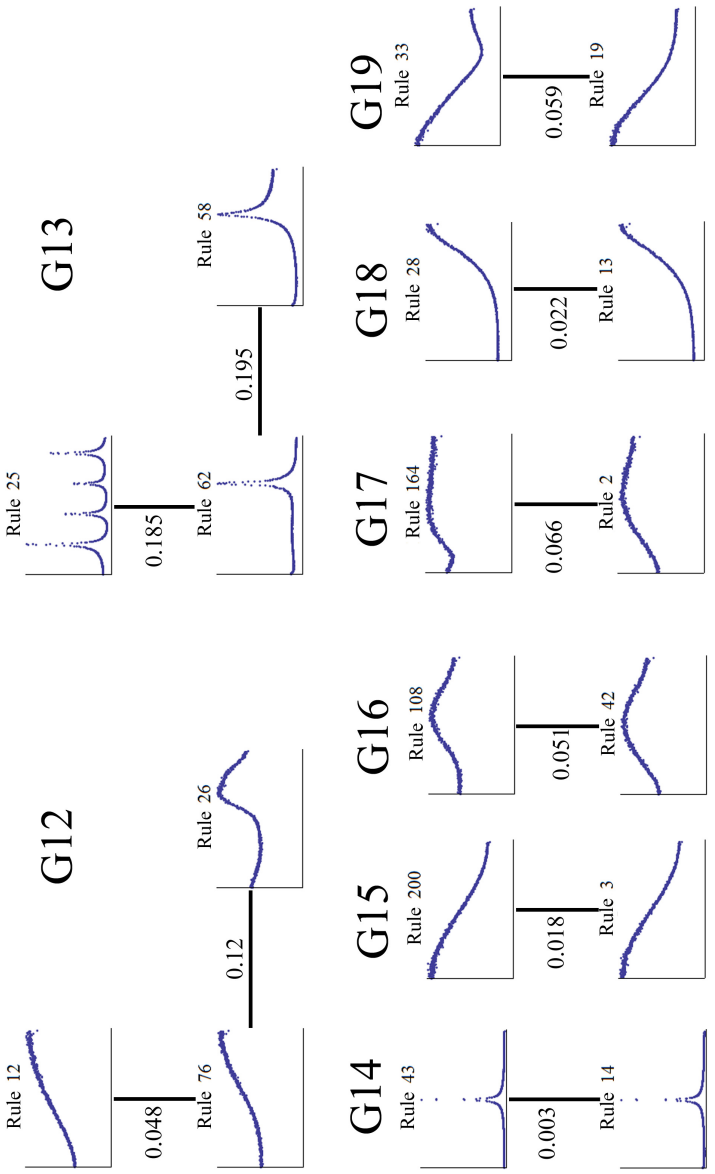


Fig. 10. Groups (12 through 19) of spectral classes, under PBC

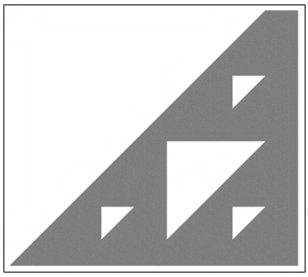


Fig. 11. Time evolution of rule 102 on a random IC of length 2,048 over 16,000 time steps. Triangular patterns are formed periodically.

Table 2. Partition of the ECA space under NPBC

Rep.	Rules in spectral class	Class
0	0, 8, 32, 40, 64, 96, 128, 136, 160	I
-	168, 192, 224, 234, 235, 238, 239, 248	-
-	249, 250, 251, 252, 253, 254, 255	-
1	1, 127	II
2	2, 16, 172, 191, 202, 216, 228, 247	II
3	3, 17, 63, 119	II
4	4, 223	II
5	5, 95	II
6	6, 20, 159, 215	II
7	7, 21, 31, 87	II
9	9, 65, 111, 125	II
10	10, 80, 175, 245	II
11	11, 47, 81, 117	II
12	12, 34, 48, 68, 140, 187, 196	II
-	206, 207, 220, 221, 243	-
13	13, 28, 69, 70, 78, 79, 92	II
-	93, 141, 156, 157, 197, 198, 199	-
14	14, 43, 84, 113, 142, 143, 212, 213	II
15	15, 30, 45, 51, 75, 85,	II
-	86, 89, 101, 106, 120, 135,	-
-	149, 169, 170, 204, 225, 240	-
18	18, 146, 182, 183	III
19	19, 55	II
22	22, 151	III
23	23, 232	II
24	24, 66, 189, 231	II
25	25, 61, 67, 103	II
26	26, 82, 167, 181	II

Table 2. (*continued*)

Rep.	Rules on spectral class	Class
27	27, 39, 53, 83	II
29	29, 71	II
33	33, 123	II
35	35, 49, 59, 115	II
36	36, 219	II
37	37, 91	II
38	38, 52, 155, 211	II
41	41, 97, 107, 121	II
42	42, 112, 171, 241	II
44	44, 100, 203, 217	II
46	46, 116, 139, 209	II
50	50, 77, 178, 179	II
54	54, 147	IV
56	56, 98, 185, 227	II
57	57, 99	II
58	58, 114, 163, 177	II
60	60, 90, 102, 105, 150, 153, 165, 195	III
62	62, 118, 131, 145	II
72	72, 237	II
73	73, 109	III
74	74, 88, 173, 229	II
76	76, 205	II
94	94, 133	II
104	104, 233	II
108	108, 201	II
110	110, 124, 137, 193	IV
122	122, 126, 129, 161	III
130	130, 144, 190, 246	II
132	132, 222	II
134	134, 148, 158, 214	II
138	138, 174, 208, 244	II
152	152, 188, 194, 230	II
154	154, 166, 180, 210	II
162	162, 176, 186, 242	II
164	164, 218	II
184	184, 226	II
200	200, 236	II

The rules were then partitioned as in the PBC case. Taking $\epsilon = 0.02$, which was established empirically, the resulting partition of the ECA rule space also led to 59 spectral classes (although not the same classes as in the PBC partition). In comparison to the $\epsilon = 0.005$ taken under the PBC case, the larger value of

the parameter under NPBC highlights the fact that similar spectra under NPBC displays greater fluctuation than the ones computed under PBC. The spectral classes obtained under NPBC are presented in Table 2.

It is worth noticing that in the partition under NPBC, the surjective rules (spectral class 15 under PBC) were divided into spectral classes 15, 60, and 154. However, their spectra are visually similar, as shown in Figure 12. In fact, most of the spectra is white-noise type, except for peaks in the lower and/or higher frequencies.

Following the same procedure introduced for PBC, a similarity graph has also been generated for NPBC. The resulting graph has 16 groups, in contrast to the one obtained under PBC, which has 19 groups. This can be explained by the generation of cycles of connections, as in the PBC case. Since the configurations

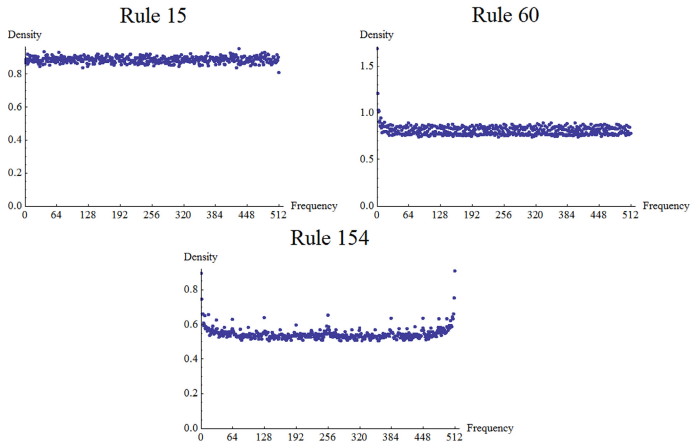


Fig. 12. Even though the surjective rules are in distinct spectral classes under NPBC, their spectra remain similar

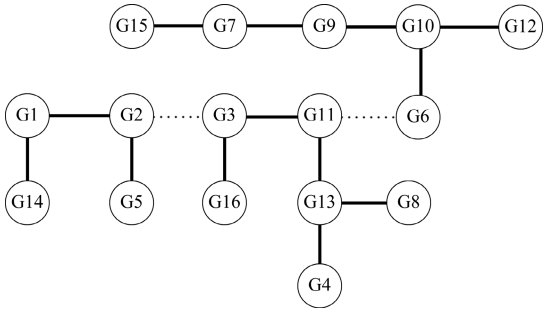


Fig. 13. Graph representing the connections among groups of spectral classes under NPBC. Solid lines depict connections of groups obtained in the second iteration of the process and dotted lines depict connections obtained in the third iteration.

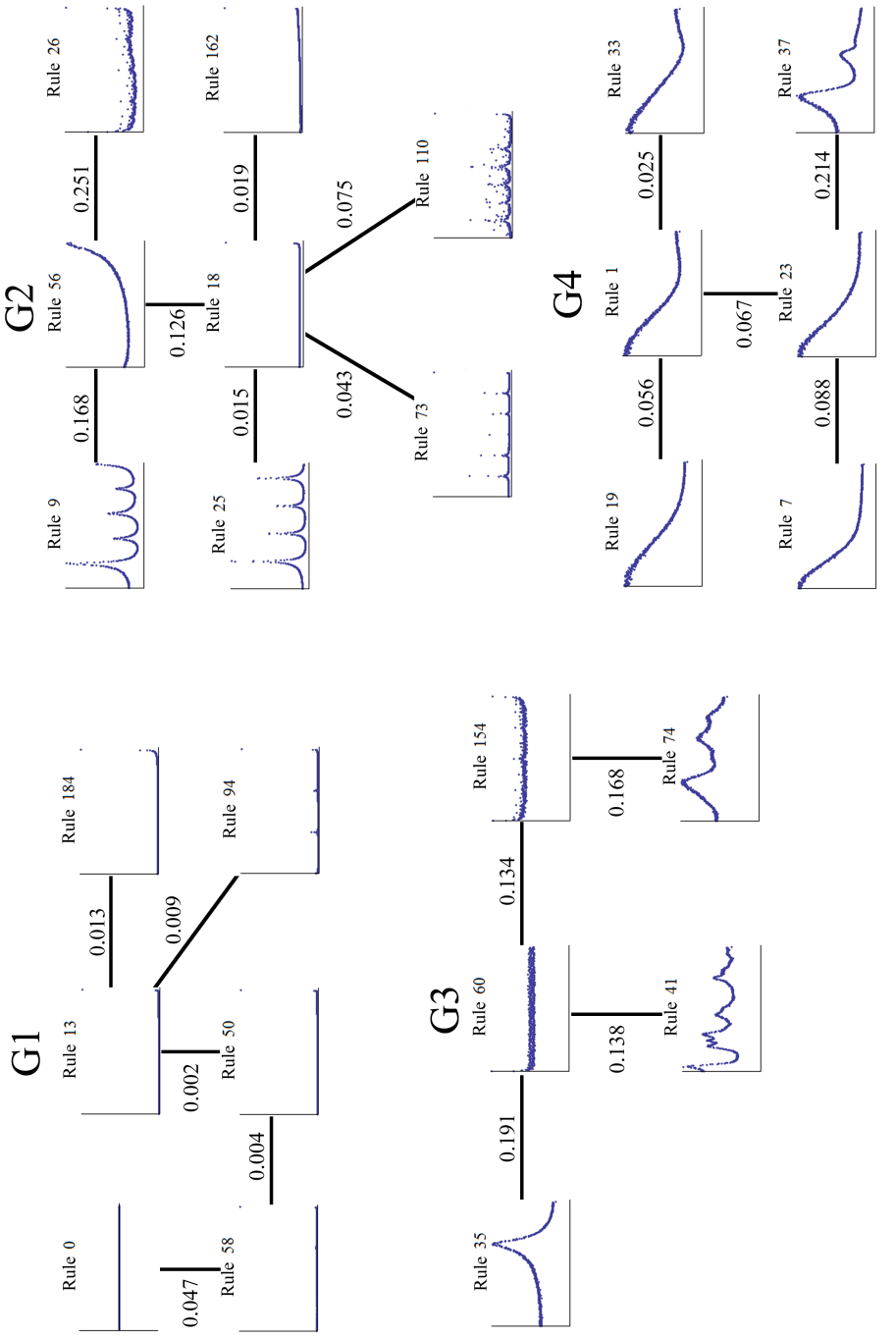


Fig. 14. Groups (1 through 4) of spectral classes, under NPBC

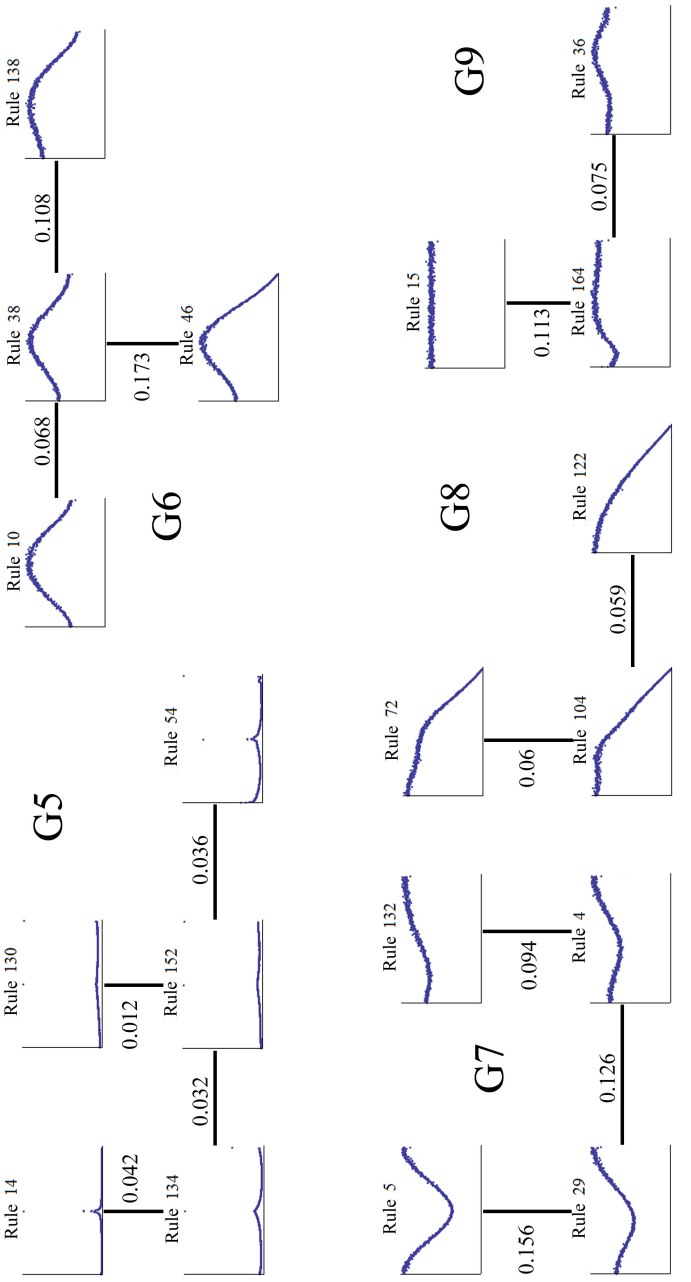


Fig. 15. Groups (5 through 9) of spectral classes, under NPBC

under NPBC display greater variation over time evolution than those under PBC, the same occurs to the spectra. Therefore, it is less likely for cycles of connections to be formed under NPBC than under PBC; consequently, the cycles formed under NPBC tend to be larger than the ones under PBC, thus yielding less groups of connected classes.

The groups of the graph are shown below (Figures 14 to 16) and the graph of the connections among groups is presented in Figure 13.

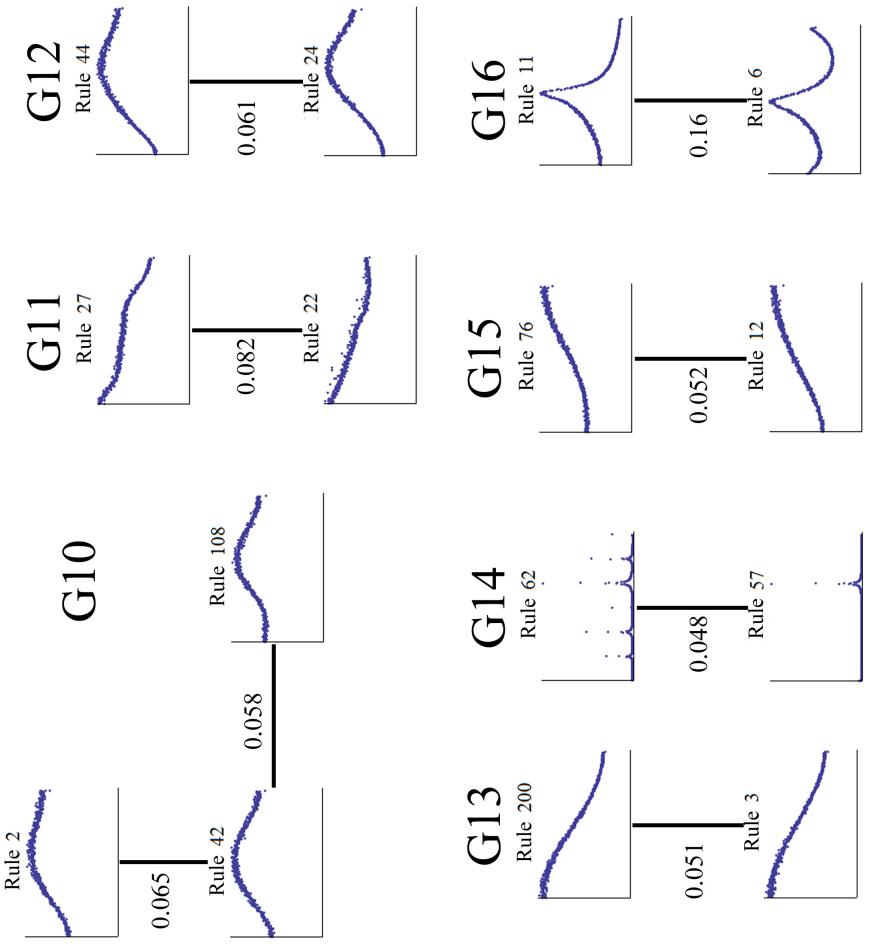


Fig. 16. Groups (10 through 16) of spectral classes, under NPBC

6 Concluding Remarks

The Fourier spectra of binary cellular automata rules give a description not only of the possible binary blocks of limit configurations obtained out of the time evolution of a rule, but also a quantitative characterisation of how binary sequences of distinct frequencies of bit changes are represented, in average, in the limit configurations obtained from each rule.

Here, a meaning for the Fourier spectrum was given, related to the dynamical behaviour of cellular automata and the possible configurations generated by their rules; also, reasons for the spectral similarity among ECA rules have been made explicit. The partition of the ECA rule space under PBC, at first based on the distance between the spectra, yielded 59 spectral classes, which were formalised in terms of dynamical behaviour, surjectivity of rules and rules with equivalent behaviour; since every set of dynamically equivalent rules is a subset of a spectral class, the partition obtained due to spectral similarity is coarser than the one arising from dynamical equivalence.

Also, a partition of the ECA rule space under non-periodic boundary conditions was presented in terms of spectral classes, resulting in 59 classes. For both PBC and NPBC, portraits of the ECA rule space in terms of graphs representing the spectral similarity among rules were given. Under NPBC, the surjective rules were subdivided into distinct spectral classes, in contrast to the PBC case; also, the graph generated under NPBC has a smaller number of groups of connected classes (16) than the one generated under PBC (19). The larger diversity of rule behaviour under NPBC makes the cycles of connections generated during the computation of the graph to be larger and more scarce than the ones generated under PBC, thus entailing that the NPBC graph has a smaller number of groups of connected classes than the PBC graph. These facts show that the spectrum of a rule display greater variation under NPBC than under PBC.

The partition method used here to compute the spectral classes relied upon the euclidean distance between the Fourier spectra of the rules, followed by a visual coherence check; but while this procedure makes sense for the small ECA space, it is unfeasible in larger spaces. In these cases, it would thus be necessary to develop means for automatic checking of the validity of a partition, which might even lead to investigations of other distance measures. Hence, the determination of an appropriate method to define a partition based only upon similarity between the Fourier spectra of the rules is a key point for further investigations.

Computing spectral similarity of rules due to symmetries and dynamical equivalence seems to be a way to establish a partial division of general cellular automata rule spaces in spectral classes.

The fact that every surjective ECA rule under PBC has shown white-noise type spectrum is an interesting point that could be investigated in other one-dimensional rule spaces, as a criterion to the automatic search for surjective, and even reversible rules in those spaces.

Finally, it would also be fruitful to be able to relate the Fourier spectra and spectral similarity to static rule parameters, such as the ones in [3], [14], [10], [16], and [8].

Acknowledgements. We are thankful to a grant provided by CAPES, the agency of the Brazilian Ministry of Education, and by a travel grant provided by FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo, that allowed us to present a preliminary version of the work at the AUTOMATA 2012 workshop.

References

[1] Hedlund, G.A.: Endomorphisms and automorphisms of the shift dynamical system. Mathematical Systems Theory 3(4), 320–375 (1969)
[2] Li, W.: Power spectra of regular languages and cellular automata. Complex Systems 1, 107–130 (1987)
[3] Li, W.: Phenomenology of nonlocal cellular automata. Journal of Statistical Physics 68(5-6), 829–882 (1992)
[4] Li, W., Packard, N.: The structure of the elementary cellular automata rule space. Structure 4, 281–297 (1990)
[5] Ninagawa, S.: $1/f$ Noise in Elementary Cellular Automaton Rule 110. In: Calude, C.S., Dinneen, M.J., Păun, G., Rozenberg, G., Stepney, S. (eds.) UC 2006. LNCS, vol. 4135, pp. 207–216. Springer, Heidelberg (2006)
[6] Ninagawa, S.: Evolution of One-Dimensional Cellular Automata by $1/f$ Noise. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 905–914. Springer, Heidelberg (2007)
[7] Ninagawa, S.: Power spectral analysis of elementary cellular automata. Complex Systems 17(4), 399–411 (2008)
[8] Oliveira, G.M., de Oliveira, P.P.B., Omar, N.: Definition and application of a five-parameter characterization of one-dimensional cellular automata rule space. Artificial Life 7(3), 277–301 (2001)
[9] Ruivo, E.L.P., de Oliveira, P.P.B.: Spectral similarity among elementary cellular automata. In: Formenti, E. (ed.) Proc. of AUTOMATA 2012: 18th Int. Workshop on Cellular Automata and Discrete Complex Systems, pp. 89–98 (2012)
[10] Shranko, A., de Oliveira, P.P.B.: Relationships between local dynamics and global reversibility of multidimensional cellular automata with hyper-rectangular neighbourhoods (unpublished manuscript)
[11] Wolfram, S.: Computation theory of cellular automata. Communications in Mathematical Physics 96(1), 15–57 (1984)
[12] Wolfram, S.: Twenty problems in the theory of cellular automata. Physica Scripta 9(3), 170–183 (1985)
[13] Wolfram, S.: A New Kind of Science. Wolfram Media (2002)
[14] Wuensche, A.: Classifying cellular automata automatically: finding gliders, filtering and relating space-time patterns, attractor basins, and the Z-parameter. Complexity 4, 73–90 (1998)
[15] Xie, H.: Distinct excluded blocks and grammatical complexity of dynamical systems. Complex Systems 9, 73–90 (1995)
[16] Zwick, M., Shu, H.: Set-theoretic reconstructability of elementary cellular automata. General Systems 1, 1–6 (1995)

Chapter 17
Wolfram's Classification and Computation in Cellular Automata Classes III and IV

Genaro J. Martínez[1], Juan C. Seck-Tuoh-Mora[2], and Hector Zenil[3]

[1] Unconventional Computing Center, Bristol Institute of Technology,
University of the West of England, Bristol, UK
Departamento de Ciencias e Ingeniería de la Computación,
Escuela Superior de Cómputo, Instituto Politécnico Nacional, México
genaro.martinez@uwe.ac.uk
[2] Área Académica de Ingeniería
Universidad Autónoma del Estado de Hidalgo, México
jseck@uaeh.edu.mx
[3] Behavioural and Evolutionary Theory Lab
Department of Computer Science, University of Sheffield, UK
h.zenil@sheffield.ac.uk

Abstract. We conduct a brief survey on Wolfram's classification, in particular related to the computing capabilities of Cellular Automata (CA) in Wolfram's classes III and IV. We formulate and shed light on the question of whether Class III systems are capable of Turing-completeness or may turn out to be "too hot" in practice to be controlled and programmed. We show that systems in Class III are indeed capable of computation and that there is no reason to believe that they are unable, in principle, to reach Turing universality.

Keywords: cellular automata, universality, unconventional computing, complexity, gliders, attractors, Mean field theory, information theory, compressibility.

1 Wolfram's Classification of Cellular Automata

A comment in Wolfram's *A New Kind of Science* gestures toward the first difficult problem we will tackle (ANKOS) (page 235): *trying to predict detailed properties of a particular cellular automaton, it was often enough just to know what class the cellular automaton was in.* The second problem we will take on concerns the possible relation between complexity of Cellular Automata and Turing universal computation, also highlighted by Wolfram in his ANKOS (page 691– on Class 4 behaviour and Universality): *I strongly suspect that it is true in general that any cellular automaton which shows overall class 4 behaviour will turn out—like Rule 110—to be universal.* The classification and identification of cellular automata (CA) has become a central focus of research in the field. In

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 237–259.
DOI: 10.1007/978-3-642-35482-3_17

[108], Stephen Wolfram presented his now well-known *classes*. Wolfram's analysis included a thorough study of one-dimensional (1D) CA, order $(k = 2, r = 2)$ (where $k \in \mathcal{Z}^+$ is the cardinality of the finite alphabet and $r \in \mathcal{Z}^+$ the number of neighbours), and also found the same classes of behaviour in other CA rule spaces. This allowed Wolfram to generalise his classification to all sorts of systems in [114].

An Elementary Cellular Automaton (ECA) is a finite automaton defined in a 1D array. The automaton assumes two states, and updates its state in discrete time according to its own state and the state of its two closest neighbours, all cells updating their states synchronously.

Wolfram's classes can be characterised as follows:

- Class I. CA evolving to a homogeneous state
- Class II. CA evolving periodically
- Class III. CA evolving chaotically
- Class IV. Includes all previous cases, known as a class of *complex rules*

Otherwise explained, in the case of a given CA:

- If the evolution is dominated by a unique state of its alphabet for any random initial condition, then it belongs to *Class I*.
- If the evolution is dominated by blocks of cells which are periodically repeated for any random initial condition, then it belongs to *Class II*.
- If for a long time and for any random initial condition, the evolution is dominated by sets of cells without any defined pattern, then it belongs to *Class III*.
- If the evolution is dominated by non-trivial structures emerging and travelling along the evolution space where uniform, periodic, or chaotic regions can coexist with these structures, then it belongs to *Class IV*. This class is frequently tagged: *complex behaviour*, *complexity dynamics*, or simply *complex*.

Fig. 1 illustrates Wolfram's classes, focusing on a specific ECA evolution rule (following Wolfram's notation for ECA [107]). All evolutions begin with the same random initial condition. Thus, Fig. 1a displays ECA Rule 32 converging quickly to a homogeneous state, Class I. Figure 1b displays blocks of cells in state one which evolve periodically showing a leftward shift, Class II. Figure 1c displays a typical chaotic evolution, where no pattern can be recognised or any limit point identified, Class III. Finally, Fig. 1d displays the so called complex class or Class IV. Here we see non-trivial patterns emerging in the evolution space. Such patterns possess a defined form and travel along the evolution space. They interact (collide), giving rise to interesting reactions such as annihilations, fusions, solitons and reflections, or they produce new structures. These patterns are referred to as *gliders* in the CA literature ('glider' is a widely accepted concept popularised by John Conway through his well-known additive binary 2D CA, the *Game of Life* (GoL) [34]). In Class IV CA we see regions with periodic evolutions and chaos, and most frequently in complex rules the background is dominated by stable states, such as in GoL. In such cases—and this is particularly true of the

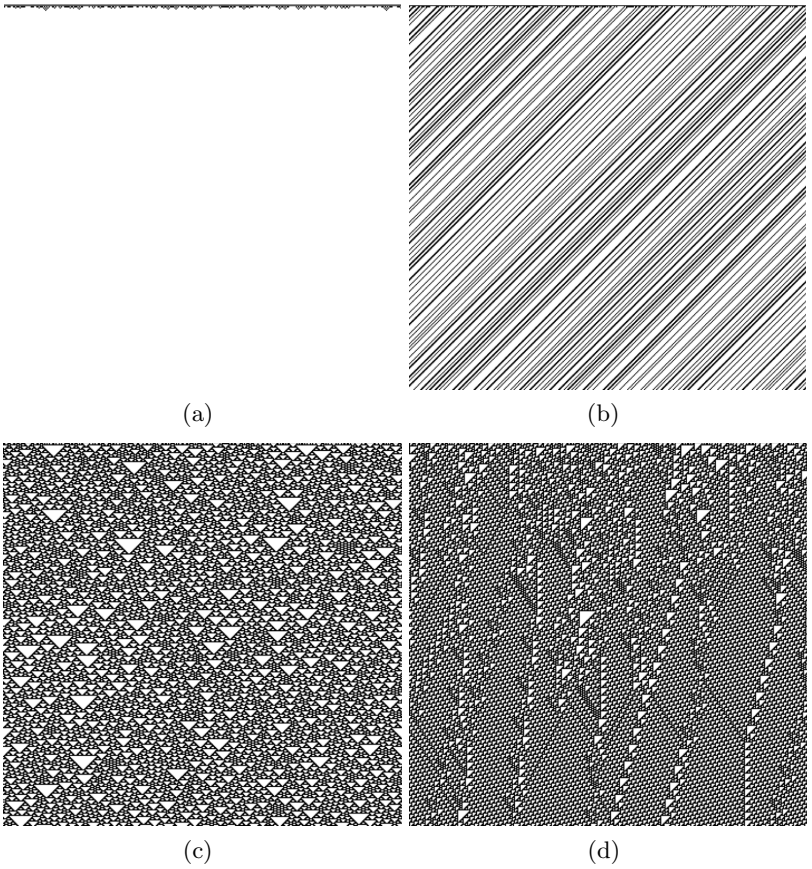


Fig. 1. Wolfram's classes represented by ECA rules: (a) Class I - ECA Rule 32, (b) Class II - ECA Rule 10, (c) Class III - ECA Rule 126, (d) Class IV - ECA Rule 110. We have the same initial condition in all these cases, with a density of 50% for state 0 (white dots) and state 1 (black dots). The evolution space begins with a ring of 358 cells for 344 generations.

complex ECA Rule 110–the CA can evolve with a periodic background (called ether) where these gliders emerge and live. Gliders in GoL and other CAs such as the 2D Brian's Brain CA [99] caught the attention of Christopher Langton, spurring the development of the field of *Artificial Life* (AL) [51, 52].

Since the publication of the paper "Universality and complexity in cellular automata" in 1984 [108], qualitative classifications of CA (an other systems) have been a much studied and disputed subject. Wolfram advanced several ECA rules as representatives for each of his classes and despite an early comment suggesting that (page 31): $k = 2, r = 1$ *cellular automata are too simple to support universal computation*, in his book "Cellular Automata and Complexity" [112] ECA Rule 110 was granted its own appendix (Table 15, Structures in Rule 110, pages 575–577). It contains specimens of evolutions, including a list of thirteen gliders

compiled by Doug Lind, and also presents the conjecture that the rule could be universal. Wolfram writes: *One may speculate that the behaviour of Rule 110 is sophisticated enough to support universal computation.*

An interesting paper written by Karel Culik II and Sheng Yu titled "Undecidability of CA Classification Schemes" [26, 94] discussed the properties of such classes, concluding that: *it is undecidable to which class a given cellular automaton belongs* (page 177). Indeed, in 1984 Wolfram [108] commented (page 1): *The fourth class is probably capable of universal computation, so that properties of its infinite time behaviour are undecidable.* Actually, we can see that no effective algorithm exists that is capable of deciding whether a CA is complex or universal, and so far only a few discovered (as opposed to constructed) cellular automata have been proven to be capable of universal computation (notably Wolfram's Rule 110 and Conway's Game of Life). However some techniques offer suitable approximations for finding certain sets of complex, though perhaps not necessarily universal rules (under Wolfram's PCE they would be, c.f. Section 4). In [20], Israeli and Goldenfeld devised a coarse-grained technique to find predictable properties of elementary CA and other systems. While they were able to reduce elementary CAs in all Wolfram's classes, they were unable to do so for some in Class III (rules 30, 45, 106 and their symmetries) and more surprisingly in Class II (rule 154 and its symmetries). Their technique showed to be able to find properties of CA at some coarse-grained level of description without accounting for small-scale details. They show that by using this technique one can take a Class III system to a Class I in order to predict some properties of the original system by a reduction of its apparent complexity, pointing out that irreducibility may not be the same as complexity (or universality) given that some irreducible rules can be coarse-grained (at least one example of an irreducible rule (110) is known for certain because its ability of Turing universality). This seems in agreement with the fact that systems in Class IV seem to show more persistent structures than systems in Class III. In "Local structure theory for cellular automata" [40] Howard Gutowitz developed a statistical analysis. An interesting schematic diagram conceptualising the umbral of classes of CA was offered by Wentian Li and Norman Packard in "The Structure of the Elementary Cellular Automata Rule Space" [54]. Pattern recognition and classification has been examined in "Toward the classification of the patterns generated by one-dimensional cellular automata" [13] by Yoji Aizawa and Ikuko Nishikawa. An extended analysis by Andrew Adamatzky under the heading "Identification of Cellular Automata" in [2] considered the problem of how, given a sequence of configurations of an unknown cellular automaton, one may reconstruct its evolution rules. A recent special issue dedicated to this problem focuses on some theoretical and practical results.[4] Klaus Sutner has discussed this classification and also the principle of computational equivalence in "Classification of Cellular Automata" [96], with an emphasis on Class IV or computable CA. An

[4] Special issue "Identification of Cellular Automata", *Journal of Cellular Automata* **2(1)**, 1–102, 2007.
`http://www.oldcitypublishing.com/JCA/JCAcontents/JCAv2n1contents.html`

interesting approach involving an additive 2D CA was described in David Eppstein's classification scheme [30][5].

We will discuss some practical and theoretical topics that distinguish such classes and explore the computing properties of CA rules, in particular in classes III and IV. Among the topics we want to explore is the feasibility of using extended analog computers (EAC) [76] for CA construction, in order to obtain unconventional computing models [4, 3]. In this classification, Class IV is of particular interest because the rules of the class present non-trivial behaviour, with a rich diversity of patterns emerging, and non-trivial interactions between gliders, plus mobile localizations, particles, or fragments of waves. This feature was useful in implementing a register machine in GoL [17] to determine its universality. First we survey some of the approximations that allow the identification of complex properties of CA and other systems.

1.1 Mean Field Approximation

The Mean field theory is a well-known technique for discovering the statistical properties of CA without analysing the evolution space of individual rules. It has been used extensively by Gutowitz in [42]. The method assumes that states in Σ are independent and do not correlate with each other in the local function φ. Thus we can study probabilities of states in a neighbourhood in terms of the probability of a single state (the state in which the neighbourhood evolves), and the probability of a neighbourhood would be the product of the probabilities of each cell in it.

Harold V. McIntosh in [67] presents an explanation of Wolfram's classes using a mixture of probability theory and de Bruijn diagrams[6], resulting in a classification based on the mean field theory curve:

- Class I: monotonic, entirely on one side of diagonal;
- Class II: horizontal tangency, never reaches diagonal;
- Class IV: horizontal plus diagonal tangency, no crossing;
- Class III: no tangencies, curve crosses diagonal.

For the one-dimensional case, all neighbourhoods are considered, as follows:

$$p_{t+1} = \sum_{j=0}^{k^{2r+1}-1} \varphi_j(X) p_t^v (1-p_t)^{n-v} \tag{1}$$

[5] For a discussion see Tim Tyler's CA FAQ at `http://cafaq.com/classify/index.php`, and more recently, a compression-based technique inspired by algorithmic information theory has been advanced[119] that offers a powerful method for identifying complex CA and other complex systems

[6] The de Bruijn diagrams have been culled from Masakazu Nasu's 1978 work on tessellation automata [83]. Wolfram himself has explored some of this in [109], later thoroughly analysed by McIntosh [68, 73], Sutner [95], Burton Voorhees [102, 103], and, particularly, exploited to calculate reversible 1D CA using de Bruijn diagrams derived from the Welch diagrams by Seck-Tuoh-Mora in [89, 91]

such that j indexes every neighbourhood, X are cells $x_{i-r}, \ldots, x_i, \ldots, x_{i+r}$, n is the number of cells in every neighbourhood, v indicates how often state '1' occurs in X, $n - v$ shows how often state '0' occurs in the neighbourhood X, p_t is the probability of a cell being in state '1', and q_t is the probability of a cell being in state '0'; i.e., $q = 1 - p$. For Mean field theory in other lattices and dimensions, please consult[41, 43].

1.2 Basins of Attraction Approximation

Andrew Wuensche, together with Mike Lesser, published a landmark book entitled "The Global Dynamics of Cellular Automata" in 1992 [105] which contained a very extended analysis of attractors in ECA. Wolfram himself had explored part of these cycles in "Random Sequence Generation by Cellular Automata" [110], as had McIntosh in "One Dimensional Cellular Automata" [73]. Notably, Stuart Kauffman in his book "The Origins of Order: Self-Organization and Selection in Evolution" [50] applies basins of attraction to sample random Boolean networks (RBN) in order to illustrate his idea that RBN constitute a model of the gene regulatory network, and that cell types are attractors. The best description of such an analysis is to be found in [117]. A basin (of attraction) field of a finite CA is the set of basins of attraction into which all possible states and trajectories will be organized by the local function φ. The topology of a single basin of attraction may be represented by a diagram, the *state transition graph*. Thus the set of graphs composing the field specifies the global behaviour of the system [105]. Generally a basin can also recognize CA with chaotic or complex behaviour using prior results on attractors [105]. Thus, Wuensche says that Wolfram's classes can be represented as a *basin classification* [105], as follows:

- Class I: very short transients, mainly point attractors (but possibly also periodic attractors), very high in-degree, very high leaf density (very ordered dynamics);
- Class II: very short transients, mainly short periodic attractors (but also point attractors), high in-degree, very high leaf density;
- Class IV: moderate transients, moderate-length periodic attractors, moderate in-degree, very moderate leaf density (possibly complex dynamics);
- Class III: very long transients, very long periodic attractors, low in-degree, low leaf density (chaotic dynamics).

1.3 Compressibility Approximation

A compression-based classification of CA (and other systems) was proposed in [119], based on the concept of algorithmic (Kolmogorov) complexity. Unlike the Mean field theory, this technique analyses the asymptotic statistical properties of CA by looking at full space-time evolution of individual rules up to an arbitrary number of steps. The method produces the following variation of Wolfram's classification [120].

- Class I: highly compressible evolutions for any number of steps;
- Class II: highly compressible evolutions for any number of steps;
- Class III: the lengths of compressed evolutions asymptotically converge to the uncompressed evolution lengths;
- Class IV: the lengths of compressed evolutions asymptotically converge to the uncompressed evolution lengths.

The four classes seem to give way to only two (Classes I and II and Classes III and IV are not distinguishable in this first approach). But it is shown how algorithmic information theory helps to separate them again, using the concept of asymptotic behaviour advanced in [119, 121].

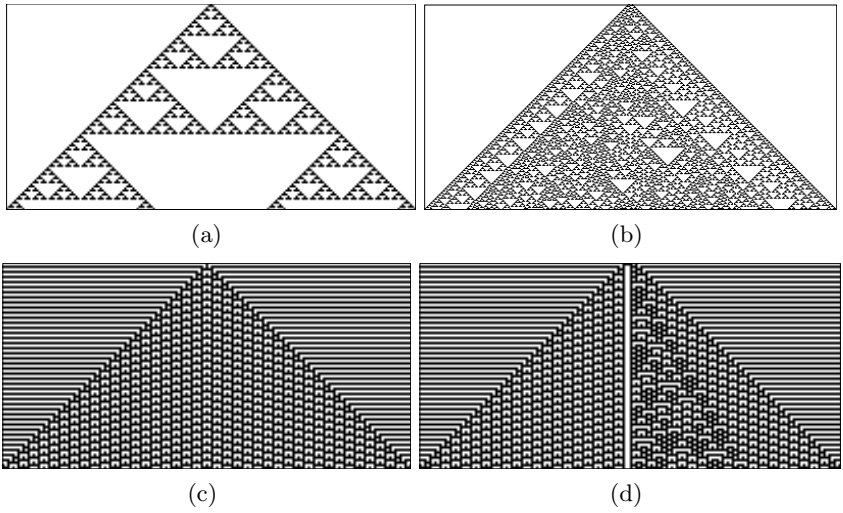


Fig. 2. To which of Wolfram's Classes do these two ECAs (Rule 22 and Rule 109) belong? (a) Wolfram's ECA Rule 22 starting from a single black cell, (b) Rule 22 starting from another initial configuration (11001), (c) Wolfram's ECA Rule 109 starting from a single black cell, (d) The same Rule 109 starting from another initial configuration (111101).

The motivation in [119] is to address one of the apparent problems of Wolfram's original classification, that of rules behaving in different ways starting from different initial configurations. In the experiments that led Wolfram to propose his classification he started the systems with a "random" initial configuration as a way to *sample* the behaviour of a system and circumvent the problem of having to choose a particular initial configuration to map a system to its possible class of behaviour. The problem resides in the fact that a CA, like any other dynamical system, may have phase transitions, behaving very differently for different initial configurations (the question is ultimately undecidable

as pointed out in [26]) but this is also a practical issue for an heuristic classification, given that systems may seem to jump from one class to another in such phase transitions. The chances of having a CA display an *average* behaviour (that is, its behaviour for *most* initial configurations) are greater when taking a "random" initial configuration, only if one assumes that there is no bias towards any particular region of the possible enumerations of initial configurations (consider the behaviour of a CA starting from one initial configuration versus another (see Figures 2). In [119] this issue is addressed with the definition of a compression-based phase transition coefficient capturing the *asymptotic behaviour* of a system, which in turn allows to separate the collapsed classes and even advance a different and alternative classification, based on the sensitivity of a CA to its initial conditions, which has also been conjectured to be related to the system's ability to transfer information, and ultimately to its computing abilities, particularly as these relate to Turing universal computation (see [121]). This approach does not solve the problem of a system that behaves in a qualitatively different manner after a certain number of initial input configurations or after a certain period of time (the same problem encountered when devising the original classification), which is not a problem of method, but is instead related to the general problem of induction and of reachability (hence to undecidability in general). Nonetheless it does address the problem of a reasonable definition of the "average behaviour" of a system (in this case a CA) under the same assumptions made for other enumerations (viz. that enumerations, especially natural ones, have no distinct regions where a system starts behaving in a completely different fashion, making it impossible to talk about the convergence in behaviour of a system). Wolfram's classes can once again be separated using the compression-based approach in combination with the following classification [120], derived from a phase transition coefficient presented in [119]:

- Class I: insensitivity to initial configurations, inability to transfer information other than isolated bits;
- Class II: sensitivity to initial conditions, ability to transfer some information;
- Class III: insensitivity to initial configurations, inability to transfer information, perhaps due to lack of (evident means of) control;
- Class IV: sensitivity to initial conditions, ability to transfer some information.

One can only understand how Classes I and III can now be together in this classification on the basis of the qualitative treatment explained above. In other words, when one changes the initial configuration of a system in either of these two classes (I and III) the system's behaviour remains the same (each evolution is equally compressible), and it is therefore considered unable to or inefficient at transferring information or programming a CA to perform (universal) computation. On the other hand, this suggests that classes II and IV may be better at transferring information, even if they may do so in different ways. This classification may tell us that some classes are more sensitive to initial configurations.

Together, the compression-based classifications capturing different behaviours of the systems capture other intuitive notions that one would expect from Wolfram's original classification. The values for ECA calculated in [119] yielded results that also suggest that one may be able to relate these measures to universality through the definition of Class IV, as given above (see [121]).

2 Universal CA Class IV versus Class III

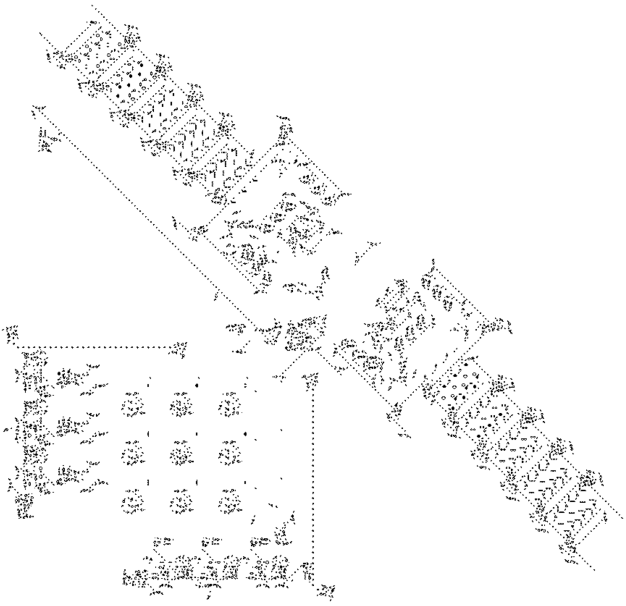
Karel Culik II and Sheng Yu have demonstrated [26] that whether a CA belongs to Class IV is undecidable. Nevertheless, some approximations have been developed, with interesting results. The use of genetic programming by Melanie Mitchell, Rajarshi Das, Peter Hraber, and James Crutchfield [75, 28] to obtain sets of rules with particles and computations is a case in point. As indeed is Emmanuel Sapin's calculation of a non-additive binary universal 2D CA with a genetic algorithm, the *R rule* [87, 88]. However, the use of evolutionary techniques has been limited to a small portion of complex CA with few states and small configurations. Up to now, brute force programming has been necessary to obtain monsters of complex patterns in huge spaces, as Eppstein shows in [31].

2.1 The Game of Life: Class IV

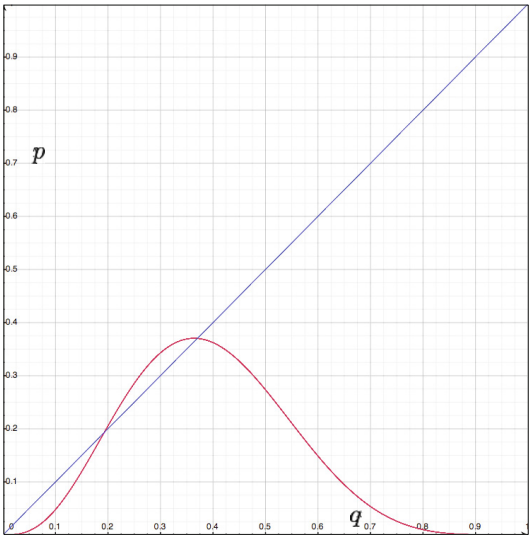
The most popular 2D CA is certainly Conway's Game of Life (GoL), a binary 2D additive CA, first published in Martin Garden's column in *Scientific American* [34]. GoL can be represented as $R(2,3,3,3)$, or typically, as the $B3/S23$ rule.[7] In 1982, Conway proved that GoL was universal by developing a register machine working with gliders, glider guns, still life and oscillator collisions [17]. However, such universality was completed by Paul Rendell's demonstration in 2000 that involved implementing a 3-state, 3-symbol Turing machine in GoL [85, 86]. The machine duplicates a pattern of 1's within two 1's on the tape to the right of the reading position, running 16 cycles to stop with four 1's on the tape. A snapshot of this implementation is provided in Fig. 3a. For details about each part and about the functionality of this machine please visit "Details of a Turing Machine in Conway's Game of Life" `http://rendell-attic.org/gol/tmdetails.htm`.

GoL is a typical Class IV CA evolving with complex global and local behaviour. In its evolution space we can see a number of complex patterns which emerge from different configurations. GoL has been studied since 1969 by Conway, and William Gosper of MIT's Artificial Life research group has taken a strong interest in it. The tradition of GoL research is very much alive, with today's GoL researchers discovering new and very complex constructions by running complicated algorithms. Just last year, GoL celebrated its 40th anniversary.

[7] An excellent forum on GoL is "LifeWiki" `http://conwaylife.com/wiki/index.php?title=Main_Page`. To complement this, you may consult "The Game of Life Sites" `http://uncomp.uwe.ac.uk/genaro/Cellular_Automata_Repository/Life.html`.



(a)



(b)

Fig. 3. (a) A 3-state, 3-symbol Turing machine in GoL by Rendell [85, 86], (b) its mean field curve

The occasion was marked by the publication of the volume "Game of Life Cellular Automata" [6], summarising a number of contemporary and historical results in GoL research as well as work on other interesting Life-like rules.

According to Mean field theory, p is the probability of a cell's being in state '1' while q is its probability of its being in state '0' i.e., $q = 1 - p$, and the *mean field equation* represents the neighbourhood that meets the requirement for a live cell in the next generation [67]. As we have already seen, horizontal plus diagonal tangency, not crossing the identity axis (diagonal), and the marginal stability of the fixed point(s) due to their multiplicity indicates Wolfram's Class IV [42], or complex behaviour. Hence, we will review the global behaviour of GoL using Mean field theory. Figure 3b shows the mean field curve for GoL, with polynomial:

$$p_{t+1} = 28p_t^3 q_t^5 (2p_t + 3q_t).$$

The origin is a stable fixed point, while the unstable fixed point $p = 0.2$ represents the fact that densities around 20% induce complex behaviour for configurations in such a distribution. $p = 0.37$ is the maximum stable fixed point where GoL commonly reaches global stability inside the evolution space.

In [122] a compression-based phase transition coefficient was calculated, showing that, as expected, GoL exhibits a high degree of variability and potential (efficient) programmability. This is in agreement with the known fact that GoL is capable of universal computation, and hence supports the idea that sensitivity to initial configurations is deeply connected to both programmability and (Turing) universality.

2.2 Life-Like Rule $B35/S236$: Class III

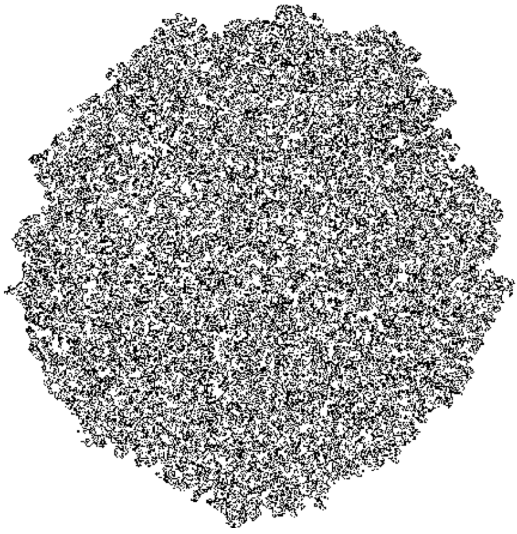
The Life-like CA evolution rule $B35/S236$ was proposed by Eppstein and Dean Hickerson as a chaotic CA with sufficient elements for developing universality. Details about these computable elements are available at `http://www.ics.uci.edu/~eppstein/ca/b35s236/construct.html`. The family of gliders and other complex constructions in this rule can be found at `http://www.ics.uci.edu/~eppstein/ca/b35s236/`.

The $B35/S236$ automaton commonly evolves chaotically. Figure 4a displays a typical chaotic evolution starting from an L-pentomino configuration; after 1,497 generations there is a population of 52,619 live cells. Here we see how a few gliders emerge from chaos and then quickly escape, although the predominant evolution over a long period is chaotic.

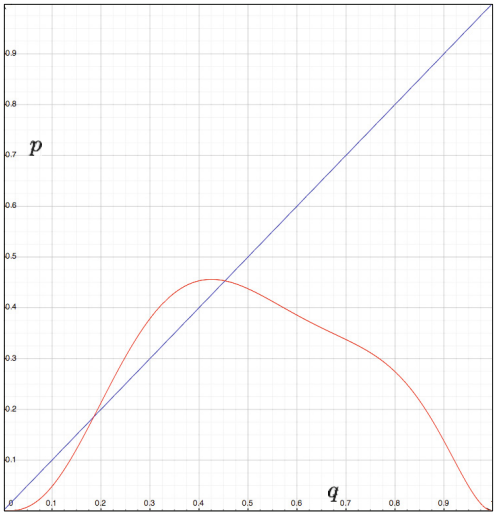
Figure 4b shows the mean field curve for CA $B35/S236$, with polynomial:

$$p_{t+1} = 28p_t^3 p_t^2 (p_t^4 + 2p_t q_t^3 + 2p_t^2 q_t^2 + 3q_t^4).$$

The origin is a stable fixed point (as in GoL) which guarantees the stable configuration in zero, while the unstable fixed point $p = 0.1943$ (again very similar to GoL) represents densities where we could find complex patterns emerging in



(a)



(b)

Fig. 4. (a) Evolution starting from an L-pentomino in Life-like CA $B35/S236$, (b) its mean field curve

$B35/S236$. $p = 0.4537$ is the maximum stable fixed point at which $B35/S236$ commonly reaches global stability.

This way, $B35/S236$ preserves the diagonal tangency between a stable and an unstable fixed point on its mean field curve. But although its values are close to those of GoL, CA $B35/S236$ has a bigger population of live cells, which

is not a sufficient condition for constructing reliable organisms from unreliable components.

2.3 ECA Rule 110: Class IV

The 1D binary CA rule numbered 110 in Wolfram's system of classification [107] has been the object of special attention due to the structures or gliders which have been observed in instances of its evolution from random initial conditions. The rule is assigned number 110 in Wolfram's enumeration because it represents the decimal base of the transition rule expanded in binary: 01110110. The transition function evaluates the neighbourhoods synchronously in order to calculate the new configuration transforming the neighbourhoods 001, 010, 011, 101 and 011 into state 1 and the neighbourhoods 000, 100 and 111 into state 0. It has been suggested that Rule 110 belongs to the exceptional Class IV of automata whose chaotic aspects are mixed with regular patterns. But in this case the background where the chaotic behaviour occurs is textured rather than quiescent, a tacit assumption in the original classification.[8] Rule 110 was granted its own appendix (Table 15) in [110]. It contains specimens of evolution including a list of thirteen gliders compiled by Lind and also presents the conjecture that the rule could be universal.

The literature on the origins of Rule 110 includes a statistical study done by Wentian Li and Mats Nordahl in 1992 [53]. This paper studies the transitional role of Rule 110 and its relation to Class IV rules figuring between Wolfram's classes II and III. The study would seem to reflect an approach to equilibrium statistics via a power law rather than exponentially. Matthew Cook wrote an eight page introduction [21] listing gliders from A through H and a glider gun.[9]. This list shows new gliders which do not appear on Lind's list, gliders with rare extensions, and a pair of gliders of complicated construction, including an amazing glider gun. Cook makes a comparison between Rule 110 and Life, finding some similarities in the behaviour of the two evolution rules and suggesting that Rule 110 may be called "LeftLife."

Looking at the rule itself, one notices a ubiquitous background texture which Cook calls "ether," although it is just one of many regular stable lattices capable of being formed by the evolution rule, and can be obtained quickly using the de Bruijn diagrams [70, 64]. McIntosh raises the issue of the triangles of different sizes that cover the evolution space of Rule 110 [71]. The appearance of these triangles suggests the analysis of the plane generated by the evolution of Rule 110 as a two dimensional shift of finite type. This suggestion is arrived at by observing that the basic entities in the lattices, the unit cells, induce the formation of upside-down isosceles right triangles of varying sizes. The significance of Rule 110 could lie in the fact that it is assembled from recognisably distinct tiles,

[8] A repository of materials on ECA Rule 110 can be found at:
`http://uncomp.uwe.ac.uk/genaro/Rule110.html`

[9] An extended list of gliders in Rule 110 is provided in
`http://uncomp.uwe.ac.uk/genaro/rule110/glidersRule110.html`

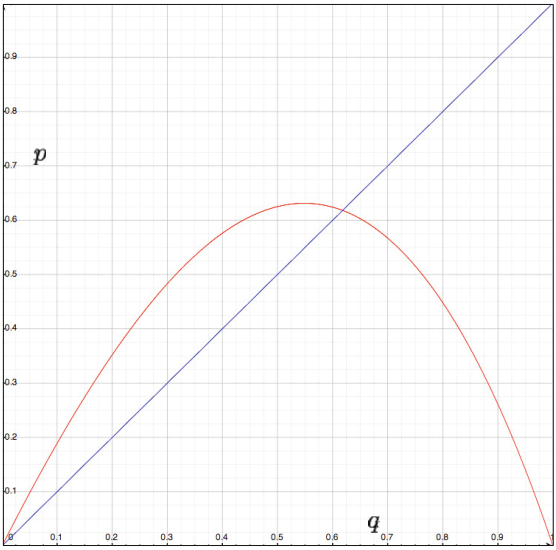


Fig. 5. Mean field curve for ECA Rule 110

and hence its evolution can be studied as a tiling problem, in the sense of Hao Wang [37]. It may even be possible to see fitting elements of one lattice into another as an instance of Emil L. Post's correspondence principle [27], which would establish the computational complexity of the evolution rule [70].

The most important result both in the study of Rule 110 and in CA theory over the last twenty years, is the demonstration that Rule 110 is capable of universal computation [22, 114, 72, 23, 65]. For so a type of system called a cyclic tag system (CTS) as a variation of a well-known model of computation (Post's tag systems) was designed to be of use for the proof and its characteristic restrictions: 1D, boundary conditions, package of gliders, and multiple collisions. CTS are a new kind of computing formalism [22, 114] used as tools for implementing computations in Rule 110.

Fig. 5b shows the mean field curve for Rule 110 with polynomial:

$$p_{t+1} = 2p_t q_t^2 + 3p_t^2 q_t.$$

The origin of Fig. 5 displays a stable fixed point (as in GoL) which guarantees the stable configuration in zero. The maximum point ($p = 0.6311$) is close to the fixed stable point in $p = 0.62$. In Rule 110 we cannot find unstable fixed points, and in any case the emergence of complex structures is ample and diverse.

A basin (of attraction) field of a finite CA is the set of basins of attraction into which all possible states and trajectories will be organised by the local function φ. The topology of a single basin of attraction may be represented by a diagram, the *state transition graph*. Thus the set of graphs composing the field specifies the global behaviour of the system [105].

As calculated in [119], rules such as Rule 110 and Rule 54 (also believed to be capable of universal computation) had a large compression-based phase transition coefficient, as discussed in Section 1.3, meaning that their ability to transfer information was well captured by the measure defined in [119] (and, interestingly, perhaps strengthens the belief that Rule 54 is capable of Turing universality).

3 *Heat* and *Programmability* in Class III

Class III CAs may turn out to be too sensitive, so the question may be whether even if they are that sensitive they can carry information from one side to another. Universality results in simple programs capable of complicated behaviour have traditionally relied on localised structures ("particles") well separated by relatively uniform regions. It could also be the case that proofs of universality of seemingly Class IV systems are easier to construct because of its "particle-like" behaviour, unlike systems seemingly in Class III.

The open problem is thus to prove computational universality in a simple program system for which an entropy measure on each time step remains near its maximum—e.g. 80% of its maximum theoretical value on at least 80% of its time steps. Can a "hot system" of this sort perform meaningful computation?

In the Game of Life, for example, there is a common intuitive notion of *heat*[10], defined as the average number of cells which change state in each generation (note the strong connection of Shannon's Entropy and the Mean Field Theory). For example, *the heat* of a glider in GoL is known to be four, because two cells are born and two die in every generation, and that for a blinker is 4, because 2 cells are born and 2 die in every generation. In general, for a period n oscillator with an r-cell "rotor", the heat is at least $2r/n$, and no more than $r(1 - (n \bmod 2)/n)$.

The concept of *heat* can clearly be associated with Wolfram's chaotic Class III, where CAs, e.g., rule 30, change state at a very high rate, (see Figures (c) 1), which is what keeps them from developing persistent structures such as are seen in Rule 110 (see Figure (d) 1). The presence of persistent structures in Wolfram's Rule 110 and Conway's Game of Life is what traditionally has been used to perform computation–implementing logic gates or transferring information over time by putting particles in the way of interacting with each other. So the question is whether CAs such as the ones belonging to Wolfram's Class III are "too hot" to transfer information and are therefore, paradoxically in this particular way, just like Class I systems which are unable to perform computation. Alternatively, Class III may be able to perform computation, as has been suggested, but it may turn out to be difficult to program such systems (if not designed to "look" like a Class III system by using first a system from a Class IV, somehow *hiding* the computing capabilities of the Class III system), and this potential similarity between the insensitivity to initial conditions of Class I and Class III systems is what the compressibility approach discussed in Section 1.3 is

[10] See `http://www.argentum.freeserve.co.uk/life.htm` accessed in July 2012.

measuring and which has been advanced in [120] as a measure of *programmability*. Wolfram identified some of these issues in his enumeration of open problems in the research on CA [113] (problems 1, 2 and 14), concerning the connections between the computational and statistical characteristics of cellular automata, measures of entropy and complexity and how to improve his classification using dynamic systems (which was one of the motivations of [119]). Wolfram asks, for example, about the rate of information transmission of a CA in relation to its Lyapunov exponent (positive for Classes III and IV) and the computational power of these systems according to their classes. Another interesting question concerns the connection to Langton's λ parameter [51] and the ongoing investigation of its connections to some of the approaches described in this paper. In [15] a similar approach is taken using Lyapunov exponents and Jacobians–anticipated by Wolfram in [110]–where the calculation of the number of cells that differ provide a metric of the average rate of transmission of information (one that is related to the more informal term *heat* in GoL).

4 Final Remarks

Usually, Class III rules are not considered candidates for computational universality. However, in some cases such rules can support complex patterns, including performing complex computations. Exploring many CA rules, including the exceptionally chaotic Life-like rule *Life Without Death* [35], one finds that there are several rules between chaos and complexity which are not included within the domain of complex behaviour. However, they present many elements equally likely to reach Turing computational universality. An important point made in this survey and review is that it seems clearly to be the case that it is not only *complex CA*[11] rules that are capable of computation, and that CA, even if simple or random-looking, may support Turing universality. Whether the encoding to make them actually compute turns out to be more difficult than taking advantage of the common interacting persistent structures in rules usually believed to belong to Wolfram's class IV is an open question.

Previous results on universal CAs (developing signals, self-reproductions, gliders, collisions, tiles, leaders, etc.) prove that unconventional computing can be obtained depending on the nature of each complex system. For example, to prove universality in Rule 110 it was necessary to develop a new equivalent Turing machine to take advantage of limitations in 1D and the same dynamics in its evolution space, e.g., mobility of gliders and boundary properties. Hence, a CTS was devised, before this system was known as a circular machine [14, 49, 79, 66]. This way, the nature of each system would determine the best environment in which to design a corresponding computer. This could be the basis of Wolfram's *Principle of Computational Equivalence* and it is also the inspiration behind the

[11] A Complex Cellular Automata Repository with several interesting rules is available at `http://uncomp.uwe.ac.uk/genaro/otherRules.html`.
We particularly recommend Tim Hutton's *Rule Table Repository* `http://code.google.com/p/ruletablerepository/`.

definition of *programmability* measures for natural computation in [120]. Wolfram's *Principle of Computational Equivalence* ultimately only distinguishes between two kinds of behaviours (despite Wolfram's own heuristic classification), namely those that are "sophisticated" enough and reach *Wolfram's threshold*, constituting a class of systems capable of computational universality, and those that fall below this threshold and are incapable of universal computation. And indeed, the compression-based classification in [119] at first distinguishes only two classes.

A number of approximations were developed or adapted to find complex CA. Perhaps the most successful technique was the one developed by Wuensche, with its Z parameter [118]. Some attempts were made by Mitchell *et. al* using genetic algorithms, although they had a particular interest in finding rules able to support complex patterns (gliders) with computational uses [28, 116]. Unfortunately, these algorithms have strong limitations when it comes to searching in large rule spaces and very complex structures. And though the technique in [119] has proven capable of identifying complex systems with great accuracy, it requires very large computational resources to extend the method to larger rule spaces if a thorough investigation is desired (though in conjunction with other techniques it may turn out to be feasible).

As it has proven to be a very rich space, new kinds of CAs are proposed all the time. e.g., reversible CA [48, 90, 69], partitioned CA [114], hyperbolic CA [58], CA with non-trivial collective behaviour (self-organization) [24, 25], asynchronous CA [32], biodiversity in CA [55], CA with memory [9, 10], morphological diversity [12], identification of CA [2], communication complexity [29, 38], pattern recognition from CA [13], to mention a few.

Some other studies dedicated to designing or identifying universal CAs are [44, 3, 4, 36, 62]. Obtaining CA of Class IV from other rules has been studied via lattice analysis [39], with memory [56, 57, 62, 7, 11, 8], asynchronous [97, 100, 18, 32], differential equations [19], partitioned [74, 78, 46, 79, 80, 77, 60, 61], parity-filter CA [84, 93, 47], number-conserving [81] changing different neighbourhoods in CA [106].

CA as *super computer models* are developed extensively in [104, 20, 16, 59, 82, 99, 111, 92, 44, 98, 33, 3, 4, 5, 1, 115, 45, 66].

Acknowledgements. G. J. Martínez wants to thank support given by EPSRC grant EP/F054343/1, J. C. Seck-Tuoh-Mora wants to thank support provided by CONACYT project CB-2007-83554 and H. Zenil wants to thank support by the FQXi under grant number FQXi-MGA-1212.

References

[1] Adamatzky, A., Costello, B.L., Asai, T.: Reaction-Diffusion Computers. Elsevier (2005)
[2] Adamatzky, A.: Identification of Cellular Automata. Taylor & Francis (1994)
[3] Adamatzky, A.: Computing in Nonlinear Media and Automata Collectives. Institute of Physics Publishing, Bristol (2001)

[4] Adamatzky, A. (ed.): Collision-Based Computing. Springer (2002)
[5] Adamatzky, A.: New media for collision-based computing. In: Adamatzky, A. (ed.) Collision-Based Computing, pp. 411–442. Springer (2002)
[6] Adamatzky, A. (ed.): Game of Life Cellular Automata. Springer (2010)
[7] Alonso-Sanz, R.: Reversible Cellular Automata with Memory. Physica D 175, 1–30 (2003)
[8] Alonso-Sanz, R.: Elementary rules with elementary memory rules: the case of linear rules. Journal of Cellular Automata 1, 71–87 (2006)
[9] Alonso-Sanz, R.: Cellular Automata with Memory. Old City Publishing (2009)
[10] Alonso-Sanz, R.: Discrete Systems with Memory. World Scientific Series on Nonlinear Science, Series A (2011)
[11] Alonso-Sanz, R., Martin, M.: Elementary CA with memory. Complex Systems 14(2), 99–126 (2003)
[12] Adamatzky, A., Martínez, G.J.: On generative morphological diversity of elementary cellular automata. Kybernetes 39(1), 72–82 (2010)
[13] Aizawa, Y., Nishikawa, I.: Toward the classification of the patterns generated by one-dimensional cellular automata. In: Ikegami, G. (ed.) Dynamical Systems and Nonlinear Oscillators, pp. 210–222. World Scientific Press (1986)
[14] Arbib, M.A.: Monogenic normal systems are universal. Journal of the Australian Mathematical 3(3), 301–306 (1969)
[15] Baetens, J.M., De Baets, B.: Phenomenological study of irregular cellular automata based on Lyapunov exponents and Jacobians. Chaos 20 (2010)
[16] Banks, E. R.: Information and transmission in cellular automata. PhD Dissertionm, Cambridge, MA, MIT (1971)
[17] Berlekamp, E.R., Conway, J.H., Guy, R.K.: Winning Ways for your Mathematical Plays, ch. 25, vol. 2. Academic Press (1982)
[18] Clapham, N.T., Barlow, M., McKay, R.I.: Toward classifying randomly asynchronous cellular automata. In: Proceedings of the 6th International Conference on Complex Systems (CS 2002), pp. 63–71 (2002)
[19] Chua, L.: A Nonlinear Dynamics Perspective of Wolfram's New Kind of Science, vol. 1, 2, 3, 4, 5. World Scientific Publishing Company (2006-2012)
[20] Codd, E.F.: Cellular Automata. Academic Press, Inc., New York (1968)
[21] Cook, M.: Introduction to the activity of Rule 110 (copyright 1994-1998 Matthew Cook) (1998), `http://w3.datanet.hu/~cook/Workshop/CellAut/Elementary/Rule110/`
[22] Cook, M.: Universality in Elementary Cellular Automata. Complex Systems 15(1), 1–40 (2004)
[23] Cook, M.: A Concrete View of Rule 110 Computation. In: Neary, T., Woods, D., Seda, A.K., Murphy, N. (eds.) The Complexity of Simple Programs, pp. 31–55. Elsevier (2011)
[24] Chaté, H., Manneville, P.: Collective behaviours in Spatially Extended Systems with Local Interactions and Synchronous Updating. Progress in Theoretical Physics 87, 1–60 (1992)
[25] Chaté, H., Ginelli, F., Grégoire, G., Peruani, F., Raynaud, F.: Modeling collective motion: variations on the Vicsek model. The European Physical Journal B 64(3-4), 451–456 (2008)
[26] Culik II, K., Yu, S.: Undecidability of CA Classification Schemes. Complex Systems 2(2), 177–190 (1988)
[27] Davis, M. (ed.): Solvability, Provability, Definability: The Collected Works of Emil L. Post. Birkhäuser, Boston (1994)

[28] Das, R., Mitchell, M., Crutchfield, J.P.: A genetic algorithm discovers particle-based computation in cellular automata. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, pp. 344–353. Springer, Heidelberg (1994)
[29] Durr, C., Rapaport, I., Theyssier, G.: Cellular automata and communication complexity. Theoretical Computer Science 322, 355–368 (2004)
[30] Eppstein, D.: Wolfram's Classification of Cellular Automata (1999), http://www.ics.uci.edu/~eppstein/ca/wolfram.html
[31] Eppstein, D.: Searching for spaceships. MSRI Publications 42, 433–452 (2002)
[32] Fatès, N., Morvan, M.: An Experimental Study of Robustness to Asynchronism for Elementary Cellular Automata. Complex Systems 16, 1–27 (2005)
[33] Fredkin, E., Toffoli, T.: Design Principles for Achieving High-Performance Submicron Digital Technologies. In: Adamatzky, A. (ed.) Game of Life Cellular Automata, pp. 27–46. Springer (2001)
[34] Gardner, M.: Mathematical Games — The fantastic combinations of John H. Conway's new solitaire game Life. Scientific American 223, 120–123 (1970)
[35] Griffeath, D., Moore, C.: Life Without Death is P-complete. Complex Systems 10(6), 437–447 (1996)
[36] Griffeath, D., Moore, C.: New constructions in cellular automata. Oxford University Press (2003)
[37] Grünbaum, B., Shephard, G.C.: Tilings and Patterns. W. H. Freeman and Company, New York (1987)
[38] Goles, E., Moreira, A., Rapaport, I.: Communication complexity in number-conserving and monotone cellular automata. Theoretical Computer Science 412, 3616–3628 (2011)
[39] Gunji, Y.-P.: Inducing Class 4 behaviour on the Basis of Lattice Analysis. Complex Systems 19(5), 177–194 (2010)
[40] Gutowitz, H.A., Victor, J.D., Knight, B.W.: Local structure theory for cellular automata. Physica D 28, 18–48 (1987)
[41] Gutowitz, H.A., Victor, J.D.: Local structure theory in more that one dimension. Complex Systems 1(1), 57–68 (1987)
[42] Gutowitz, H.A.: Mean Field vs. Wolfram Classification of Cellular Automata (1989), Historical link http://www.santafe.edu/~hag/mfw/mfw.html, Functional link http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.4525
[43] Gutowitz, H.A., Victor, J.D.: Local structure theory: calculation on hexagonal arrays, and iteraction of rule and interaction of rule and lattice. Journal of Statistical Physical 54, 495–514 (1999)
[44] Hey, A.J.G.: Feynman and computation: exploring the limits of computers. Perseus Books (1998)
[45] Hutton, T.J.: Codd's self-replicating computer. Artificial Life 16(2), 99–117 (2010)
[46] Imai, K., Morita, K.: A computation-universal two-dimensional 8-state triangular reversible cellular automation. Theoret. Comput. Sci. 231, 181–191 (2000)
[47] Jakubowski, M.H., Steiglitz, K., Squier, R.: Computing with Solitons: A Review and Prospectus. Multiple-Valued Logic 6(5-6) (2001) (also republished in [4])
[48] Kari, J.: Representation of Reversible Cellular Automata with Block Permutations. Mathematical Systems Theory 29(1), 47–61 (1996)
[49] Kudlek, M., Rogozhin, Y.: Small Universal Circular Post Machine. Computer Science Journal of Moldova 9(25), 34–52 (2001)
[50] Kauffman, S.: The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press (1993)
[51] Langton, C.G.: Self Reproduction in Cellular Automata. Physica D 10, 135–144 (1984)

[52] Langton, C.G.: Studying Artificial Life with Cellular Automata. Physica D 22(1-3), 120–149 (1986)
[53] Li, W., Nordahl, M.G.: Transient behaviour of cellular automaton Rule 110. Physics Letters A 166, 335–339 (1992)
[54] Li, W., Packard, N.: The Structure of the Elementary Cellular Automata Rule Space. Complex Systems 4(3), 281–297 (1990)
[55] Redeker, M., Adamatzky, A., Martínez, G.J.: Expressiveness of Elementary Cellular Automata (submitted)
[56] Martínez, G.J., Adamatzky, A., Alonso-Sanz, A., Seck-Tuoh-Mora, J.C.: Complex dynamic emerging in Rule 30 with majority memory. Complex Systems 18(3), 345–365 (2010)
[57] Martínez, G.J., Adamatzky, A., Alonso-Sanz, A.: Complex dynamics of cellular automata emerging in chaotic rules. Int. J. Bifurcation and Chaos 22(2) (2012)
[58] Margenstern, M.: Cellular Automata in Hyperbolic Spaces. Old City Publishing, Inc. (2007)
[59] Margolus, N.H.: Physics-like models of computation. Physica D 10(1-2), 81–95 (1984)
[60] Margolus, N.H.: Crystalline Computation. In: Hey, A.J.G. (ed.) Feynman and Computation: Exploring the Limits of Computers, pp. 267–305. Perseus Books (1998)
[61] Margolus, N.H.: Universal Cellular Automata Based on the Collisions of Soft Spheres. In: Griffeath, D., Moore, C. (eds.) New Constructions in Cellular Automata, pp. 231–260. Oxford University Press (2003)
[62] Martínez, G.J., Adamatzky, A., Seck-Tuoh-Mora, J.C., Alonso-Sanz, A.: How to make dull cellular automata complex by adding memory: Rule 126 case study. Complexity 15(6), 34–49 (2010)
[63] Martínez, G.J., Morita, K., Adamatzky, A., Margenstern, M.: Majority Adder Implementation by Competing Patterns in Life-Like Rule $B2/S2345$. In: Calude, C.S., Hagiya, M., Morita, K., Rozenberg, G., Timmis, J. (eds.) Unconventional Computation. LNCS, vol. 6079, pp. 93–104. Springer, Heidelberg (2010)
[64] Martínez, G.J., McIntosh, H.V., Seck-Tuoh-Mora, J.C.: Gliders in Rule 110. Int. J. of Unconventional Computing 2(1), 1–49 (2006)
[65] Martínez, G.J., McIntosh, H.V., Seck-Tuoh-Mora, J.C., Chapa-Vergara, S.V.: Reproducing the cyclic tag system developed by Matthew Cook with Rule 110 using the phases f_1_1. Journal of Cellular Automata 6(2-3), 121–161 (2011)
[66] Martínez, G.J., Adamatzky, A., Stephens, C.R., Hoeflich, A.F.: Cellular automaton supercolliders. International Journal of Modern Physics C 22(4), 419–439 (2011)
[67] McIntosh, H.V.: Wolfram's Class IV and a Good Life. Physica D 45, 105–121 (1990)
[68] McIntosh, H.V.: Linear Cellular Automata via de Bruijn Diagrams (1991), http://delta.cs.cinvestav.mx/~mcintosh/oldweb/cf/debruijn.html
[69] McIntosh, H.V.: Reversible Cellular Automata (1991), http://delta.cs.cinvestav.mx/~mcintosh/oldweb/ra/ra.html
[70] McIntosh, H.V.: Rule 110 as it relates to the presence of gliders (1999), http://delta.cs.cinvestav.mx/~mcintosh/comun/RULE110W/RULE110.html
[71] McIntosh, H.V.: A Concordance for Rule 110 (2000), http://delta.cs.cinvestav.mx/~mcintosh/comun/ccord/ccord.html
[72] McIntosh, H.V.: Rule 110 Is Universal! (2002), http://delta.cs.cinvestav.mx/~mcintosh/comun/texlet/texlet.html

[73] McIntosh, H.V.: One Dimensional Cellular Automata. Luniver Press (2009)
[74] Morita, K., Harao, M.: Computation universality of one-dimensional reversible (injective) cellular automata. Trans. IEICE Japan E-72, 758–762 (1989)
[75] Mitchell, M., Hraber, P.T., Crutchfield, J.P.: Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations. Complex Systems 7(2), 89–130 (1993)
[76] Mills, J.W.: The Nature of the Extended Analog Computer. Physica D 237(9), 1235–1256 (2008)
[77] Morita, K., Margenstern, M., Imai, K.: Universality of reversible hexagonal cellular automata. Theoret. Informatics Appl. 33, 535–550 (1999)
[78] Morita, K.: A simple construction method of a reversible finite automaton out of Fredkin gates, and its related problem. Trans. IEICE Japan E-73, 978–984 (1990)
[79] Morita, K.: Simple universal one-dimensional reversible cellular automata. Journal of Cellular Automata 2, 159–166 (2007)
[80] Morita, K.: Reversible computing and cellular automata—A survey. Theoretical Computer Science 395, 101–131 (2008)
[81] Morita, K., Tojima, Y., Imai, K., Ogiro, T.: Universal Computing in Reversible and Number-Conserving Two-Dimensional Cellular Spaces. In: Adamatzky, A. (ed.) Collision-Based Computing, pp. 161–199. Springer (2002)
[82] Margolus, N., Toffoli, T., Vichniac, G.: Cellular-Automata Supercomputers for Fluid Dynamics Modeling. Physical Review Letters 56(16), 1694–1696 (1986)
[83] Nasu, M.: Local Maps Inducing Surjective Global Maps of One-Dimensional Tesselation Automata. Mathematical Systems Theory 11, 327–351 (1978)
[84] Park, J.K., Steiglitz, K., Thurston, W.P.: Soliton-like behaviour in automata. Physica D 19, 423–432 (1986)
[85] Rendell, P.: Turing universality of the game of life. In: Adamatzky, A. (ed.) Collision-Based Computing, pp. 513–540. Springer (2002)
[86] Rendell, P.: A Universal Turing Machine in Conway's Game of Life. In: Proceedings of the 2011 International Conference on High Performance Computing & Simulation, pp. 764–772. IEEE Xplore (2011) 10.1109/HPCSim.2011.5999906
[87] Sapin, E., Bailleux, O., Chabrier, J.-J., Collet, P.: A New Universal Cellular Automaton Discovered by Evolutionary Algorithms. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 175–187. Springer, Heidelberg (2004)
[88] Sapin, E., Bailleux, O., Chabrier, J.-J., Collet, P.: Demonstration of the Universality of a New Universal Cellular Automaton. Int. J. Unconventional Computing 3(2), 79–103 (2007)
[89] Seck-Tuoh-Mora, J.C., Chapa-Vergara, S.V., Martínez, G.J., McIntosh, H.V.: Procedures for calculating reversible one-dimensional cellular automata. Physica D 202, 134–141 (2005)
[90] Seck-Tuoh-Mora, J.C., Hernández, M.G., McIntosh, H.V., Chapa-Vergara, S.V.: The Inverse behaviour of a Reversible One-Dimensional Cellular Automaton Obtained by a Single Welch Diagram. In: Adamatzky, A., Sanz, R.A., Lawniczak, A., Martínez, G.J., Morita, K., Worsh, T. (eds.) Automata 2008: Theory and Applications of Cellular Automata, pp. 114–125. Luniver Press (2008)
[91] Seck-Tuoh-Mora, J.C., Martínez, G.J., McIntosh, H.V.: The Inverse behaviour of a Reversible One-Dimensional Cellular Automaton Obtained by a Single Welch Diagram. Journal of Cellular Automata 1(1), 25–39 (2006)
[92] Sipper, M.: Evolution of Parallel Cellular Machines. LNCS, vol. 1194. Springer, Heidelberg (1997)
[93] Siwak, P.: Iterons of Automata. In: Adamatzky, A. (ed.) Collision-Based Computing, pp. 299–354. Springer (2002)

[94] Sutner, K.: A note on Culik-Yu classes. Complex Systems 3(1), 107–115 (1989)
[95] Sutner, K.: De Bruijn graphs and linear cellular automata. Complex Systems 5(1), 19–30 (1991)
[96] Sutner, K.: Classification of Cellular Automata. In: Meyers, R.A. (ed.) Encyclopedia of Complexity and Systems Science, Part 3, pp. 755–768 (2009)
[97] Suzudo, T.: Spatial pattern formation in asynchronous cellular automata with mass conservation. Physica A 343, 185–200 (1994)
[98] Toffoli, T.: Non-Conventional Computers. In: Webster, J. (ed.) Encyclopedia of Electrical and Electronics Engineering, pp. 455–471. Wiley & Sons (1998)
[99] Tommaso, T., Norman, M.: Cellular Automata Machines. The MIT Press, Cambridge (1987)
[100] Tomassini, M., Venzi, M.: Evolving Robust Asynchronous Cellular Automata for the Density Task. Complex Systems 13(1), 185–204 (2002)
[101] Israeli, N., Goldenfeld, N.: Computational Irreducibility and the Predictability of Complex Physical Systems. Phys. Rev. Lett. 92, 74105–74108 (2004)
[102] Voorhees, B.H.: Computational analysis of one-dimensional cellular automata. World Scientific Series on Nonlinear Science, Series A, vol. 15 (1996)
[103] Voorhees, B.H.: Remarks on Applications of De Bruijn Diagrams and Their Fragments. Journal of Cellular Automata 3(3), 187–204 (2008)
[104] von Neumann, J.: Theory of Self-reproducing Automata. University of Illinois Press, Urbana and London (1966) (edited and completed by A. W. Burks)
[105] Wuensche, A., Lesser, M.: The Global Dynamics of Cellular Automata. Addison-Wesley Publishing Company (1992)
[106] Worsch, T., Nishio, H.: Achieving Universality of CA by Changing the neighbourhood. Journal of Cellular Automata 4(3), 237–246 (2009)
[107] Wolfram, S.: Statistical Mechanics of Cellular automata. Reviews of Modern Physics 55(3), 601–644 (1983)
[108] Wolfram, S.: Universality and complexity in cellular automata. Physica D 10, 1–35 (1984)
[109] Wolfram, S.: Computation Theory of Cellular Automata. Communications in Mathematical Physics 96, 15–57 (1984)
[110] Wolfram, S.: Random Sequence Generation by Cellular Automata. Advances in Applied Mathematics 7, 123–169 (1986)
[111] Wolfram, S.: Cellular Automata Supercomputing. In: Wilhelmson, R.B. (ed.) High Speed Computing: Scientific Applications and Algorithm Design, pp. 40–48. University of Illinois Press (1988)
[112] Wolfram, S.: Cellular Automata and Complexity. Addison-Wesley Publishing Company (1994)
[113] Wolfram, S.: Twenty Problems in the Theory of Cellular Automata. Physica Scripta T9, 170–183 (1985)
[114] Wolfram, S.: A New Kind of Science. Wolfram Media, Inc., Champaign (2002)
[115] Worsch, T.: Cellular Automata as Models of Parallel Computation. In: Meyers, R.A. (ed.) Encyclopedia of Complexity and Systems Science, pp. 741–755. Springer (2009)
[116] Wolz, D., de Oliveira, P.P.B.: Very effective evolutionary techniques for searching cellular automata rule spaces. Journal of Cellular Automata 3(4), 289–312 (2008)
[117] Wuensche, A.: Genomic Regulation Modeled as a Network with Basins of Attraction. In: Altman, R.B., Dunker, A.K., Hunter, L., Klien, T.E. (eds.) Pacific Symposium on Biocomputing 1998, pp. 89–102. World Scientific, Singapore (1998)
[118] Wuensche, A.: Classifying Cellular Automata Automatically. Complexity 4(3), 47–66 (1999)

[119] Zenil, H.: Compression-based Investigation of the Dynamical Properties of Cellular Automata and Other Systems. Complex Systems 19(1), 1–28 (2010)
[120] Zenil, H.: Nature-like Computation and a Measure of Computability. In: Dodig-Crnkovic, G., Giovagnoli, R. (eds.) Natural Computing/Unconventional Computing and its Philosophical Significance. SAPERE Series. Springer (2012)
[121] Zenil, H.: On the Dynamic Qualitative behaviour of Universal Computation. Complex Systems 20(3), 265–278 (2012)
[122] Zenil, H.: Programmability Tests for Natural Computation with the Game of Life as a Case Study. Journal of Experimental and Theoretical Artificial Intelligence (accepted) (forthcoming)

Part VI

Irreducibility and Computational Equivalence

Chapter 18
Exploring Wolfram's Notion of Computational Irreducibility with a Two-Dimensional Cellular Automaton

Drew Reisinger, Taylor Martin, Mason Blankenship, Christopher Harrison, Jesse Squires, and Anthony Beavers

The Digital Humanities Laboratory, The University of Evansville, USA

Abstract. The notion of computational irreducibility says that a problem is computationally irreducible when the only way to solve it is to traverse a trajectory through a state space step by step using no shortcut. In this paper, we will explore this notion by addressing whether computational irreducibility is a consequence of how a particular problem is represented. To do so, we will examine two versions of a given game that are isomorphic representations of both the play space and the transition rules underlying the game. We will then develop a third isomorph of the play space with transition rules that seem to only be determined in a computationally irreducible manner. As a consequence, it would seem that representing the play space differently in the third isomorph introduces computational irreducibility into the game where it was previously lacking. If so, we will have shown that, in some cases at least, computational irreducibility depends on the representation of a given problem.

Keywords: Cellular automaton, computational equivalence, computational irreducibility, formal equivalence, Haugeland, Wolfram.

1 The Problem

In *A New Kind of Science* [11], Stephen Wolfram observes that "when viewed in computational terms most of the great historical triumphs of theoretical science turn out to be remarkably similar in their basic character. For at some level almost all of them are based on finding ways to reduce the computational work that has to be done in order to predict how some particular system will behave" (p. 737 [11]). Systems for which this is not possible are termed "computationally irreducible" (CI). Furthermore, "whenever computational irreducibility exists in a system, it means that in effect there can be no way to predict how the system will behave, except by going through almost as many steps of computation as the evolution of the system itself" (p. 739 [11]). While the notion exceeds the scope of cellular automata, much of Wolfram's research in this area involves them. Indeed, this is one place where the notion of CI is transparent. For instance, in

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 263–272.
DOI: 10.1007/978-3-642-35482-3_18

John Conway's *Game of Life*, a two-dimensional cellular automaton, artificial agents, or "counters," operate according to four rules:

1. Every counter with two or three neighboring counters survives for the next generation.
2. Each counter with four or more neighbors dies (is removed) from overpopulation.
3. Every counter with one neighbor or no neighbors dies from isolation.
4. Each empty cell adjacent to exactly three neighbors–no more, no fewer–is a birth cell. A counter is placed on it at the next move. (p. 120 [1])

At each time stamp, or tick, all of the agents in the system follow these rules. The question that we can ask concerning CI is whether it is possible to predict in advance what the state of the system will be in, say, fifty ticks, given a particular initial configuration without having to step through each tick and let the system evolve. If we cannot find a "shortcut," as it were, the system is computationally irreducible. In *Artificial Intelligence: The Very Idea* [2], John Haugeland presents a pair of isomorphs to demonstrate the notion of formal equivalence, which should not be confused with Wolfram's notion of computational equivalence. The latter says that "almost all processes that are not obviously simple can be viewed as computations of equivalent sophistication" (pp. 716-717 [11]). Formal equivalence, on the other hand, occurs when the following conditions hold:

1. For each distinct position in one system there is exactly one corresponding position in the other system.
2. Whenever a move would be legal in one system, the corresponding move (i.e., from the corresponding position to the corresponding position) would be legal in the other system.
3. All of the starting positions correspond. (p. 62 [2]).

To demonstrate the notion of formal equivalence, Haugeland invites his reader to consider a couple of formal games. The first one is defined thusly:

1. The tokens are thirty-three plastic chips, each with two letters written on it: one from early in the alphabet (A-G), and one from late in the alphabet (T-Z).
2. In the starting position, all of the chips are together in a white basket, except for the one marked *DW*, which is all by itself in a black basket.
3. A move consists of exchanging two chips in the white basket for one in the black; but there's a restriction on which chips can be exchanged for which:
   a. *either* all three chips must have the same early letter and sequential late letters, *or else* they must have the same late letter and sequential early letters;
   b. *and* the middle letter in the sequence can't be on the one chip going from the black basket to the white.

For example:
AX, BX <=> CX and AX, AW <=> AV are legal moves;
but ET, EV <=> EZ isn't legal (second letters not sequential);
AW, BX <=> CY isn't legal (no same letter); and
AX, CX <=> BX isn't legal (middle letter in black basket). (pp. 60-61 [2])

Haugeland then invites his reader to consider a different game with tokens that are all identical and that are laid out in a particular spatial configuration as follows:

Table 1. A spatial representation of Haugeland's game

		X	X	X		
		X	X	X		
X	X	X	X	X	X	X
X	X	X		X	X	X
X	X	X	X	X	X	X
		X	X	X		
		X	X	X		

In this game, the X's represent marbles on a constrained play space. A move consists of jumping one marble over another and then moving the jumped marble off the board. The goal of the game is to end with just one marble left in the center of the board.

The import of Haugeland's demonstration becomes clear when we assign identifiers to the game tokens in the manner shown in Table 2 ([2]). Here, game pieces on the spatialized board represent the tokens in the first game as presented above. By turning the original game into a spatialized version of the same, we have reduced the number of rules, thereby simplifying the game. (For a similar strategy involving the "Towers of Hanoi" game, see p. 84 [4]). Nevertheless, for each token in one game there is a token that plays the same computational role in the other, *and* for each move in one game there is a computationally-equivalent move in the other game. In other words, the games are isomorphs of each other regarding both play state and transition rules for moving from one state to another.

In the investigation that follows, we will construct a third representation of the game state as a compact collection of six "vectors" (to be defined subsequently). To do so, we will treat the spatialized version of Haugeland's game as a cellular automaton in order to address the question of whether computational irreducibility depends on the representation of a given problem.

Israeli and Goldenfeld ([3]) investigated a similar question concerning whether the level of description of a problem can change its computational reducibility. Unlike our work, however, they did so by employing homomorphs that do not preserve all of the information contained in the original system. Our investigation differs insofar as we developed an alternate representation that is an actual

Table 2. Reproduced from Haugeland p. 61 [2]

		AV	AW	AX		
		BV	BW	BX		
CT	CU	CV	CW	CX	CY	CZ
DT	DU	DV		DX	DY	DZ
ET	EU	EV	EW	EX	EY	EZ
		FV	FW	FX		
		GV	GW	GX		

isomorphic redescription of the original game. This is important because it responds to a claim that Wolfram makes in [11] concerning counterexamples to computationally irreducible systems:

> If one views the pattern of behavior [in a system] as a piece of data ... regularities in it allow a compressed description to be found. But the existence of a compressed description does not on its own imply computational reducibility. For any system that has simple rules and simple initial conditions – including for example rule 30 – will always have such a description. (p. 746 [11])

In other words, Wolfram claims that changing the representation of a system does not change whether it is computationally reducible or irreducible; however, we think that it can. To show this, we will adopt an inverted strategy by taking a computationally reducible system and transforming it into a computationally irreducible one by changing the representation without a loss of information, that is, again, by creating an isomorphic and not merely a homomorphic representation of the original system. In so doing, we will introduce irreducibility into a formally equivalent system where it was lacking previously.

2 Representing the Board as a Vector Space

The first four vectors are generated based on the number of pieces in each row, column and both diagonal directions of the board. To generate the row vector, we simply counted the number of pieces in each row on the board; we then did the same for the columns and then for both diagonal directions. Thus, the row vector <3,3,7,6,7,3,3> signifies that on a 7 x 7 automaton there are three pieces in the first row, three in the second, seven in the third, and so on. Similarly, if the above were a column vector, it would signify that there are three pieces in the first column, three in the second, seven in the third, and so on. Since there are 13 diagonals in each direction, there are two diagonal vectors each with 13 components. For the board described above, for instance, the diagonal vector in each direction would be the same, <0,0,2,4,5,4,2,4,5,4,2,0,0>. These vectors represent the initial state of Haugeland's game as presented above. Together, we call this four-vector representation the "RCD feature detection method," "RCD" standing for

"row, column, diagonal." By itself, the method cannot represent all unique board states. For instance, the vector set r = <1,2,3,1,4,2,0>, c = <1,1,1,5,2,1,2>, *d1* = <0,0,0,1,3,2,1,2,2,1,1,0,0>, *d2* = <0,0,2,1,2,2,1,2,1,1,1,0,0> (where *d1* represents the diagonals that run from the top left to the bottom right, and *d2* represents that diagonals that run from the top right to the bottom left) represents both of the non-identical boards in Table 3.

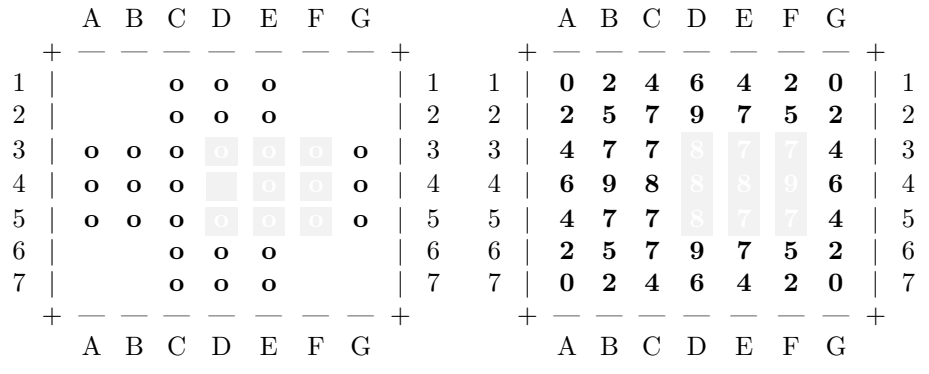
Table 3. Two different board constructions produced from the same set of RCD vectors

	A	B	C	D	E	F	G				A	B	C	D	E	F	G	
1	.	.	.	.	o	.	.	1		1	.	.	.	.	o	.	.	1
2	.	.	o	o	.	.	.	2		2	.	.	.	o	o	.	.	2
3	.	.	.	o	.	o	o	3		3	.	o	.	o	.	.	o	3
4	.	.	.	o	.	.	.	4		4	.	.	.	o	.	.	.	4
5	o	o	.	o	.	.	o	5		5	o	.	.	o	.	o	o	5
6	.	.	.	o	o	.	.	6		6	.	.	o	o	.	.	.	6
7	.	.	.	.	.	.	.	7		7	.	.	.	.	.	.	.	7
	A	B	C	D	E	F	G				A	B	C	D	E	F	G	

To account for this disparity, we added another method that yields what we call "heat map representations." This method involves two vectors, one that defines the board space and the other the pieces on the board. Both of these vectors are necessary because, together, they represent the distribution of pieces *and* the playable spaces for any arbitrary game board. For simplicity's sake, we will start by explaining the piece vector, though both piece and space vectors are determined by counting an individual piece or space plus the number of its neighbors, defining "heat" here as a measure of how clustered together the game elements are. The heat map for the pieces of Haugeland's game above is thus represented by Table 4. To calculate the heat for the piece in the middle of the 3 x 3 highlighted space in the table, for instance, we considered the 3 x 3 space containing it. The heat of the cell was then determined by counting the number of pieces within this space. In this example, the highlighted cell contains a piece that has seven neighboring cells containing pieces; therefore, its heat value is eight. Repeating the same procedure, we then calculated the heat value of each cell, thereby arriving at the heat map displayed in Table 4. Each cell also has a second heat value calculated in a similar manner except that we count the number of empty but playable spaces in the 3 x 3 square instead of the number of pieces.

With these two heat maps defined, we constructed the piece heat vector and space heat vector as follows. The first component of each vector is the number of cells that have a heat value of one. In our example, for instance, the first component of the piece heat vector is zero, since there are no cells with a piece

Table 4. The left image is of the initial board state and the right image is of the corresponding heat map



heat of one. The second component is the number of cells with a heat value of two. Since there are eight of these, the second component is an eight. The third is the number of cells with a heat value of three, and so forth. This sequence continues up to the ninth component, which counts the number of cells with a heat value of nine. In Table 4, the entire piece heat vector is <0,8,0,8,4,4,12,5,4>.

The four RCD vectors together with the two heat vectors describe a third game that is formally equivalent (i.e., isomorphic) to Haugeland's game. This technique represents any board state as a unique set of these six vectors. To show that the reverse correspondence also holds, we developed an algorithm that reconstructs the spatial layout of the board using only the vectors.

Our algorithm uses a stochastic hill climbing search that begins with a random arrangement of pieces on the board, which is, in turn, converted to vectors using the methods described above. These are then compared to a set of vectors representing the goal state of the desired board. Together, both sets of vectors are compared to calculate an error value for each possible future state, and then the state with the lowest error is selected. In the event of a tie, one is selected at random. The process continues until the error value is zero, in which case the vectors serving as the goal successfully reproduce the state of the board.

The procedure for calculating the aforementioned error value is adapted from backpropagation methods used in training artificial neural networks that employ a *mean* square of errors. In our use, the algorithm compares the vectors of the current board with the vectors of the goal by subtracting each component in the current vectors from its counterpart in the goal, squaring this difference, and then *summing* these squares into a single number that represents the error. Since this calculation only uses information contained in the vector representation, the algorithm recreates the spatial board state solely by virtue of the vectors.

To evaluate the performance of the algorithm, we generated a random sample of 450 board configurations. We obtained the vectors from each board in the sample and used the algorithm to reconstruct the boards from these vectors.

For our evaluation run, the algorithm successfully reconstructed all 450 of the generated boards. This result confirms our claim that the vector representation fully encodes the board states in Haugeland's game.

Having shown that we could reconstruct the board using vectors, we then faced the problem of determining equivalent transition rules for Haugeland's game necessary to show that the games are formally equivalent. This is where we run into computational irreducibility. In principle, one could construct these rules by enumerating all of the possible state transitions and then retaining those that entail legal moves in the original game; however, we were not able to construct the game rules explicitly. The difficulty in determining transitions rules for the vector space lies in their elusive, disjunctive, nonlinear nature.

When a piece is moved, the RCD vectors change in a linear manner. If a piece jumps along a column over another piece, it decrements one component of the column vector by one. See the change of the fourth component of the C vector from five to four in Table 5. Three components of the row vector change, but they do so in a predictable manner. Two adjacent components of the row vector decrement by one, while one additional adjacent component is incremented. See the R vector in Table 5 below and the changes in the third through fifth components in the up move. Also note the changes in the fourth through sixth components in the down move. Both diagonal vectors change in a similar manner. Note, however, the unpredictable changes in H1 between the different moves that are highlighted in the table.

Table 5. Changes in vector values given two different available moves in the game

	Initial Board	Jump Up	Jump Down
R	<3,3,5,6,6,2,3>	<3,3,6,5,5,2,3>	<3,3,5,5,5,3,3>
C	<3,2,6,5,6,3,3>	<3,2,6,4,6,3,3>	<3,2,6,4,6,3,3>
D_1	<0,0,2,4,5,1,2,4,4,4,2,0,0>	<0,0,2,4,5,2,1,3,4,4,2,0,0>	<0,0,2,4,5,1,1,3,5,4,2,0,0>
D_2	<0,0,2,3,4,3,3,3,4,4,2,0,0>	<0,0,2,3,4,2,2,4,4,4,2,0,0>	<0,0,2,3,5,2,2,3,4,4,2,0,0>
H_1	<9,15,7,7,8,11,6,2,0>	**<9,15,8,8,10,9,4,2,0>**	**<9,15,5,12,10,9,3,2,0>**
H_2	<15,9,4,0,0,0,0,0,0>	<13,6,4,3,1,0,0,0,0>	<12,6,4,3,3,0,0,0,0>

While the RCD vector transitions can be easily predicted as demonstrated, the heat vector transitions are another matter, as they are based on the clustering of pieces. There does not appear to be a clear and easy set of transition rules as in the previous cases. Two jumps from the same board configuration produce very different heat vectors despite similar structure in the RCD representation. The heat vectors vary when moving based on how tightly clustered the pieces are in the resulting board. This clustering of the pieces is not fully represented by the RCD vectors because they only show the number of pieces in each row, column, and diagonal and do not account for the relative positions of those pieces. Thus the transitions for the heat vectors cannot be directly derived from the RCD transition rules. In other words, while the heat vector transitions do need to be

Table 6. Changes in heat vectors throughout a game

Heat Vector	Modified Vector	Change		
<8,16,4,8,4,4,12,5,4>	-	U	D	N
<8,16,6,6,7,5,8,5,4>	<N,N,U,D,U,U,D,N,N>	3	2	4
<9,15,8,6,7,3,10,3,4>	<U,D,U,N,N,D,U,D,N>	3	3	3
<10,15,7,5,6,4,7,4,5>	<U,N,D,D,D,U,D,U,U>	4	4	1
<9,17,5,8,6,4,7,4,3>	<D,U,D,U,N,N,N,N,D>	2	3	4
<10,15,8,7,8,2,7,4,2>	<U,D,U,D,U,D,N,N,D>	3	4	2
<11,14,7,7,7,2,7,4,2>	<U,D,D,N,D,N,N,N,N>	1	3	5
<13,13,8,6,7,3,5,4,2>	<U,D,U,D,N,U,D,N,N>	3	3	3
<17,12,6,6,6,3,5,4,2>	<U,D,D,N,D,N,N,N,N>	1	3	5
<15,13,7,8,6,4,6,1,1>	<D,U,U,U,N,U,U,D,D>	5	3	1
<16,11,9,8,3,4,4,2,1>	<U,D,U,N,D,N,D,U,N>	3	3	3
<16,13,9,9,3,6,1,1,1>	<N,U,N,U,N,U,D,D,N>	3	2	4
<18,12,7,7,5,4,2,1,1>	<U,D,D,D,U,D,U,N,N>	3	4	2
<17,13,8,9,5,3,1,0,1>	<D,U,U,U,N,D,D,D,N>	3	4	2
<20,11,4,12,2,3,2,0,1>	<U,D,D,U,D,N,U,N,N>	3	3	3
<17,14,6,13,2,2,1,0,0>	<D,U,U,U,N,D,D,N,D>	3	4	2
<14,11,9,9,2,3,0,1,0>	<D,D,U,D,N,U,D,U,N>	3	4	2
<14,13,7,6,3,3,0,1,0>	<N,U,D,D,U,N,N,N,N>	2	2	5
<18,13,5,4,3,2,1,1,0>	<U,N,D,D,N,D,U,N,N>	2	3	4
<18,14,6,5,1,2,1,0,0>	<N,U,U,U,D,N,N,D,N>	3	2	4
<19,11,8,3,2,2,0,0,0>	<U,D,U,D,U,N,D,N,N>	3	3	3
<20,10,7,3,1,2,0,0,0>	<U,D,D,N,D,N,N,N,N>	1	3	5
<20,13,6,3,1,0,0,0,0>	<N,U,D,N,N,D,N,N,N>	1	2	6
<20,12,8,1,0,0,0,0,0>	<N,D,U,D,D,N,N,N,N>	1	3	5
<27,8,4,2,0,0,0,0,0>	<U,D,D,U,N,N,N,N,N>	2	2	5
<25,10,3,0,0,0,0,0>	<D,U,D,D,N,N,N,N,N>	1	3	5
<23,8,2,0,0,0,0,0,0>	<D,D,D,N,N,N,N,N,N>	0	3	6
<26,5,0,0,0,0,0,0,0>	<U,D,D,N,N,N,N,N,N>	1	2	6

dependent in some way on the RCD vectors to ensure that all six vectors in the representation are kept in sync, they do not follow any discernible pattern in their transitions, despite the fact that the RCD vectors on which they need to depend do change linearly. This lack of linearity in the transitions of the heat vectors is demonstrated by the sporadic nature of the heat vector transitions shown in Table 6. Note the lack of any discernible patterns in the changes in heat vectors between moves. See in particular the columns labeled U, D, and N below, which stand for a transformation upward, downward, or no change, respectively.

From the data shown in Table 6, it would seem that we are dealing with a truly nonlinear set of transition rules necessary for computational irreducibility. Not only that, there appears to be no readily discernible pattern for the transition rules, though further mathematical analysis is required for definitive proof.

3 Conclusions and Caveats

To demonstrate formal equivalence, Haugeland introduced two isomorphic representations of the same underlying game: one in which plastic chips are exchanged between two baskets according to a purely symbolic set of rules, the other in which tokens are laid out in a spatial configuration such that a move consists of jumping one token with another and removing the jumped token. The spatial representation yielded simpler rules while preserving both the play space and the transition rules of the underlying game. In both cases, the transition rules for the games appear to be computationally reducible.

However, when we constructed a third isomorphic representation of the game's state space with our compact collection of six vectors, we lost the ability to formulate transition rules easily. Indeed it appears that the only method for determining the transition rules from one state to another involves stepping through the state space, and thus the process appears to be computationally irreducible. If so, then we have shown that computational irreducibility is contingent upon the representation of a given problem. Nevertheless, Zenil, Soler-Toscano, and Joosten [6] have developed a framework for empirically searching for shortcuts to computations. Applying their framework to our representation may reveal a counterexample to our claim. On the other hand, their work with predicting the outputs of Turing machines presents examples of functions that are easily predictable in their binary encoding but become more complex in decimal. Both their results and our vector representation exhibit an increase in complexity caused by a change in representation and, therefore, come to bear on the larger question addressed in this paper.

Our proof here is empirical, not mathematical. But it nonetheless provides an existence proof of a relationship between the representation of a problem and the question of reducibility. To arrive at a more definitive claim about Wolfram's work, it would be necessary to take one of Wolfram's computationally irreducible cellular automata, formulate an isomorphic representation of it, and then determine whether transition rules of the equivalent system are computationally reducible. This effort would, in effect, reverse the procedure laid out in this paper, but we leave it for the next stages of this project.

Acknowledgments. We would like to thank Cody Baker, Jake Brake, Matthew Hamilton, Kasey Michel, Samantha Miller, Aaron Ricketts, Justin Simerly, and Karolina Tóth for their helpful contributions in the development of this paper.

References

[1] Gardner, M.: Mathematical games: The fantastic combinations of John Conway's new solitaire game "life". Scientific American 223, 120–123 (1970)
[2] Haugeland, J.: Artificial intelligence: The very idea. MIT Press (1985)

[3] Israeli, N., Goldenfeld, N.: Computational irreducibility and the predictability of complex physical systems. Physics Review Letters 92, 074105 (2004)
[4] Norman, D.: Things that make us smart: Defending human attributes in the age of the machine. Basic Books (1994)
[5] Wolfram, S.: A new kind of science. Wolfram Media (2002)
[6] Zenil, H., Soler-Toscano, F., Joosten, J.: Empirical encounters with computational irreducibility and unpredictability. Minds and Machines Online First (retrieved April 25, 2012)

Chapter 19
Unpredictability and Computational Irreducibility

Hervé Zwirn[1] and Jean-Paul Delahaye[2]

[1] UFR de Physique (Université Paris 7), and
CMLA (ENS Cachan) & IHPST (CNRS), France
herve.zwirn@m4x.org
[2] Laboratoire d'Informatique Fondamentale de Lille (CNRS), France
jean-paul.delahaye@lifl.fr

Abstract. We explore several concepts for analyzing the intuitive notion of computational irreducibility and we propose a robust formal definition, first in the field of cellular automata and then in the general field of any computable function f from N to N. We prove that, through a robust definition of what means "to be unable to compute the n^{th} step without having to follow the same path than simulating the automaton or to be unable to compute $f(n)$ without having to compute $f(i)$ for $i = 1$ to $n-1$", this implies genuinely, as intuitively expected, that if the behavior of an object is computationally irreducible, no computation of its n^{th} state can be faster than the simulation itself.

Keywords: Complexity, logical depth, cellular automata, irreducibility, computation.

1 Introduction

It is now common knowledge that it is not because the behavior of a system is deterministic that it is possible to predict it. That has been proven in mathematics in the first part of the 20^{th} century through the work of Kurt Gödel on the formal axiom systems and of Alan Turing [7, 8, 6, 39] on the computing machines. That has also been proven in physics after Henri Poincaré, Edward Lorenz and the subsequent works on the so called "deterministic chaos" during the second half of the XX^{th} century [23, 35, 39, 29, 30, 5]. So, we now know that even if a system is deterministic it can be the case that we can't predict its behavior in the long run. The reasons for this unpredictability can be multiple. If the behavior of a dynamical non linear system is not predictable, it is because very near initial conditions can lead to very far states after a certain time. So, predicting the behavior in the long term would mean knowing initial conditions with an infinite precision which is impossible. As far as computing machines are concerned, the halting problem is known to be undecidable. That means that no algorithm, if fed with the description of a particular Turing machine, can tell if this machine is going to stop or to

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 273–295.
DOI: 10.1007/978-3-642-35482-3_19

compute forever. Incidentally, a consequence of this undecidability is that for every formal system, there exists a Turing machine that doesn't halt but for which it's impossible to prove that it will never stop. In this case, what we can't predict is a fact about a fully deterministic machine.

Another kind of unpredictability is of interest. It concerns also computing machines but is linked with their computation time. Predicting the behavior of a computation machine is to be able to find the result it computes faster than the machine itself. Of course in any case, using a modern computer obviously enables to get faster the result of a given program running on an older one. A more interesting definition is: given a Turing machine computing a definite function, is there another Turing machine [16, 19, 18, 26, 27, 34, 17] computing the same function faster (i.e. in a smaller number of steps)? It is useful to be more precise and to restrict the type of the Turing machines allowed to compete. The efficiency of different kinds of Turing machines is not the same. For example, when deciding whether a string of length l is a palindrome, a 1-tape Turing machine will need a number of steps of $\mathrm{O}(l^2)$ whereas a 2-tape Turing machine will use only $\mathrm{O}(l)$ [27, 17]. Therefore an immediate question occurs: Is it possible by adding an arbitrary number of tapes to increase indefinitely the gap between the speed of 1-tape Turing machines and the speed of k-tape Turing machines? The answer is no for a well known theorem states that "given any k-tape Turing machine M operating within time $f(n)$, we can construct a 1-tape Turing machine M' operating within time $\mathrm{O}(f(n)^2)$ and such that for any input x, $M'(x) = M(x)$ [27]". That means that whatever fast k-tape machine is considered for computing a function, there is a 1-tape Turing machine doing the same job in just the square of the time needed by the k-tape machine. So the best we can do is achieving quadratic savings in time. The link with our subject of unpredictability is clear: if a computation is performed through the fastest algorithm that can perform it, it will not be possible to predict the result. Of course, these considerations are of interest only when infinite processes are considered (i.e. computation of a function $f(n)$ for all values of n). For a finite computation of a given data d the fastest algorithm that computes d is always something like: print "d". A more precise definition will be given below and more will be said as well about the so called "speed-up" theorems.

The question of unpredictability can be put forward in a more direct way: given a physical system whose behavior can be calculated by simulating explicitly each step in its evolution, is it always possible to predict the outcome without tracing each step? That means: is there always a shortcut to go directly to the n^{th} step? A computer is of course such a physical system. Wolfram conjectured [33, 20, 37] that in most cases the answer is no. This "no" expresses the concept of computational irreducibility (CIR). This question has been widely analyzed in the context of cellular automata (CA) by Wolfram [34]. A cellular automaton (CA) is computationally irreducible if in order to know the state of the system after n steps there is no other way than to evolve the system n times according to the equations of motion. That means that there is no short-cut, no way

to compress the dynamic. Thus the system appears to be unpredictable. The intuition behind this definition is that there is no other way to reach the n^{th} state than to go through the (n-1) previous ones. The consequence is that it's impossible to reach the n^{th} state faster than the automaton itself. While really appealing from the intuitive standpoint, this definition lacks for robustness. If we take it at face value, asking that there is no other possibility than to go through exactly the same states than the automaton itself, it's too restrictive. Imagine for example, that there is an algorithm A_1 such that it gives for each input n a result differing only by a small difference from the n^{th} state of the automaton and that for computing the result for n, it goes through all the previous results for $i < n$. Imagine as well that there is an algorithm A_2 that computes the n^{th} state of the automaton from the n^{th} result of the algorithm A_1. Then, the composed algorithm $A_2 \circ A_1$ (where the symbol o stands for the composition) will give the n^{th} state of the automaton when given n as input but will not follow the same path than the automaton itself. Nevertheless, if no other algorithm is able to give directly the n^{th} state of the automaton, we will not be willing to consider the existence of $A_2 \circ A_1$ as a counter example of the fact that the automaton is CIR. That means that demanding that the only way to reach the n^{th} state is to follow the exact path of the automaton through the $(n-1)$ previous states is too restrictive. So a first question is: how far can we depart from this path? A second question is related to the computation time. Could it be enough to say that a CA is CIR if it's impossible to know its n^{th} state faster than itself even if it is possible to compute this n^{th} state without having to go through all the previous states? Even if these questions have been raised inside the cellular automata field, they concern as well the field of all computable functions. What does that mean for a function $f(n)$ from N to N (the set of all positive or null integers) to be CIR? Is there any function $f(n)$ such that it's necessary to compute all the $f(i)$ for $i = 1$ to $n - 1$, to get $f(n)$?

In this paper, we deal with these questions and attempt to find a robust formal definition of the concept of computational irreducibility. Our goal is general and we look for a concept of computational irreducibility that is applicable to any system. Computational irreducibility is related to a system and a computation model. For the sake of simplicity, we'll use as our typical examples of systems the cellular automata and we'll consider Turing machines as implementing the concept of computation. The reader will easily convince himself that these choices don't imply any loss of generality and at the end of the paper, we'll aim at giving the most general form to our results so as not to be restricted to the field of cellular automata which is used only as a suitable example.

2 Turing Machines

We assume the reader to be familiar with the concept of Turing machines [16, 19, 26, 27, 34, 17] but we give nevertheless the basic notions.

A Turing machine is a theoretical device capable of manipulating a list of cells called tape (infinite in principle), using an access pointer called the head through a finite program. Each cell can contain a symbol from a finite alphabet. The time is discrete and each instruction is executed in one step of time. The head is always positioned over a particular cell which it is said to scan. The machine can be in internal states (members of a finite set always containing at least a start state and a halting state). At time 0, the head is supposed to scan the leftmost cell (the string written on the tape at the beginning is considered as input, possibly empty) and the machine to be in the start state. At each step, the head reads the symbol written on the scanned cell, erases it or writes another symbol, goes left or right and changes its internal state. The program of the machine is a finite rule (sometimes called the transition function) which states exactly what the head must do depending on the symbol written on the scanned cell and the internal state of the machine. When the machine reaches the halting state the computation is finished and what is written on the tape is the result of the computation. Of course, the machine can possibly never stop. It is also possible to consider Turing machines with several tapes. As we'll see, the consideration of multi-tape Turing machines is important when dealing with the concept of complexity. Multi-tape Turing machines are also very useful to facilitate the design of machines that compute functions of interest.

The main point with the Turing machines model is that it is very simple and that through the Church-Turing thesis [15, 11], it allows the computation of any computable function. More precisely, the Church-Turing thesis says that a function can be computed by some Turing machine if and only if it can be computed by some machine of any other reasonable and general model of computation. Put simply, that means that a function is in any way computable if and only if it can be computed with a Turing machine.

The fact that the description of any Turing machine through its transition function is itself computable shows that there exist Turing machines able to simulate any other Turing machine. Such a machine is called a Universal Turing machine. A Universal Turing machine U is such that given the index i of any Turing machine T under a given numbering, it simulates the computation of T on every input argument m:

For all i, if T is the i^{th} Turing machine, then for all m, $U(i, m) = T(m)$.

We'll see that Universal Turing machines are fundamental tools for defining many important notions in the following.

3 Some General Speed-Up Results

Notations. We'll need in the following some notations for comparing the order of magnitude of different functions. We recall here the standard notations.

1. $f(n) = O(g(n))$ if there are constants $c > 0$, $n_0 > 0$ such that $\forall n > n_0$, $|f(n)| \leq c|g(n)|$.

2. $f(n) = o(g(n))$ if $lim_{n\to\infty}\ f(n)/g(n) = 0$.
3. $f = \omega(g)$ if there is a constant $n_0 > 0$ such that $\forall c > 0$, $|f(n)| > c|g(n)|$.
4. $f(n) = \Omega(g(n))$ if there is a constant $c > 0$ such that $|f(n)| \geq c|g(n)|$ infinitely often.

It is well known that the time complexity of a problem may depend on the model of computation. We mentioned above the problem of deciding if a string is a palindrome which is $O(n^2)$ in the 1-tape Turing machines model and $O(n)$ in the 2-tape Turing machines model.

If we adopt the computation model of the k-tape Turing machines, some results help understanding the limits on the savings that can be expected either by increasing the number of tapes or by designing more efficient machines doing the same computation.

A first result [27] says that we can't expect more than a quadratic saving through allowing an arbitrary number of tapes.

Theorem 3.1. Given any k-tape Turing machine M operating within time $f(n)$, it's possible to construct a 1-tape Turing machine M' operating within time $O(f(n)^2)$ and such that for any input x, $M(x) = M'(x)$.

The meaning of this result is that the best k-tape machine that can be designed for doing a computation will never operate in less that $O(f(n)^{1/2})$ if the best 1-tape Turing machine doing the same computation operates in a time $O(f(n))$.

More generally, the Cobham-Edmonds thesis [9, 17] states that a problem has time complexity t in some "reasonable and general" model of computation if and only if it has time complexity $poly(t)$ in the model of 1-tape Turing machines.[3] The time complexity of the problems in all reasonable models is polynomially related.

A second result [27] is known as linear speed-up.

Theorem 3.2. For any k-tapes Turing machine M operating in time $f(n)$ there exists a k'-tapes Turing machine M' operating in time $f'(n) = \varepsilon f(n) + n$ (where ε is an arbitrary small positive constant) which simulates M.

This linear speed-up means that the main aspect of complexity is captured through the function $f(n)$ irrespectively of any multiplicative constant. $\mathbf{DTIME}(f(n))$ is the class of functions[4] computable by a k tape Turing machine in $f(n)$ steps. This result means that $\mathbf{DTIME}(f(n)) = \mathbf{DTIME}(\varepsilon f(n))$ and so it's legitimate to define $\mathbf{DTIME}(f(n))$ as the class of functions computable by a Turing machine in $O(f(n))$ steps.

More interesting is the Time Hierarchy theorem [17] which states that for 2-tape Turing machines:

[3] $Poly(t)$ stands for any function polynomial in t.
[4] More precisely the class of decision problems.

Theorem 3.3. $\mathbf{DTIME}(f(n))$ is strictly included in $\mathbf{DTIME}(f(n)g(n))$ when $g(n) = \omega(\log n)$ and $f(n) > n$.

For example, there are some functions that are computable in $\mathrm{O}(n^2 \log^2 n)$ and not computable in $\mathrm{O}(n^2)$. Using the Cobham-Edmonds thesis this gives similar hierarchy theorems for any reasonable models of computation.

A simple consequence is theorem 3.4:

Theorem 3.4. A universal Turing machine cannot be significantly sped-up (more than by a factor $\mathrm{O}(\log^2(n))$).

Proof: The reason why such a universal Turing machine cannot be significantly sped-up is the following. A multi-tape universal Turing machine needs only be slower by logarithmic factor [17] compared to the machines it simulates. Assume T computes in time $\mathrm{O}(t(n))$, then $U(i)$ computes in time $\mathrm{O}(t(n) \log n)$. If U could be significantly sped-up then any function simulated by U would as well be sped-up the same way, up to the factor $\log n$. Then, let's consider a function f in $\mathbf{DTIME}(O(n^2 \log^2 n)) - \mathbf{DTIME}(O(n^2))$. Speeding-up U by a factor $\mathrm{O}(log^2(n))$ would mean being able to compute f through U in a time $\mathrm{O}(n^2)$, which is impossible.

4 Chaitin-Kolmogorov Complexity and Bennett's Logical Depth

We give here the basic notions of algorithmic complexity and logical depth. A good reference is the book by Li and Vitanyi [22] and others [21, 7, 22, 6, 36, 38]. We'll then explore the possible conceptual links between these two notions and CIR.

4.1 The Chaitin-Kolmogorov complexity

Intuitively, the Chaitin-Kolmogorov complexity [5] (sometimes called simply algorithmic complexity) $K(s)$ of a string s is the length of the shortest computer program p that outputs s if it is given as input to a universal machine $U : K(s) = min\{\ell(p)|U(p) = s\}$ where $\ell(p)$ is the length of p.

This program is called the minimal program or the shortest description for s and sometimes noted s^*. It is possible to show that this definition depends on the choice of the universal machine only up to a constant. More precisely:

[5] We present here the so called "prefix complexity" in which no program is a proper prefix of another program. There are many technical reasons for imposing this restriction. See [22] for an extensive presentation.

Theorem 4.1.1. (invariance theorem). Given two Universal Turing machines U and V, there exists a constant C_{UV} depending only of U and V such that for every $s : |K_U(s) - K_V(s)| < C_{UV}$.

Proof: Roughly, this can be understood through the fact that the universal Turing machine U can be simulated through the other one V by a simulation program p_{VU}. So, if s^*_U is the minimal program for s relatively to U, $p_{VU} \circ s^*_U$ (where $\circ$ stands here for composition) is a program computing s on V. Hence $K_V(s) \leq \ell(p_{VU} \circ s^*_U) = \ell(p_{VU}) + \ell(s^*_U) = K_U(s) + \ell(p_{VU})$ (and vice versa).

This invariance theorem gives all its interest to the definition because it states that the algorithmic complexity of a string is a good measure (choosing a universal Turing machine is roughly similar to choosing the zero of a temperature scale for a thermometer). It also shows that from an asymptotic point of view, the complexity of a string does not depend on the chosen machine (the simulation program becomes negligible).

The algorithmic complexity of a string s is a measure of how regular is the string. If the string contains many redundancies it will be easy to compress and its complexity will be low. For instance, the string $(01)^{1000}$ of one thousand times "01" is easy to describe in a way much shorter than its length (we just did it). On the contrary, if there is no redundancy, the only way to describe a random string is to enumerate all of its bits. So the shortest program with the output s will be "print s" and its length will be of the order of magnitude of the length of s (plus the length of the program print). A finite string will be random (i.e. with no redundancy at all) if its complexity is roughly equal to its size. Equivalently, that means that it is not compressible. The right statistical definition of a random infinite string has been given by Martin-Löf[6] [12, 13, 8, 14]. An interesting point is the link between the algorithmic complexity and randomness for infinite strings.

Theorem 4.1.2. (infinite binary sequence). An infinite binary sequence s is random if there is a constant c such that for all $n : K(s_{1:n}) \geq n - c$ (where $s_{1:n}$ stands for the initial segment of the n first bits of s)

Another point worth noticing is the fact that among the 2^n binary strings of length n, less than a proportion of 2^{-p} have a complexity smaller than $n - p$. The reason why is easy to understand. There are less than 2^k strings of length inferior to k, then less than 2^k programs of length inferior to k. So there is a maximum number 2^k of strings of complexity smaller than k. Then the proportion of strings of complexity smaller than k among the 2^n strings of length n is less than 2^{k-n}. Equivalently (by letting $k = n - p$) less than a proportion

[6] [24] In precise terms s is random if it is not contained in any G_δ set determined by a constructive null cover.

of 2^{-p} have a complexity smaller than $n-p$. That means that almost all strings are incompressible and hence random.

4.2 Bennett's Logical Depth

Bennett's logical depth [3, 4, 1, 2] is an attempt to measure the amount of non random or "useful" information in a string. Roughly, the logical depth of a string s is defined as the time required by a universal Turing machine to generate s from its shortest description [7]. Note that this time is at least equal to the length of the string s since s has to be written (which needs at least as many steps as the number of bits of s).

The computation models used to estimate the time of computation need to be reasonable. Bennett considers what he calls "fast universal Turing machines". For example, these machines must be able to run a "print s" program in a time linear in the length of s. More generally, a fast universal Turing machine must be able to simulate any computation done on another machine in a time bounded by a polynomial linear in the time needed by the other machine, whatever other machine is considered.

Intuitively, an object is deep if it contains some redundancy (hence is not random), but an algorithm requires extensive resources to exploit that redundancy.

It's also possible to define the depth of a string s relative to another string w, which is the computation time to produce s from w by the minimal program of s. The more the depth $D(s)$ of a string s will be large relative to its length n, the deeper s will be. For example a string s with $D(s) = O(n^n)$ will be deeper than a string s' with $D(s') = O(n^2)$. Some strings are supposed to have a non linear depth in this precise meaning. That's the case of the string composed of the n bits $(a_1, a_2, \ldots, a_n)$ where a_i is 1 if the i^{th} Turing machine (under a given numbering) halts and 0 otherwise. A problem with the logical depth is that it is not computable (there is the same problem with the algorithmic complexity). As a result it's often impossible to prove things rigorously. A large literature has been extensively written these last years on the subject and some authors [8] have proposed more sophisticated but computable definitions of depth.

In this paper, we are mainly interested about the possible conceptual links between logical depth and CIR. Hence, we will not dive into technical subtleties out of the scope of our subject and will adopt the simplest (if not the most correct) definition of the logical depth: $D(s)$ is the computation time of the minimal program for s.

[7] Actually, for technical reasons, this definition is not totally satisfying and the correct one is: the depth at significance level l of a string s is the least time required by a universal Turing machine to generate s by a program that is not compressible itself by more than l bits. We'll ignore this subtlety in the following.

[8] See for example [25].

The main point about the logical depth is that it can easily be seen that it is a measure of the amount of useful information while the algorithmic complexity is clearly a measure of random information.

A random string with maximal algorithmic complexity (of order of magnitude of its length) is not deep. Logical depth and algorithmic complexity are complementary notions. A deep string can be seen as a highly organized object or as an object produced by a long computation.

A string with a simple organization (n times "0" for instance) can be deep if n is deep but will always have a low depth relative to its length. For a string to be really deep, it will necessarily be produced after a long computation having let some tracks in its structure. The slow growth law [3] expresses the fact that it is very unlikely that a deterministic program transforms quickly a shallow string into a deep one. This is an indication that the logical depth concept is reaching its goal of describing the organization hidden in a string and that it is a very important mathematical notion that can even be used in concrete applications [36].

5 Elementary Cellular Automata

For the sake of simplicity and as a useful intuitive guide but without any loss of generality, we will address the problem of defining the computational irreducibility first inside the framework of Elementary Cellular Automata (ECA). Cellular automata (CA) were originally introduced by von Neumann and Ulam [28, 31] in the 1940's as a possible way of simulating self reproduction in biological systems. A CA is a dynamical system composed of a lattice of cells inside a one or many dimensions array. Each cell can contain a value from a given finite alphabet. The system evolves in time according to an update rule that gives a cell's new state as a function of the values of the other cells in a given neighborhood (for instance the eight immediate neighbors in a square array). A ECA is a one dimensional CA which has two possible values for each cell (0 or 1) and update rules that depend only on the two nearest neighbor values. According to Wolfram [32, 34]:

> the evolution of an elementary cellular automaton can completely be described by a table specifying the state a given cell will have in the next generation based on the value of the cell to its left, the value the cell itself, and the value of the cell to its right. Since there are $2 \times 2 \times 2 = 2^3 = 8$ possible binary states for the three cells neighboring a given cell, there are a total of $2^8 = 256$ elementary cellular automata, each of which can be indexed with an 8-bit binary number.

A point worth noticing is that the number of update rules to perform to reach the n^{th} configuration going successively through all the configurations is growing at most as n^2. That's easy to see through considering an initial state with only one black cell. Then computing the next configuration needs applying one of the

8 rules describing the automaton. At the next step, there can be at most 3 black cells, then 5 black cells and so on. If we number 0 the initial configuration, the n^{th} configuration contains at most $2n+1$ black cells. Thus, provided that no intermediate configuration collapses, the number of rules to apply to reach the n^{th} configuration is: $3+5+\ldots+(2n-1) = n^2-1$. If the length of the initial configuration is l, then the number of update rules to perform will be $l+2$ for the next configuration, then $l+4$, and so on... The total number of update rules to perform will be at most $n^2+n(l-1)-1$.

Does that mean that the computation of the n^{th} configuration of every ECA among the 256 possible ones always needs to perform n^2 rules? No, since some ECA's show trivial behavior [9]. Rules 0, 40 or 96 give immediately vanishing configurations. Rules 4,12, 36 or 76 give stable configurations with a unique black cell under the initial one. These ECA can be simulated [10] in time $O(n)$ and of course the computation of the n^{th} configuration can be done in a constant time since all the configurations are identical. A little bit more interesting are the rules 2, 6, 16, and many similar others giving rise to a sloping black line of one cell length. The computation of the n^{th} configuration can be done directly in time $O(n)$. The simulation can be done either in time $O(n^2)$ or time $O(n)$ depending on the fact that we demand that all the successive configurations keep written or not. In the following we'll just demand that all the configurations appear and not that they keep written. So, the simulation can be done in $O(n)$.

Consider now the ECA rule 158 (see Fig.1). It's easy to see that the n^{th} configuration is always a string of length $2n+1$ with the following structure: for n odd, 1110011001100... ..0011 and for n even, 111011101110... ..11101. So, while simulating rule 158 needs a time $O(n^2)$, it is not complicated to compute directly the n^{th} configuration in a time $O(n)$.

The same situation appears with many other ECA. Consider the ECA rule 90 (see Fig.2). The situation is simple since the 2^nth configuration is the string "$1(0)^{2n-1}1$". The configuration 2^n+1 is the string "$1(0)^2n-11$". Here again, while the simulation of rule 90 needs a time $O(n^2)$, it's easy to compute directly the n^{th} configuration in time $O(n)$.

For all these automata (rules 2, 6, 16, ... giving only a slopping black line, rules 90, 158, ... giving more complex but regular configurations) computing the n^{th} configuration can be done in time $O(n)$. However this is not always the case. Consider for example the ECA rule 30 (see Fig.3).

It seems much more difficult to see any reasonable way to find a rule giving directly the structure of the n^{th} configuration. That seems as well difficult with ECA rule 110 (see Fig.4).

[9] In the following, we'll consider only the behavior of ECA from an initial state with only one black cell.

[10] A simulation of a ECA A is the enumeration of the successive configurations of A. We'll define below a ECA Turing machine representing a ECA A as a Turing machine computing successively all the configurations of A.

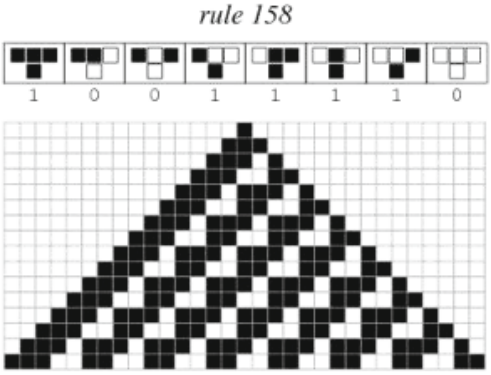


Fig. 1.

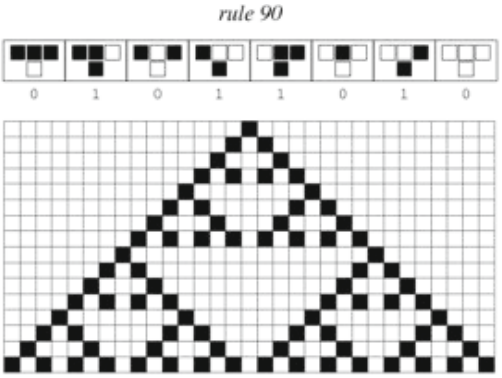


Fig. 2.

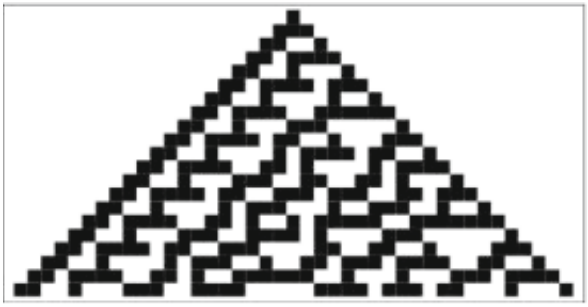


Fig. 3.

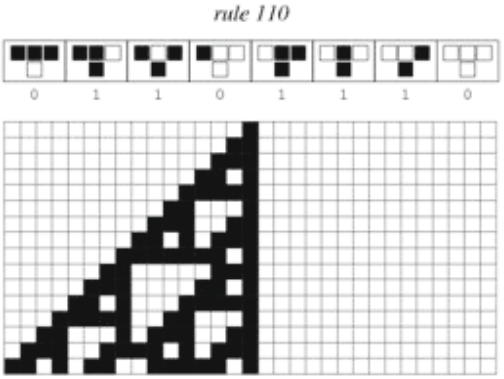


Fig. 4.

That looks even more difficult if we have a look at a larger number of configurations (see Fig.5).

Rule 110 is of particular interest since it has been proven [34, 10] to be capable of universal computation. It's up to now the simplest rule known to be universal.

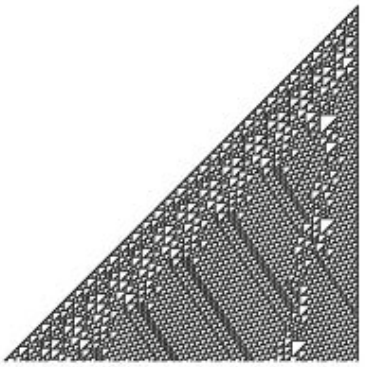


Fig. 5.

ECA	Simulation time	E_n direct computation time	Class
Rules, 0, 8, 32, 40, 96, ... and Rules 4, 12, 36, 44, 76 ...	O(n)	O(1)	1
Rules 2, 6, 16, 24, ...	O(n)	O(n)	2
Rules 18, 26, 90, 158, ...	O(n^2)	O(n)	3
Rules 90, 110, ... ??	O(n^2)	?	4

We can draw a classification[11] according to the following classes (noticing that the simulation time must be at least $O(n)$ since a simulation is the enumeration of n successive configurations and that the direct computation time can't be greater than the simulation time):

Class 2 and 4 are the only possible classes for CIR ECA. Automata in class 2 have the property that the number of cells that change between two successive configurations is bounded by a constant and so is the time to compute the $(n+1)^{th}$ configuration from the nth configuration. We don't want to consider them as CIR. Indeed, their behavior is very simple and hence can easily be predictable. So the only possible class for CIR ECA is class 4. The consideration of such automata naturally raises the question of whether it is possible – but just not obvious – to find a rule for directly computing the n^{th} configuration of such automata or if this is really impossible as claimed by Wolfram. So the question is: "is there any automaton in class 4?". This question can actually be split into:

1. is it possible to compute the n^{th} configuration without having to perform n^2 rules?
2. is it possible to compute the n^{th} configuration without computing the $(n-1)$ previous ones?

An answer "no" to the second question implies an answer "no" to the first one if all the configurations have the maximum length $2n+1$, but the reverse is not true. And the implication is not true if the lengths of the configurations are $O(n)$. To give a more precise meaning to these questions, we now address the computation model in which we'll try to state them.

ECA can be simulated through 1-tape 2-symbols Turing machines if we adopt the following convention: a Turing machine T will be said to simulate an automaton if, starting with the number n and the initial configuration (numbered 0) on its otherwise blank tape, it stops with the n first configurations of the automaton written on its tape. How are we to recognize the different states? A simple convention makes that easy: let's say that the number representing a configuration will be written with all the symbols doubled. For instance, starting from the left to the right, a line containing two black cells followed by a blank one and then a black one (1101) will be represented on the tape by "11110011". This insures that the string "01" with 0 at an odd position will never appear and can be reserved for separating the different lines. Thus, at the end of the computation, the tape will start by "01" followed by the value of the first line (with figures doubled) then "01" separating the first line from the second one and so on...The tape will be by ended by "01". For example:

01110111001101111100110001

[11] Be careful not to confuse this classification with Wolfram's classification which bears some similarities with this one.

describes 3 lines. The first line with a black cell, the second line with a black cell, a blank one and then a black one, and the third line with cells black, black, white, black, white. This convention allows as well to specify the initial state of the ECA we want to simulate. At the beginning, the tape will contain "01" then the number n of configurations to be computed (written with figures doubled) then "01" then the initial configuration (written with the same convention).

It's easy to see that this way of encoding ECA's through 1-tape 2-symbols Turing machines is not very efficient. The head will have to go back and forth permanently: back to read the three cells of the configuration $i-1$ that correspond to the current computed cell of the configuration i, forth to write the current cell, back again to read the next three cells of the configuration $i-1$, forth to write the next cell and so on. Since the configuration i has $(2i+1)$ cells, the head will have to go back and forth $(2i+1)$ times. During each trip, it will go through $2(2i-1)$ cells. With our coding that implies $4(2i-1)+4$ moves. Thus it will need $(2i+1)[4(2i-1)+4]$ steps for computing the configuration i from configuration $i-1$ (apart some details that doesn't change the argument). The total number of steps for computing the n^{th} configuration is then $\Sigma(2i+1)[4(2i-1)+4]$ for $i=1$ to n. Since $\Sigma(2i+1)[4(2i-1)+4] = O(\Sigma i^2) = O(n^3)$, we compute ECA in $\mathrm{O}(n^3)$. We could save space in allowing 3 symbols. That would avoid doubling each bit. But the gain would only be linear.

Is it possible to be more efficient? Yes, it suffices to use 2-tape 2-symbols Turing machines. When the computation starts, one tape contains the number of configurations that is to compute and the initial configuration, the second tape is blank. Now, the head of the first tape reads the initial configuration and the head of the second tape writes the next configuration according to the update rules. It's easy to see that reading configuration $i-1$ and writing configuration i can be done in $(2i+1)$ steps (here, no need to go back and forth). When configuration i is written the role of the heads is reversed. If the initial configuration has length 1, apart from a small subtlety to deal with the counter n allowing to stop after having written n configurations (which just adds a linear number of steps), the total number of steps when the machine halts is $\Sigma(2i+1)$ for $i=1$ to n: i.e. $\mathrm{O}(n^2)$. A length of the initial configuration strictly greater than 1 will only have a constant multiplicative impact.

Can we save more time? Here the answer is no for unless an intermediate computation collapses, we have seen that the number of updating rules to perform is n^2-1. So we can say that in general the most efficient Turing machines for simulating ECA's can be chosen among 2 tape 2-symbols Turing machines computing in $\mathrm{O}(n^2)$.

Definition (ECA Turing machine). Let's denote by E_n the n^{th} state of the ECA A. A Turing machine T_A will be called a ECA Turing machine representing A if:

- For all n, T_A computes E_n on input n. (It's important to notice that this is the same Turing machine which on input n computes E_n : E_n is uniformly computed by T_A).
- during the computation, the T_A tapes contain successively in an increasing order from $i = 1$ to $n - 1$, the configurations E_i.

In the following, we will abbreviate "for all n, T computes E_n on input n" by "T computes every E_n".

A ECA Turing machine representing an automaton A is exactly a program simulating the behavior of the automaton.

We can now translate the question to decide if an automaton A is CIR or not in a question expressed in terms of ECA Turing machines representing A: Let A be a ECA and let T_A be a ECA Turing machine representing A and running in time $\mathrm{O}(n^2)$:

1. is it possible to find a Turing machine which on input n computes E_n faster than T_A?
2. is it possible to find a Turing machine which on input n computes E_n without computing the $(n-1)$ previous E_i (i.e. which is not a ECA Turing machine representing A)?

6 Tentative Definitions

We'll consider first definitions in the framework of ECA Turing machines but, remembering that we seek a general definition, we shall come back to the case of functions $f(n)$ from N to N to check if these definitions are robust enough.

Before giving our preferred definitions, it's worth presenting previous attempts linked to specific intuitions. These definitions, while intuitively appealing, don't work for reasons that preclude to use them as capturing correctly the concept of CIR.

6.1 Definition Linked to the Algorithmic Complexity

Tentative definition 1 (CIR). A ECA will be said CIR if and only if $\exists c > 0, \forall n, K(E_n) > K(n) + c$.

The intuition behind this definition is that the length of the states grows as n and thus that the ECA is not evolving through fixed states or states that oscillate or vanish. In a certain sense, the automaton keeps the memory of the number of iterations for reaching a given configuration. But this definition is too weak since it is respected by many automata that are obviously not CIR: for example, the automaton which, starting from a unique black cell, adds a black cell on each side of the previous configuration. The n^{th} configuration contains $2n - 1$ contiguous black cells so $K(E_n)$ is of the order of magnitude of $K(n)$ but this automaton

is not CIR. Besides that, this definition, while a priori appealing for ECA is not applicable to function $f(n)$ like predicates for example, since the only values that a predicate can take are 0 or 1. So the complexity of $f(n)$ can't grow.

6.2 Definition Linked to the Logical Depth

Tentative definition 2 (CIR). A ECA will be said CIR if and only if $\forall n, D(E_{n+1}) > D(E_n)$ where D, the Bennett's logical depth, is understood as a measure of the content of computation.

As we saw previously, the more a string is profound the more it needs time to be computed by its minimal (or near minimal) program. The intuition here is that the successive configurations of a CIR automaton should result from more and more computation. For a ECA satisfying this definition, it would be necessary to compute longer to get the $(n+1)^{th}$ configuration than to get the n^{th} one. Its configurations would be deeper and deeper. The problem with this definition is that one can imagine that the behavior of a CIR automaton is such that even if in the average the configurations become deeper and deeper, it can happen that suddenly there is a fall in the successive depths. So the following definition is preferable.

Tentative definition 3 (CIR). A ECA will be said CIR if and only if $\forall n, D(E_n) = \Omega(n^2)$

We saw that to compute the n^{th} state the ECA needs n^2 steps when each k^{th} configuration contains $2k+1$ cells. The intuition is that nothing is lost in the computation and that in the long range, about n^2 steps are necessary to compute the n^{th} configuration even if occasionally the depth of one configuration can drop down to a lower value.

What is wrong with these definitions? It seems that they capture correctly the fact that the n^{th} state needs a lot of computation in average to be produced and that it is not possible to get it quickly. Isn't it the very meaning of CIR? Actually not. Once again, the consideration of predicates is a good way to see the problem. The n^{th} state of a predicate function will be either 0 or 1 and can neither be complex nor deep.

Considering the case of possibly CIR predicates shows that the real meaning of CIR is located inside the very succession of states not inside any single state. CIR is meaningful only regarding the way the different states are related each others. Definitions 2 and 3 mean that for each state, the time required to compute it from its minimal program must be long but this minimal program can be different for each state. On the contrary, what CIR means implies that there exists no general program which can compute for every n the n^{th} state from n faster than the ECA. So that's a different meaning we need to capture. Nevertheless, this definition is interesting by itself and worthwhile to be explored. This will be done in another paper.

7 Preferred Definitions

7.1 A First Attempt

Let A be a ECA and let T_A be a ECA Turing machine representing A and running in time $\mathrm{O}(n^2)$, we saw that the question of deciding if a ECA A is CIR or not can be split into:

1. is it possible to find a Turing machine which computes every E_n faster than T_A?
2. is it possible to find a Turing machine which on input n computes E_n without computing the $(n-1)$ previous E_i (i.e. which is not a ECA Turing machine representing A)?

We now turn to the problem of giving a precise formulation of these questions. The first question will lead to the tentative definition 4 and the second question to the definition 5.

Definition (*efficient* ECA Turing machine). We will say that a ECA Turing machine M_A^{eff} representing A is an efficient ECA Turing machine representing A if no other ECA Turing machine representing A can compute the configuration E_n faster than M_A^{eff}. Let $T(M_A^{eff}(n))$ the time for M_A to compute E_n. More precisely, we will say that a ECA Turing machine M_A^{eff} representing A is an efficient ECA Turing machine representing A if for any other ECA Turing machine M_A representing A : $T(M_A^{eff}(n)) = O(T(M_A(n)))$ i.e. there are constants $c > 0$, $n_0 > 0$ such that $\forall n > n_0$, $T(M_A^{eff}(n)) \leq cT(M_A(n))$ where $T(M_A(n))$ is the time for M_A to compute E_n.

Of course when $T(M_A(n)) = O(n^2)$, it is always possible to design a ECA Turing machine representing A and computing in time greater than $\mathrm{O}(n^2)$, if for example, the related program is not efficient. But, as we saw previously, it is also possible to have $T(M_A(n)) = O(n)$ for simpler automata such that rules 4, 12, 36 and many similar others [12]. For these automata, the number of cells that change between two successive configurations is bounded by a constant and so is the time to compute the $(n+1)^{th}$ configuration from the n^{th} configuration. As we said previously, we don't want to call CIR these automata. So we will demand that $T(M_A(n)) = O(n^2)$ (i.e. automata of classes 3 and 4) and of course, we will exclude class 3. This leads to the following definition.

Tentative Definition 4 (CIR). A ECA A will be said CIR if and only if no Turing machine computing every E_n computes the configuration E_n in a number of steps less than $\mathrm{O}(n^2)$.

[12] $T(M_A(n)) = O(n)$ or $T(M_A(n)) = O(n^2)$. One could think of an intermediate situation with an automaton for which $\mathrm{O}(n^2) > T(M_A(n)) > \mathrm{O}(n)$. For example $T(M_A(n)) = O(n \log n)$. Interestingly enough, none of the 256 ECA has this property.

The intuition behind this definition is that it is impossible to compute the state E_n faster that the automaton itself. There is no way to predict the result of the computation done by the automaton because in order to know what is the state E_n, whatever general program is used, it will need as much time as running the automaton itself. What's wrong with this definition? We have just defined CIR as the fact that the most efficient program to compute E_n can't run faster than the simulation. That is of course a first step (linked to the first part of the question above) but that is not enough to capture the whole intuition about CIR. The intuition that we want to preserve is that in order to know the state of the system after n steps it is necessary to follow approximately the same path if not exactly the same. The main difficulty here is to define exactly what is meant by "approximately the same path".

7.2 The Final Definition

Definition (approximation of a ECA Turing machine). Let M_A^{eff} be an efficient ECA Turing machine representing A and computing in $T(M_A^{eff}(n))$. A Turing Machine T will be said to be an approximation of a ECA Turing machine representing A if and only if there is a function F such that $F(n) = o(T(M_A^{eff}(n))/n)$ and a Turing machine P[13]. That's a such that:

1. on input n, T computes a result r_n and halts.
2. during the computation, the T tape contains as a substring successively in an increasing order from $i = 1$ to $n - 1$, data r_i from which the Turing machine P computes E_i in a number of steps $F(i)$.

Intuitively, an approximation of a ECA Turing machine is a Turing machine doing a computation that is near the computation made by an ECA Turing machine. We are going to show below how it is possible to build a ECA Turing machine from any approximation of a ECA Turing machine. The factor $1/n$ in $o(T(M_A^{eff}(n))/n)$ takes into account the fact that there are n necessary steps to compute E_n with a ECA Turing machine and that we want the computation time of P between r_i and E_i to be much shorter than the average of the computation time between E_{i-1} and E_i.

Definition 5 (CIR ECA). A ECA A will be said CIR if and only if any Turing machine computing every E_n is an approximation of a ECA Turing machine representing A.

Theorem 7.2. If a ECA A is CIR then no Turing machine computing every E_n can compute faster than an efficient ECA-Turing machine representing A. More

[13] Here again, it's important to notice that this is the same Turing machine P which on input r_i computes $f(i)$

precisely, if M is a Turing machine computing every E_n then $T(M_A^{eff}(n)) = O(T(M(n)))$.

Proof: We shall start by proving the following result.

Lemma. Given any M approximation of a ECA Turing machine representing A, there exists a ECA Turing machine M' representing A (we'll call it the daughter of M) computing in a time $T(M'(n)) \sim T(M(n))$.

Proof: Since M is an approximation of a ECA Turing machine representing A, there are a Turing machine P and a function F associated as mentioned in the definition. Let's consider the Turing machine M' which does exactly the same computation than M but for each i, when r_i is written on its tape, which computes E_i through P from r_i in a time $F(i)$, writes E_i on its tape and resumes the computation at the exact point where it left M computation. It's clear that M' is a ECA Turing machine representing A. From the fact that $F(n) = o(T(M_A(n))/n)$ the additional time will be at most $O(T(M_A(n)))$. Hence, M' will compute in $T(M'(n)) = T(M(n)) + o(T(M_A(n)))$. Since M' is a ECA Turing machine representing A, $T(M'(n)) \geq TM_A(n)$. Hence $T(M(n)) + o(T(M_A(n))) \geq T(M_A(n))$. From that, it follows that:

$$\lim_{n\to\infty} \frac{T(M'(n))}{T(M(n))} = \lim_{n\to\infty} \frac{\circ(T(M_A(n)))}{T(M(n))} = 1 \qquad (1)$$

Therefore, $T(M'(n)) \sim T(M(n))$. M and its daughter M' compute in the same time.

We can now prove the theorem. If A is CIR, a Turing machine M computing every E_n is an approximation of a ECA Turing machine representing A. From the lemma, M', the daughter of M (which computes in the same time than M) is a ECA Turing machine representing A. So $T(M_A^{eff}(n)) = O(T(M'(n)))$ and since $T(M'(n)) \sim T(M(n))$ we get $T(M_A^{eff}(n)) = O(T(M(n)))$.

Through theorem 7.2, it is clear that if a ECA A is CIR, it will be impossible to compute its n^{th} configuration faster than A itself (i.e. an efficient ECA Turing machine representing A). Hence A will satisfy automatically the tentative definition 4. It will also be necessary to follow a path similar to the path followed by A. That's exactly the meaning (now fully formalized) of Wolfram's claim.

Remark: Previously, we explicitly discarded ECA of class 2 (with simulation time $O(n)$) because we didn't want to accept that any of them be CIR. In definition 5, we don't suppose that the efficient ECA Turing machine representing a CIR ECA must compute in $O(n^2)$. So, it seems that we open the door to find a CIR ECA A such that an efficient ECA Turing machine representing A computes in $O(n)$, provided that any Turing machine computing every E_n is an approximation of a ECA Turing machine representing A. Actually, it's easy to see that none of the class 2 ECA has this property since for all of them it is possible to compute directly the n^{th} configuration without having to compute all the previous ones.

8 Generalization

We now leave the ECA framework and switch to the generalization of the definition to any function from **N** to **N**. We mimic the same process of definitions for functions than for ECA. In the following f will be a function from **N** to **N**.

Definition (E-*Turing machine*). A Turing machine T_f will be called a E-*Turing machine* representing f if:

1. T_f computes every $f(n)$.
2. during the computation, the T_f tape contains as a substring successively in an increasing order from $i = 1$ to $n - 1$, the values $f(i)$.

A E-*Turing machine* representing a function f is a program enumerating the function f through the computation of the successive values $f(i)$. It is the equivalent for a given function of what a ECA Turing machine simulating the ECA through the enumeration of all its successive states is for the ECA.

Definition (Efficient E-*Turing machine*). We will say that a E-*Turing machine* M_f^{eff} representing f is an efficient E-*Turing machine* representing f if no other E-*Turing machine* representing f can compute $f(n)$ faster than M_f. Let $T(M_f^{eff}(n))$ the time for M_f to compute $f(n)$. More precisely, we will say that a E-*Turing machine* M_f^{eff} representing f is an efficient E-*Turing machine* representing f if for any other E-*Turing machine* M_f representing $f : T(M_f^{eff}(n)) = O(T(M_f(n)))$ i.e. there are constants $c > 0$, $n_0 > 0$ such that $\forall n > n_0$, $T(M_f^{eff}(n)) \leq cT(M_f(n))$.

Definition (approximation of a E-*Turing machine*). Let M_f^{eff} be an efficient E-*Turing machine* representing a function f. For every input n, M_f^{eff} computes $f(n)$ and halts in a time $T(M_f^{eff}(n))$. A Turing Machine T will be said to be an approximation of a E-*Turing machine* representing f if and only if there is a function F such that $F(n) = o(T(M_f^{eff}(n))/n)$ and a Turing machine P such that:

1. on input n, T computes a result r_n and halts.
2. during the computation, the T tape contains as a substring successively in an increasing order from $i = 1$ to $n-1$, data r_i from which P computes $f(i)$ in a number of steps $F(i)$.

Definition 5 (CIR *function*). A function $f(n)$ from N to N will be said CIR if and only if any Turing machine computing every $f(n)$ is an approximation of a E-*Turing machine* representing f.

Theorem 8.1. If a function f is CIR then no Turing machine computing every $f(n)$ can compute faster than an efficient E-*Turing machine* representing

f. More precisely, if M is a Turing machine computing every $f(n)$ then $T(M_f^{eff}(n)) = O(T(M(n)))$.

Proof: The proof is identical to the proof of theorem 7.2.

9 Conclusion

We derived a robust definition for the computational irreducibility and we proved that, through a robust definition of what means "to be unable to compute the state n without having to follow the same path than the computation simulating the automaton or the function", this implies genuinely, as intuitively expected, that if the behavior of an object is computationally irreducible, no computation of its n^{th} state can be faster than the simulation itself. For CIR automata, functions or processes, there is no short-cut allowing to predict the n^{th} state faster than waiting till all the steps are done. In this sense, these objects are unpredictable.

An open problem is now to prove that explicit objects are really CIR. Possible candidates are:

- $F(n) = 2^n$ (in base 10)
- $F(n) = n!$
- $F(n) =$ the n^{th} prime number
- $x_n = 4x_{n-1}(1 - x_{n-1})$ (logistic map for $x_0 \in [0, 1]$ and having a limited number of digits)
- Rule 30
- Rule 110
- .....

This opens the way for philosophical discussions and applications. Assume for example that the process leading from inanimate matter to life, or from neurons to consciousness be CIR. In this case, there could be no way to "understand" what is life or consciousness since understanding a phenomenon is being able to predict the final state from the initial conditions, or at least to anticipate intuitively what is going to happen. For a CIR process this is impossible [14]. CIR could also be a key for explaining emergent phenomena. These points will be studied elsewhere.

The concept of approximation can be extended so as to give a classification of computable functions with similar properties in terms of irreducibility, algorithmic complexity and logical depth. This will be presented in the part II of this paper (forthcoming).

[14] See a more extensive discussion of this point in [40].

References

[1] Antunes, L., Matos, A., Souto, A., Vitany, P.: Depth as Randomness Deficiency. Theory Comput. Syst. 45, 724–739 (2009)

[2] Ay, N., Müller, M., Szkola, A.: Effective complexity and its relation to logical depth. IEEE Transactions on Information Theory, 4593–4607 (2010)

[3] Bennett, C.H.: Logical Depth and Physical Complexity. In: Herken, R. (ed.) The Universal Turing Machine- a Half-Century Survey. Oxford University Press (1988)

[4] Bennett, C.H.: How to Define Complexity in Physics and Why. In: Gregersen, N.H. (ed.) From Complexity to Life: On the Emergence of Life and Meaning. Oxford University Press, New York (2003)

[5] Bishop, R.: Chaos. Stanford Encyclopedia of Philosophy (2008), `http://plato.stanford.edu/entries/chaos/`

[6] Calude, C.: Information and Randomness: An Algorithmic Perspective, 2nd edn. Springer, Berlin (2002)

[7] Chaitin, G.: Algorithmic Information Theory. Cambridge University Press (1992)

[8] Chaitin, G.: Exploring Randomness. Springer, London (2001)

[9] Cobham, A.: The intrinsic computational difficulty of functions. In: Proc. Logic, Methodology, and Philosophy of Science II. North Holland (1965)

[10] Cook, M.: Universality in Elementary Cellular Automata. Complex Systems 15, 1–40 (2004)

[11] Copeland, J.: The Church-Turing Thesis, Stanford Encyclopedia of Philosophy (2002), `http://plato.stanford.edu/entries/church-turing/`

[12] Delahaye, J.P.: Randomness, Unpredictability and Absence of Order. In: Dubucs, J.-P. (ed.) Philosophy of Probability, pp. 145–167. Kluwer, Dordrecht (1993)

[13] Delahaye, J.P.: Information, complexité et hasard. Editions Hermès, Paris (1998)

[14] Downey, R.G., Hirschfeldt, D.: Algorithmic Randomness and Complexity. Springer (2010)

[15] Gandy, R.: Church's Thesis and Principles for Mechanisms. In: Barwise, J., Keisler, H.J., Kunen, K. (eds.) The Kleene Symposium. North-Holland, Amsterdam (1980)

[16] Garey, M., Johnson, D.S.: Computers and Intractability. Freeman, New York (1979)

[17] Goldreich, O.: Computational Complexity, a conceptual perspective. Cambridge University Press (2008)

[18] Hartley, R.: Theory of Recursive Functions and Effective Computability. McGraw-Hill (1967); MIT Press (1987)

[19] Hopcroft J.E., Ullman J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley (1979); 3rd edition (with Rajeev Motwani) (2006)

[20] Israeli, N., Goldenfeld, N.: Computational Irreducibility and the Predictability of Complex Physical Systems. Phys. Rev. Lett. 92 (2004)

[21] Kolmogorov, A.N.: Three Approaches to the Quantitative Definition of Information. Problems Inform. Transmission 1(1), 1–7 (1965)

[22] Li, M., Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications. Springer (1997)

[23] Lorenz, E.: The Essence of Chaos. University of Washington Press, Seattle (1993)

[24] Martin-Löf, P.: The Definition of Random Sequences. Information and Control 9(6), 602–619 (1966)

[25] Moser, P.: A general notion of useful information. In: Neary, T., Woods, D., Seda, A.K., Murphy, N. (eds.) CSP. EPTCS, vol. 1, pp. 164–171 (2008)
[26] Odifreddi, P.: Classical Recursion Theory: The Theory of Functions and Sets of Natural Numbers. North-Holland (1989)
[27] Papadimitriou, C.H.: Computational Complexity. Addison-Wesley (1994)
[28] von Neumann, J.: Theory of Self-Reproducing Automata. University of Illinois Press, Urbana (1966)
[29] Ott, E.: Chaos in Dynamical Systems, 2nd edn. Cambridge University Press (2002)
[30] Schuster H.G., Wolfram S.: Just Deterministic Chaos an Introduction, 4th revised and Enlarged edn. WILEY-VCH Verlag GmbH & Co. KGaA (2005)
[31] Ulam, S.: Random Processes and Transformations. In: Proc. International Congress of Mathematicians, Cambridge, MA, vol. 2, pp. 264–275 (1952)
[32] Wolfram, S.: Statistical Mechanics of Cellular Automata. Rev. Mod. Phys. 55, 601–644 (1983)
[33] Wolfram, S.: Undecidability and intractability in theoretical physics. Phys. Rev. Letters 54(8) (1985)
[34] Wolfram, S.: A New Kind of Science. Wolfram Media, Inc. (2002)
[35] Zak, M., Zbilut, J.P., Meyers, R.: From Instability to Intelligence, Complexity and Predictability in Nonlinear Dynamics. Springer (1997)
[36] Zenil, H., Delahaye, J.P., Gaucherel, C.: Image Characterization and Classification by Physical Complexity. Complexity 17(3), 26–42 (2012), http://arxiv.org/abs/1006.0051
[37] Zenil, H., Soler-Toscano, F., Joosten, J.J.: Empirical Encounters with Computational Irreducibility and Unpredictability. Minds and Machines 21 (2011), http://arxiv.org/abs/1104.3421
[38] Zenil, H., Delahaye, J.P.: Numerical Evaluation of Algorithmic Complexity for Short Strings: A Glance into the Innermost Structure of Randomness. In: Applied Mathematics and Computation (2012)
[39] Zwirn, H.: Les limites de la connaissance. Editions Odile Jacob, Paris (2000)
[40] Zwirn, H.: Les systèmes complexes. Odile Jacob (2006)

Chapter 20
Computational Equivalence and Classical Recursion Theory

Klaus Sutner

School of Computer Science, Carnegie Mellon University, USA

1 Universality

Two central results in Wolfram's *A New Kind of Science* [17] have attracted close scrutiny by the computability theory community: the first is the computational universality of elementary cellular automaton rule number 110, the other the Principle of Computational Equivalence, see section 2 below.

The discovery of universality is arguably the key insight in A. Turing's seminal 1936 paper [15] where he introduced his now eponymous machines: there is a single computing machine that can emulate all other possible computing machines. While Turing established universality only with respect to other Turing machines, experience has since shown that his definitions are quite robust and extend to random-access machines, parallel machines, probabilistic machines and even quantum computation. From a modern perspective, the single universal machine takes as input an arbitrary program and corresponding data, and then simulates the execution of the program on the given data. If the computation is successful, it returns the same output as the program would have returned when executed on the given input. If the computation of the program fails to terminate, the universal simulator likewise fails to terminate. Of course, it is precisely this idea of using programs as input that has made modern digital computers possible and it was only a few years after Turing's discovery that von Neumann set out to produce a usable universal machine–with very modest resources, such as a random access store of size $32 \times 32 \times 40$ bits and an access speed of 32 microseconds. The original construction of a universal Turing machine is somewhat tedious and a lot of effort has gone into designing smaller or simpler systems that can be shown to be universal–typically by demonstrating that they can emulate an already known universal system. It is intuitively clear that there is a trade-off in the construction of a universal Turing machine: one can either keep the number of states low or the number of alphabet symbols small. For example, it is known that 24 states suffice given a binary tape alphabet; at the other extreme, two states suffice if the alphabet has size 18, [10].

One-dimensional cellular automata are one model of computation that is easily seen to be computationally universal: we can think of the one-dimensional grid of cells as the tape of a Turing machine. It is not hard to add a bit more information to a cell that indicates the position of the tape head and the state of

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 297–307.
DOI: 10.1007/978-3-642-35482-3_20

the Turing machine. The operation of the Turing machine is then easily simulated by a cellular automaton. It fact, the construction turns a highly parallel model into an artificially sequential one by performing relevant operations only in the immediate neighborhood of the special cell that indicates state and head position. It is indicative of the robustness of the notion of universality that one can build cellular automata that are capable of simulating not just Turing machines but even any other cellular automaton, a property known as *intrinsic universality*; see [17] for an example of an intrinsically universal cellular automaton. In fact, two states and a neighborhood of size 5 suffice; on the other hand, with 4 states a neighborhood of size 3 suffices. If we are content with plain universality, it was shown in [17] that 2 states and a neighborhood of size 3 suffice, though apparently not if intrinsic universality is the goal. Cellular automata with just 2 states and 3 neighbors are often referred to as *elementary cellular automata* because of their great simplicity; their *local maps* are simply ternary Boolean function. There are only 256 such functions and it is quite straightforward to set up computational experiments that examine all elementary cellular automata. To extend the local map to a *global map* operating on bi-infinite sequences of bits on chops these sequences into overlapping blocks of three consecutive bits and then applies the ternary map in parallel and synchronously to all these blocks. Here is the local map of elementary cellular automaton number 110, given as a truthtable for the ternary function $f(x, y, z)$:

x	y	z	f	x	y	z	f
0	0	0	0	1	0	0	0
0	0	1	1	1	0	1	1
0	1	0	1	1	1	0	1
0	1	1	1	1	1	1	0

It has been pointed out that if one thinks of y as a control bit, and of x and y as the actual inputs, then for $y = 0$ the effect of rule 110 is a left-shift, whereas $y = 1$ corresponds to an application of nand to x and z. The nand operation is well-known to be a universal in the realm of Boolean functions, but, of course, in the cellular automaton we are not at ease to choose the interconnections between these gates arbitrarily. As it turns out, the evolution of configurations under rule 110 spontaneously gives rise to fairly complicated, yet orderly geometric structures. One simple example can be seen if figure 1. With considerable effort one can control these geometric structures and exploit them to simulate cyclic tag systems and thus establish universality, see [17] and [1].

The universality argument for rule 110 differs in two interesting ways from other constructions in one-dimensional cellular automata. First, universal machines are typically constructed in a very careful and deliberate manner, they are not discovered in "the wild." Unsurprisingly, the type of arguments required to establish universality for a given, rather than constructed, machine are quite complicated. Since an elementary cellular automaton is described by just 8 bits, it is quite surprising and even counterintuitive that one of these simple devices should turn out to be capable of performing computations of arbitrary complexity.

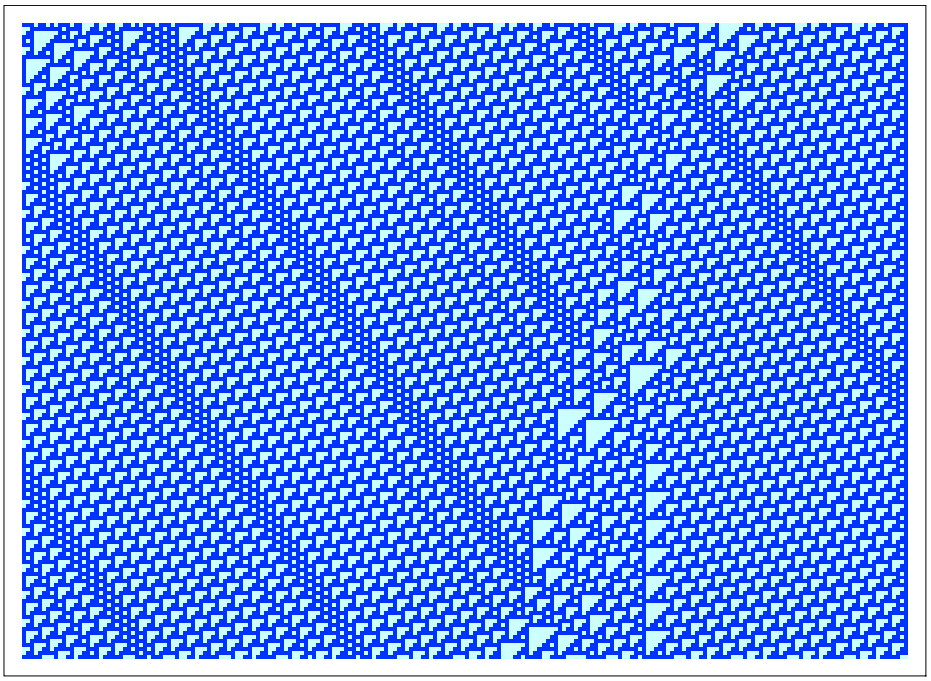


Fig. 1. A fragment of the evolution of a one-point seed configuration under rule 110

The second major difference is that the argument requires mildly infinitary configurations. In the past, in computability arguments involving cellular automata, it has been standard practice to rely on configurations of finite support, configurations of the form $\ldots 0000\, x\, 0000 \ldots$ where 0 is an arbitrarily chosen symbol in the alphabet and x is a finite string of letters. By contrast, the configurations used in the universality argument have the form ${}^{\omega}uwv^{\omega} = \ldots uuuwvvv \ldots$ where u, w and v are all finite words. Let us refer to these configurations as *almost periodic configurations* . We write $\mathcal{C}_{ap}$ for the collection of all almost periodic configurations. We can recover standard spatially periodic configurations, corresponding to finite cellular automata with periodic boundary conditions, by setting $u = v$ and $w = \varepsilon$. Similarly, configurations of finite support correspond to $u = v = 0$. While almost periodic configurations are infinite, they have a canonical finite description and can thus be handled naturally within the context of ordinary computability theory.

The choice of almost periodic configurations as a framework for the universality proof may seem *ad hoc*, but it is actually quite natural. For consider the first-order structure $\mathfrak{A} = \langle \mathcal{C}, G \rangle$ where $\mathcal{C}$ is the set of all configurations and the global map G is interpreted as the edge relation of a directed graph: there is an edge $x \to y$ if, and only if, $G(x) = y$. We refer to $\mathfrak{A}$ as the phasespace of

the corresponding cellular automaton. Then $\mathfrak{A}$ is uncountable and thus problematic from the perspective of computability theory. However, $\mathfrak{A}$ has a countable subgraph $\mathfrak{A}_{ap} = \langle \mathcal{C}_{ap}, G \rangle$. As it turns out, this subgraph is an elementary substructure in the sense of model theory: exactly the same first-order sentences hold over both structures. Note that this fails even when we consider only symmetric configurations of the form ${}^{\omega}uwu^{\omega}$.

In other words, any assertion about the short-term evolution of general configurations in a one-dimensional cellular automaton are true if, and only if, the same assertion is true for only almost periodic configurations. An observer who is limited to short-term evolution cannot distinguish between the two settings. For example, properties such as "is reversible," "has a 5-cycle" or "is 2-to-1" will hold in the full space exactly when they hold in the smaller space. In addition, there is a general decision algorithm that makes it possible to automatically verify these assertions, see [5, 6] for a description of the necessary automata theoretic machinery. The algorithm requires fairly complicated constructions such as the determinization of Büchi automata and fails to be practical even for formulae of relative modest complexity. Still, the standard quadratic time algorithms algorithms for testing injectivity, openness and surjectivity of the global map can be derived fairly easily from the general decision algorithm, see [13, 14]. Note that our restriction to one-dimensional cellular automata is critical here, the first-order theory of two or higher dimensional automata is undecidable in general. In fact, it was shown by J. Kari that even reversibility of the global map is undecidable, see [4], so sentences with only existential quantifiers do not admit a decision procedure in higher dimensions.

Of course, descriptions of phasespace from the perspective of first-order logic are necessarily quite limited. Of much greater interest are questions relating to the long-term evolution of configurations such as "does some configuration evolve to another in finitely many steps" or "does every orbit end in a fixed point?" It is precisely the long-term evolution of configurations that is critical for the proof of universality for rule 110; one has to make sure that certain structures and interaction persist indefinitely. Pinning down the necessary details is quite difficult and often requires recourse to geometric intuition. While there is no reason to doubt the accuracy of the argument, it would still be interesting to construct a proof that can be verified by a machine. Proof-assistants are still fairly complicated and difficult to use, but G. Gonthier has recently succeeded in constructing a very detailed machine-checked proof of the Four Color Theorem using the Coq proof assistant, see [3]. It is not implausible that a formal, verifiable proof could also be constructed for the universality of rule 110, in Coq or a similar environment. Note that the undecidability result also has consequences in lower complexity classes. Considerable effort has also gone into streamlining the simulations, in particular one can now avoid an exponential slow-down. For example, it was shown that it is $\mathbb{P}$-complete to predict the state of a particular cell at time t under rule 110, given an initial configuration, see [9].

2 Computational Equivalence

As Turing argued in his paper, universal Turing machines are the most complicated computational devices possible. To make this intuition technically precise it is best to consider *decision problems* solved by Turing machines, rather than the functions they compute. Formally, a decision problem is comprised of a class of instances and a subclass of Yes-instances. A decision problem is *decidable* if there is an algorithm that accepts as input an arbitrary instance and determines, in a finite number of steps, whether the given instance is a Yes-instance. For example, in number theory one is naturally interested in primality. Primality can be modeled as a decision problem where the natural numbers are the instances and the collection of primes forms the Yes-instances. Much effort has gone into developing fast algorithms to test primality and there is now a method that has running time polynomial in the size of the input (though probabilistic algorithms are vastly superior in practice).

The set of instances is always trivial, so for purposes of complexity classifications one identifies the whole decision problem with the set of Yes-instances. As it turns out, there is a natural class of decision problems that are not quite decidable, but nearly so. An excellent example, again from number theory, is the solvability of Diophantine equations, equations of the form $P(x_1, x_2, \ldots, x_n) = 0$ where P is a multivariate polynomial with integer coefficients and we are looking for an integral solution. Diophantine equations are notoriously difficult to deal with; in fact, this problem was enshrined in Hilbert's famous list of critical open problems in mathematics from 1900. For example, the quadratic equation $x^2 - 991y^2 - 1 = 0$ has a trivial solution $x = 1$, $y = 0$, but the smallest positive solution is

$$x = 379516400906811930638014896080$$
$$y = 12055735790331359447442538767$$

and is obviously quite difficult to find. It was shown by Matiyasevic in 1970 that there is no decision algorithm for Diophantine equations, [8]. But the problem is *semidecidable* in the sense that there is a brute-force search algorithm: enumerate all potential solutions $a = (a_1, a_2, \ldots, a_n) \in \mathbb{Z}^n$ in some natural order. For each a, evaluate the polynomial on a and, if the result is 0, stop and return Yes. If no solution exists the search fails to terminate. More generally, a problem is *semidecidable* if there is an algorithm that correctly returns Yes after finitely many steps when given a Yes-instance, but computes forever otherwise, without returning an answer. In a sense, the class of semidecidable problems is even more fundamental and natural than the class of decidable problems. Note that a problem is decidable if, and only if, both the problem and its negation are semidecidable. In the interesting direction, this can be seen by interleaving the computations of two semidecision algorithms.

Another way of looking at semidecidable problems is to consider algorithms that generate the set A of Yes-instances. These algorithms are non-terminating (unless the set in question is finite) and produce a sequence of objects $a_0, a_1, a_2, \ldots, a_n, \ldots$ so that $A = \{ a_i \mid i \geq 0 \}$. We can think of the algorithm as enumerating A in stages; at stage s a finite computation is performed and a_s is thrown into A. As an aside, this enumeration can be made to be strictly monotonic if, and only if, the set is decidable (again ignoring the finite case). Note that it is not allowed to remove an object from A once it has been added. Sets that can be described in this fashion are called *recursively enumerable* and it is not hard to see that they coincide with the semidecidable sets. It is this description as recursively enumerable sets that is particularly relevant to the *Principle of Computational Equivalence* (PCE).

The Principle of Computational Equivalence, proposed by Wolfram in [17], can be shortly stated like so: "... almost all processes that are not obviously simple can be viewed as computations of equivalent sophistication." The reference also states that "... all processes, whether they are produced by human effort or occur spontaneously in nature, can be viewed as computations." The latter assertion is fairly uncontroversial if one is willing to adopt a rather relaxed view of what exactly constitutes a computation. For example, is a waterfall just an arbitrary physical process or is there a specific computation being carried out whose computational purpose can be articulated in any way? Searle has warned against an overly simplified interpretation of physical computation, see [11].

So how about PCE? It seems safe to assume that the "processes" in question would include the execution of an algorithm. We can think of this execution as a logical process, or, if one prefers to stay closer to physics, as a particular physical process that corresponds to the execution of the algorithm on some particular hardware such as a digital computer. In particular we can consider a non-terminating algorithm that enumerates a semidecidable set by constructing its elements in stages. We could declare such an enumeration algorithm to be simple if there is a computational shortcut to the enumeration: instead of having to wait, potentially forever, until a certain potential element appears, we can use another algorithm to decide whether it will ever appear. In other words, the semidecidable set in question is already decidable. For example, given a monotonic enumeration (p_n) of the primes and a number q we can decide whether q is prime by generating the p_n in order and stopping when either $q = p_n$ or when $q < p_n$ for the first time. Monotonicity is critical here, this approach does not work if the enumeration is not in order. Of course, this is a rather generous interpretation of simplicity; the corresponding algorithm might be enormously complicated and require huge resources to execute. Since we are inflating the class of simple processes this should not detract from PCE.

Which enumerations fail to be simple in this sense? The ones where the computational shortcut does not exist, where the enumerated set is in fact undecidable. In his paper, Turing gave the ur-example of such a problem: the Halting

Problem, the question of whether a given Turing machine halts on some particular input. Note that the problem is semidecidable, we can simply simulate the machine for as many steps as are necessary to have it stop, if it does in fact stop at some point. If not, the simulation just goes on forever.

The Halting Problem is remarkable in that, in a strong technical sense, it encapsulates information about all possible finite computations. As a consequence, membership in any semidecidable set would be decidable if only one could get access to a fictitious database that stores information about the Halting Problem. To formalize this idea, Turing introduced the notion of an *oracle Turing machine*, a Turing machine that has additional access to an "oracle," really a database that contains complete information about some decision problem B. The Turing machine is then allowed to query the oracle: "Is x in B?" and will receive the correct answer in one step. Turing never explored his oracle machines in depth, but E. Post and S. Kleene later used his idea to introduce a partial order on decision problems, see [7]: problem A is *Turing reducible* to problem B if membership in A can be decided by an oracle Turing machine given B as an oracle, written $A \leq_T B$. If B can in turn be decided given A as oracle we say that the two problems are *Turing equivalent*. A collection of all Turing equivalent problems is called a *Turing degree* and encapsulates the notion of problems of the same level of complexity. The importance of the Halting Problem now can be seen clearly: any semidecidable problem is Turing reducible to the Halting Problem. More generally, any semidecidable problem with the property that all semidecidable problems are reducible to it is called *complete*. Thus, so far, we have two natural classes of semidecidable problems: the decidable ones and the complete ones.

Naturally one might try to find others that fail to fall into either class. The question whether there is a semidecidable problem that is incomplete but fails to be decidable has become known as Post's Problem. A positive solution is called an *intermediate set* or *intermediate degree*. Perhaps surprisingly, a careful inspection of known natural semidecidable decision problems does not turn up any examples of intermediate sets; ultimately they all appear to be either decidable or complete. While it may be quite difficult to establish either classification, there are no natural examples known to date where a problem turns out to be intermediate. Post's question is fairly straightforward, but the answer supplied by computability theory is somewhat problematic. In the 1950's, R. Friedberg and A. Muchnik, two researchers working independently of each other in the US and Russia respectively, solved Post's Problem in the affirmative; moreover, they used essentially the same technique. Their construction is now known as the *priority method* and has become the weapon of choice in computability theory.

The technical details of the construction are quite complicated, but the main idea is rather intuitive: we try to enumerate two semidecidable problems A and B with the property that neither one is reducible to the other; hence neither one can be decidable or complete. How do we make sure that, say, $A \not\leq_T B$? If we had $A \leq_T B$, then there would need to be an oracle Turing machine M that,

given B as oracle, correctly decides membership questions about A. In order to break this relation, we could try to find some number a such that M with oracle B proclaims that $a \notin A$: we can then throw a into A (recall that adding elements is the only operation at our disposal). The key problem now is that there is not just a single machine M to consider, but a whole infinite family (M_e) of oracle machines, each one of which might produce a reduction. Moreover, we also need to do the same for B with oracle A. As a consequence we have to contend with infinitely many *requirements*

$$(R_{2e}) \quad A \not\leq_T B \text{ via } M_e \qquad\qquad (R_{2e+1}) \quad B \not\leq_T A \text{ via } M_e$$

for all e. We try to discharge each one of these requirements as just indicated. Unfortunately, the requirements may clash: suppose we have just placed a into A to deal with (R_{2e}). It may happen later that because of another requirement $(R_{2e'+1})$ we place an element into B. But now we may have inadvertently changed the oracle for M_e, it might be the case that this machine, with the new B as oracle, now returns true instead of false to the query "is $a \in A$?" So (R_{2e}) is broken and we need to work on this requirement again in the future.

To deal with these infinitely many mutually clashing requirements one uses a simple priority ordering: requirement (R_s) always takes priority over (R_t) whenever $s < t$. Some care is needed to protect higher priority requirements from intrusion by lower priority ones; conversely one needs to make sure that the lower priority requirements ultimately get a chance to be satisfied. When everything is carefully arranged, a simple induction argument shows that ultimately all requirements are satisfied and we do indeed succeed in constructing two recursively enumerable sets A and B that are mutually incomparable with respect to Turing reductions.

Alas, these solutions to Post's Problem produced by a priority construction are somewhat unsatisfactory in that they are strikingly artificial. Martin Davis [2] states:

> But one can be quite precise in stating that no one has produced an intermediate recursively enumerable degree about which it can be said that it is the degree of a decision problem that had been previously studied and named.

Hao Wang [16] is even less complimentary:

> The study of degrees [of unsolvability] seems to be appealing only to some special kind of temperament since the results seem to go into many different directions. Methods of proof are emphasized to the extent that the main interest in this area is said to be not so much the conclusions proved as the elaborate methods of proof.

Of course, all this pointed criticism does not change the technical core of the result: there is a semidecidable problem A such that $\emptyset <_T A <_T H$ where H

is the Halting Problem. In fact, it turns out that the structure of the upper semi-lattice of the semidecidable degrees is highly complicated. For example, by Sacks' density theorem, for any two semidecidable problems A and B such that $A <_T B$ there there exists a third in between: $A <_T C <_T B$ for some semidecidable C. By repeating this argument one can construct a collection of semidecidable Turing degrees that that are ordered by Turing reductions just like the rationals, a less than obvious result.

3 Information Hiding and Observers

How do these results about intermediate semidecidable degrees coexist with PCE? Certainly, at first glance, any intermediate semidecidable set appears to wreak havoc with PCE: the construction is clearly an example of a process, albeit an exceedingly technical and complicated one, presumably much different from processes more closely associated with physics. However, on closer inspection, there is a clear objection to this alleged counterexample: a universal Turing machine is working in the background, without it we could not work on all the infinitely many requirements in a single construction. The intermediate sets constructed are in a sense obtained by hiding lots of information, and in particular all the details of the actual enumeration in a very deliberate way.

Let us pursue the perspective of physical processes a bit further. In physics we observe and measure certain aspects of the events unfolding and use our observations to draw conclusions about the complexity of the process. Of course, we have to trust the observer to use the right instruments, looking at the wrong aspects of the process might lead to a conclusion of simplicity when in fact something complicated is going on.

Now suppose we wish to use a similar approach to studying the Friedberg-Muchnik construction. An observer monitoring the construction from the outside will easily conclude that there is a universal Turing machine at work. Of course, we might not be using this universal machine to perform computations of maximal complexity, in which case we could still claim to have a process of intermediate complexity. Alas, no such luck: the disjoint union of A and B, meaning $\{\, 2a \mid a \in A \,\} \cup \{\, 2b+1 \mid b \in B \,\}$, is already complete, see [12]. Only if we remove B from consideration is A intermediate. So clearly the whole construction should be considered to be a "process of maximum sophistication." One might argue that the problem here lies with the particular approach taken by Friedberg-Muchnik, but it turns out that a similar argument can be made for a wide class of priority constructions. In fact, it is entirely unclear how to avoid this phenomenon: an observer with access to all the details of the construction can easily extract a complete set from it.

How, then, could one formalize a version of PCE so as to open the possibility of a proof, or perhaps of a conclusive refutation? First, we need to set up a sandbox in which the computational process takes place. A one-dimensional cellular automaton operating on almost period configurations seems like a good

choice since the elementary steps in the computation are clearly laid out in this setting. Second, we introduce an observer. At every stage of the construction, the observer has access to the finite strings u, w and v that determine the configuration. However, the computational power of the observer must be exceedingly small so as to prevent any complicated computation on his part. For example, consider a trivial process that produces the configuration $^\omega 0\, 1^n\, 0^\omega$ at stage n. If the observer had even modest computational power it could exploit the string 1^n to perform an independent computation of some n steps and produce an "observation" that has no connection to the actual process. To avoid this problem, we restrict the observer to be a finite state machine: the observer has the ability to filter out some part of the detailed process and can rewrite this part slightly, producing a finite string as output. The collection of all such observed strings is the *observation language*. Note that there always is an observer that produces a trivial observation language, say, a language containing only the empty word. But there also might be more interesting observations that can be drawn from a given process. For example, for the Friedberg-Muchnik process there is an observer that produces an intermediate observation language. Alas, there is yet another that produces a complete observation language.

This leads to a fairly natural classification of processes. A process is undecidable if there is at least one observer whose observation language is undecidable. A process is complete if there is at least one observer for it whose observation language is complete. Lastly, a process is intermediate if it is undecidable but fails to be complete: we can find an observer that produces an undecidable observation, but no observer will return a complete observation. It is a labor of love to check that known constructions of intermediate degrees, formalized in the way just outlined, all produce complete processes rather than the desired intermediate ones. We strongly suspect that in the right framework some version of PCE is provably correct. Alas, we are currently unable to establish this claim; new methods and tools in computability theory appear to be necessary to make any progress in this direction.

References

[1] Cook, M.: Universality in elementary cellular automata. Complex Systems 15(1), 1–40 (2004)
[2] Davis, M.: Foundations of mathematics (2003), `http://www.cs.nyu.edu/mailman/listinfo/fom`
[3] Gonthier, G.: Formal proof–the four-color theorem. Notices AMS 55(11), 1382–1393 (2008)
[4] Kari, J.: Reversibility of 2D cellular automata is undecidable. Physica D 45, 397–385 (1990)
[5] Khoussainov, B., Nerode, A.: Automatic Presentations of Structures. In: Leivant, D. (ed.) LCC 1994. LNCS, vol. 960, pp. 367–392. Springer, Heidelberg (1995)
[6] Khoussainov, B., Rubin, S.: Automatic structures: overview and future directions. J. Autom. Lang. Comb. 8(2), 287–301 (2003)

[7] Kleene, S.C., Post, E.L.: The upper semi-lattice of degrees of recursive unsolvability. Annals of Mathematics 59, 379–407 (1954)
[8] Matiyasevich, Y.: Hilbert's Tenth Problem. MIT Press (1993)
[9] Neary, T., Woods, D.: P-completeness of Cellular Automaton Rule 110. In: Bugliesi, M., et al. (eds.) ICALP 2006. LNCS, vol. 4051, pp. 132–143. Springer, Heidelberg (2006)
[10] Rogozhin, Y.: Small universal Turing machines. Theor. Comput. Sci. 168(2), 215–240 (1996)
[11] Searle, J.R.: Is the Brain a Digital Computer. In: Philosophy in a New Century, pp. 86–106. Cambridge University Press (2008)
[12] Soare, R.I.: The Friedberg-Muchnik theorem re-examined. Canad. J. Math. 24, 1070–1078 (1972)
[13] Sutner, K.: Model checking one-dimensional cellular automata. J. Cellular Automata 4(3), 213–224 (2009)
[14] Sutner, K.: Cellular automata, decidability and phasespace. Fundamenta Informaticae 140, 1–20 (2010)
[15] Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. P. Lond. Math. Soc. 42, 230–265 (1936)
[16] Wang, H.: Popular Lectures on Mathematical Logic. Dover, New York (1993)
[17] Wolfram, S.: A New Kind of Science. Wolfram Media, Champaign (2002)

Part VII

Reflections and Philosophical Implications

Chapter 21
Wolfram and the Computing Nature

Gordana Dodig-Crnkovic

Computer Science Laboratory
School of Innovation, Design and Engineering
Mälardalen University, Sweden
gordana.dodig-crnkovic@mdh.se

Abstract. Stephen Wolfram's work, and especially his New Kind of Science, presents as much a new science as a new natural philosophy-natural computationalism. In the same way as Andrew Hodges, based on Alan Turing's pioneering work on computability and his ideas on morphological computing and artificial intelligence, argues that Turing is best viewed as a natural philosopher we can also assert that Wolfram's work constitutes natural philosophy. It is evident through natural and formal computational phenomena studied in different media, from the book with related materials to programs and demonstrations and computational knowledge engine. Wolfram's theoretical studies and practical computational constructs including *Mathematica* and Wolfram|Alpha reveal a research program reminiscent of Leibniz' Mathesis universalis, the project of a universal science supported by a logical calculation framework. Wolfram's new kind of science may be seen in the sense of Newton's Philosophiæ Naturalis Principia Mathematica being both natural philosophy and science, not only because of the new methodology of experimental computer science and simulation, or because of particular contributions addressing variety of phenomena, but in the first place as a new unified scientific framework for all of knowledge. It is not only about explaining special patterns seen in nature and models of complex behaviors; it is about the computational nature derived from the first computational principles. Wolfram's as well as Turing's natural philosophy differs from Galileo's view of nature. Computation used in modeling is more than a language. It produces real time behaviors of physical systems: computation is *the way nature is*. Cellular automata as explored by Wolfram are a whole fascinating computational universe. Do they exhaust all possible computational behaviors that our physical universe exhibit? If we understand *physical processes as computations* in a more general sense than the computations performed by symbol manipulation done by our current computers, then universal Turing machines and universal cellular automata exhibit only a subset of all possible information-processing behaviors found in nature. Even though mathematically, there is a principle of computational equivalence, in physical nature exists a hierarchy of emergent processes on many levels of organization that exhibits different physical behavior and thus can be said compute with

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 311–323.
DOI: 10.1007/978-3-642-35482-3_21

different expressive power. This article argues that, based on the notion of computing nature, where computing stands for all kinds of information processing, the development of natural computationalism have a potential to enrich computational studies in the same way as the explorations in the computational universe hold a promise to provide computational models applicable to the physical universe.

1 Evolving Ideas of Systèmes du Monde

Cosmogonies as accounts of the origin and the nature of the universe evolve with growth of human knowledge through allegories, myths, models, theories and paradigms. This development goes in parallel with the increase in the size of the known universe – from immediate surroundings in the age of great myths, to the earth, solar system, Milky Way, to astonishing 500 billion galaxies - according to current state of knowledge. After a long history of mythopoethic and allegoric accounts of the origins and functioning of the universe, Antiquity formulated first natural philosophical and scientific theories. For Pythagoras, numbers were the essence and the principle of the universe, while for Plato geometry was fundamental. Plutarch (Convivialium disputationum, liber 8,2) reports: "Plato said God geometrizes continually". This was in modern times re-interpreted by Gauss as "o theos arithmetizei," or "God computes", [20]. Irrespective of the choice of arithmetic or geometry, the laws of the universe are governed by mathematical principles, even though one is discrete and the other continuous.

Leibniz (1646-1716) with his philosophy of Monadology holds a special place when it comes to the Systèmes du Monde. Monads were defined as elementary automata constituting the complex world through communicating networks [14]. In the Section 18 of Monadology, Leibniz depicts a monad as follows: "All simple substances or created Monads might be called Entelechies, for they have in them certain perfection (echousi to enteles); and a certain self-sufficiency (autarkeia) which makes them the sources of their internal activities and, so to speak, incorporeal automata." Leibniz had visionary ideas about calculating machines, he introduced binary notation and argued for the essential role of formal languages [1]. Wiener, in The Human Use of Human Beings, describes Leibniz as a forerunner of cybernetics "Leibniz, dominated by ideas of communication, is in more than one way the intellectual ancestor of the ideas of this book for he was also interested in machine computation and automata." ([25], p. 19). According to contemporary informational interpretation of [24], Leibniz's monads can be interpreted as information carriers programmed by divine code to change informational contents of their internal states. The divine coding guaranteed correspondence between the activities of monads and the world of phenomena.

Système du Monde of the Clockwork (mechanistic) universe is an example of a flawlessly lawful scientifically-based universe, in the form of a perfect machine, governed by the laws of physics. Laplace (1749-1827) believed that a Supreme Intelligence, based on the laws of nature and on knowledge of the positions and

velocities of all particles in the universe at any moment could infer the state of the universe at any future or past time according to the laws of mechanics discovered by Newton (1642-1727). Even though the universe-automaton is a physical system, Galileo (1564-1642) [9] in his book The Assayer - Il Saggiatore, points to vital connection between physics and mathematics, claiming that the way to understand nature is through mathematics.

The mechanistic world is based on the following principles, Dodig Crnkovic and Müller (2011) [4]:

M1 The ontologically fundamental entities of the physical reality are physical structures (space-time & matter-energy) and change of physical structures (dynamics).
M2 All the properties of any complex physical system can be derived from the properties of its components.
M3 Change of physical structures is governed by laws.
M4 The observer is outside of the system observed.

Mechanistic models assume that the system is closed, isolated from the environment, and laws of conservation (energy, mass, momentum, etc.) thus hold. Environment, if modeled at all, is treated as a perturbation for the steady state of the system.

The limits of a mechanistic universe and determinism were uncovered by the increasing use of computers as tools of exploration, especially in the biological world. What begins to emerge nowadays is a fundamentally new paradigm of not only sciences but even a more general paradigm of the universe, comparable in its radically novel approach with its historical predecessors the Mytho-poetical Universe, the Universe of Ideal Mathematical Principles and the Mechanistic Universe. This new paradigm is dubbed Info-Computational Universe; for the details, see [3].

Our current understanding of the fundamentality of information and computation for the structure and dynamics of the natural world, has led to an articulation of the universe as a computer, a network of computational processes on informational structures.

2 The Computing Universe – Naturalist Computationalism

> Will we find the whole of physics? I don't know for sure. But I think at this point it's sort of almost embarrassing not to at least try." (Wolfram talk from the 2010 TED Conference)

The idea of computing nature (natural computationalism, pancomputationalism) is old, and in a general sense can be traced back to Leibniz. Among the first contemporary researchers sharing computational view of nature are Konrad Zuse, Edward Fredkin, Tommaso Toffoli and Stephen Wolfram, together with Jürgen Schmidhuber, Seth Lloyd, Charles Seife, and Gregory Chaitin.

Konrad Zuse was the first to suggest in 1967 that the physical behaviour of the entire universe is being computed on a basic level, possibly on cellular automata, by the universe itself, which he referred to as "Rechnender Raum" or Computing Space [28]. "The idea that space might be defined by some sort of causal network of discrete elementary quantum events arose in various forms in work by Carl von Weizsäcker (ur-theory), John Wheeler (pregeometry), David Finkelstein (spacetime code), David Bohm (topochronology) and Roger Penrose (spin networks). General arguments for discrete space were also sometimes made–notably by Edward Fredkin, Marvin Minsky and to some extent Richard Feynman–on the basis of analogies to computers and in particular the idea that a given region of space should contain only a finite amount of information." ([26], p. 1026). Zuse had the idea of "going beyond quantum mechanics in discretizing physics, a vision he shared with the late Einstein and many researchers, among others Fredkin, Toffoli, Margolus, and Wolfram." [20].

Wolfram [26], based on extensive studies of cellular automata, advocates for a pancomputationalist view as a new dynamic kind of reductionism in which the complexity of behaviors and structures found in nature are derived (generated) from a few basic structures and processes:

> I strongly suspect that the vast majority of physical laws discovered so far are not truly fundamental, but are instead merely emergent features of the large-scale behavior of some ultimate underlying rule. And what this means is that any simplicity observed in known physical laws may have little connection with simplicity in the underlying rule. So perhaps in the end there is the least to explain if I am correct that the universe just follows a single, simple, underlying rule." ([26], p. 471)

Wolfram and Fredkin ([8]), in the similar vein as Zuse, assume that the universe is, on a fundamental level, a discrete system. Following the principle that "the ultimate model of physics is to be as simple as possible" Wolfram ([26], p. 475) expects the features of the universe to emerge "purely from properties of space". This presupposes that space is the independent first principle. It is however also possible that space-time and matter-energy emerge at once; that there is no space without matter-energy. But in this context, this is a detail. The most important is the expressive power, productivity and internal coherence of models, and models can differ.

Moreover, even though discrete models possess many attractive features, physics regularly uses both. Lesne [11], argues for the necessity of continuum in physical modeling of the world. Here is the summary:

> This paper presents a sample of the deep and multiple interplay between discrete and continuous behaviours and the corresponding modellings in physics. The aim of this overview is to show that discrete and continuous features coexist in any natural phenomenon, depending on the scales of observation. Accordingly, different models, either discrete or continuous

> in time, space, phase space or conjugate space can be considered. ([11], p.185)

However the computing universe (natural computationalism) does not critically depend on the discreteness of the models of the physical world. There are digital as well as analog, discrete and continuous-state models as well as computers. On a quantum-mechanical level, the universe performs, on characteristically dual wave-particle objects, both continuous and discrete computation, Lloyd [12].

3 Turing and the Computing Nature

Not only Leibniz can be seen as a predecessor of natural computationalism, Turing can be added to the list as well, based on his conviction that machines (can be made that) can think and on his work on unorganized machines (neural networks) and morphogenesis.

Turing is well known in the first place for his contributions to the theory of computation, computer science, (Turing machine model) Turing ([23, 22]), and artificial intelligence (Turing test), but for his biographer Hodges, Turing is ultimately a natural philosopher:

> He thought and lived a generation ahead of his time, and yet the features of his thought that burst the boundaries of the 1940s are better described by the antique words: natural philosophy. ([10], p.3)

It is important to notice that Turing's natural philosophy goes further than Galileo's view about the language of nature:

> Philosophy [i.e. physics] is written in this grand book – I mean the universe – which stands continually open to our gaze, but it cannot be understood unless one first learns to comprehend the language and interpret the characters in which it is written. It is written in the language of mathematics, and its characters are triangles, circles, and other geometrical figures, without which it is humanly impossible to understand a single word of it; without these, one is wandering around in a dark labyrinth." ([9], p.237)

Computing differs from mathematics in that computers not only calculate numbers, but more importantly produce real time behaviors. Turing studied a variety of natural phenomena and proposed their computational modeling. He made a pioneering contribution in the elucidation of connections between computation and intelligence and his work on morphogenesis provides evidence for natural philosophers' approach.

Turing's paper on morphogenesis proposed a chemical model as the basis of the development of biological patterns, Turing [21]. He did not originally claim that the physical system producing patterns actually performs computation through morphogenesis. Nevertheless, from the perspective of contemporary

natural computationalism and particularly info-computationalism we can argue that morphogenesis is a process of morphological computing [6].

Physical process, though not computational in the traditional sense, presents natural (unconventional), physical, morphological computation. An essential element in this process is the interplay between the informational structure and the computational process – information self-structuring.

The process of computation implements physical laws which act on informational structures. Through the process of computation, structures change their forms, as argued in [4]. All computation on some level of abstraction can be viewed as morphological computation – a form-changing/form-generating process on informational structures [6].

4 Generation of Form by Morphogenetic and Morphological Computing

> With this background it becomes understandable that we need no **intelligent design** of complex structures, but only very simple rules for local elements that generate global structures during their evolution. ([14], p.9)

Generation of form can be studied by cellular automata based on rules defining updated configurations of a grid of cells, and equivalent rules for other simple programs, but it can also be studied in physical systems undergoing morphogenesis or metamorphoses. In such systems underlying physical laws express themselves as a computation causing changes of existing forms. This process has been studied in robotics and nano-systems [13], and recently even on the macroscopic scales in materials and architectureas computing matter and material computation [15], but it deserves more attention as a basic phenomenon of form generation in physical matter, especially intricate in living systems.

Despite the mathematical principle of computational equivalence[1], in physical nature there is a hierarchy of emergent processes on many levels of organization exhibiting different physical behaviors. Based on the notion of computing nature, where computing stands for all kinds of information processing, the development of natural computationalism enriches our understanding of computation by computational studies of physical systems, in the similar way as the explorations in the computational universe provide new models applicable to the physical universe. The process goes in both directions – from the physical to the models and the other way round [18].

[1] "Almost all processes that are not obviously simple can be viewed as computations of equivalent sophistication." ([26], pp.5 and 716-717). "Almost any dynamical system that doesn't lead to random or transparently fixed or oscillatory behavior, is likely to be a universal computer." (Goertzel, Dynamical Psychology, 2002)

5 Criticisms of the Computational Views of the Universe

In his article on Physical Computation for The Stanford Encyclopedia of Philosophy, Gualtiero Piccinini [16] presents several critical arguments against Pancomputationalism (Naturalist computationalism). The unlimited Pancomputationalism, the most radical version of Pancomputationalism, according to Piccinini asserts that "every physical system performs every computation–or at least, every sufficiently complex system implements a large number of non-equivalent computations". I argue these to be two substantially different claims. The first one, that every system executes every computation, has little support in physics and other natural sciences. Different sorts of systems perform different sorts of dynamical behaviors. The second claim, that a sufficiently complex systems implement a large number of different computations, is in accordance with natural sciences and essentially different from the claim that every system performs every computation [4].

As for the sources of Naturalist computationalism, Piccinini identifies several:

One source is "a matter of relatively free interpretation" which computation a system performs. This may well be true of human computational devices like fingers, pebbles, abacuses, and computers even though interpretations once chosen are kept constant (thus no longer free), in order to allow social communication of results.

Another source of Pancomputationalism is the causal structure of the physical world. That claim goes one step further than the first one, actually searching for the basis of "free interpretation". We can freely chose systems used for calculation/computation, but the computational operations performed are predictable because of the laws of physics which guarantee that physical objects behave in the same way and according to physical laws so that we can predict and use their behaviour for computation.

Info-computationalism is in the Piccinini scheme based on the third source:

> A third alleged source of pancomputationalism is that every physical state carries information, in combination with an information-based semantics plus a liberal version of the semantic view of computation. According to the semantic view of computation, computation is the manipulation of representations. According to information-based semantics, a representation is anything that carries information. Assuming that every physical state carries information, it follows that every physical system performs the computations constituted by the manipulation of its information-carrying states (cf. [19]). Both information-based semantics and the assumption that every physical state carries information (in the relevant sense) remain controversial.

The use of the word "manipulation" seems to suggest a conscious intervention, while computation in general, as understood within the framework of Computing Nature/Natural Computationalism/Pancomputationalism, is a natural dynamical process that drives (through the physical interaction mechanisms) changes

in informational structures. Notwithstanding Piccinini's skepticism, there are well established theories in computer science which do exactly the job of connecting computational processes and informational structures as suggested by info-computationalism [4].

Recently, Piccinini made a substantial move in the direction of Natural Computationalism by advocating, what he calls the modest view of the physical Church-Turing thesis [17]. Here his claim in short is that not all of physical computation is Turing-machine computable. This view agrees with our best knowledge about Natural Computation today and it also brings us closer back to the Turing's work concerning unorganized machines with oracles (advice, learning).

Yet another interesting source of criticism towards Natural Computationalism and in particular Info-Computationalism is expressed in [4]:

> There might be a set of computing procedures that is larger than the one defined by Church-Turing – and there is certainly a mathematical set of computable functions larger than that computable by Turing machine (e.g. that computable by Turing's idea of his machine plus "oracle"). (...) My understanding of 'computer', as suggested by [23], is that such machines characteristically go beyond mere calculators (like those already invented by Leibniz and Pascal) in that they are universal; they can, in principle, compute any algorithm, because they are programmable – in this sense, Zuse's Z3 was the first computer (1941). If this feature of universality is a criterion for being a computer, then analog machines do not qualify because they can only be programmed in a very limited sense. (...) First, how can you guarantee that the notion of 'computing' you are using here is in any sense unified, i.e. one notion?" ([4], p.162.)

So in what way is physical computation/natural computation important? One of the central questions within computing, cognitive science, AI and other related fields is about computational modeling (and simulating) of intelligent behaviour. What can be computed and how? It has become obvious that we must have richer models of computation, beyond Turing machines, if we are to efficiently model and simulate biological systems. What exactly can we learn from nature and especially from intelligent organisms?

It has taken a more than sixty years from the first proposal of Turing test he called the "Imitation Game", described in Turing [22] p. 442, to the recent (2011) IBM's Watson machine winning Jeopardy by purely computational means. That is just the beginning of what Turing believed one day will be possible – a construction of computational machines capable of generally intelligent behavior as well as the accurate computational modeling of natural world.

6 Computation vs. Universal Computation

Computation is a process that a physical system undergoes when processing information (computing). Computation as a phenomenon is studied within several

research fields: theory of computation, including computability theory, physics, biology, logic, and so on. It is worth noticing that the German, French and Italian languages use the respective terms "Informatik", "Informatique" and "Informatica" (Informatics in English) to denote Computing, indicating close relationships between computation and information. In [3] it is argued that information constitute the structure, the fabric of the universe, while computation is synonymous with physical process that, implementing physical laws, incessantly changes informational structures.

The ability of a computer to perform universal computation (i.e. to process not only input data but also the code describing any other computing machine) is considered central. Here is the explanation given by [20]:

> The notion of universal computation is robust in the sense that any universal computer can emulate any other universal computer (regardless of efficiency and overhead), so that it does not really matter which one is actually implemented. (...) So, when it comes to their generic properties, it is not really important whether automaton universes are modeled to be Cellular Automata, Turing Machines, colliding billiard balls [8], or biological substrates.

[7] notices:

> This is also the disadvantage. It is hard to think about the properties of the members of a class when each member can do everything. The field of Computer Science has very few examples of useful or meaningful analytic solutions as to what some digital system will or won't do. On the contrary, there is a celebrated proof that, in general, there are no analytical shortcuts that can tell the future state of some general computation any quicker than doing the computation step by step (this is the so called "halting problem" for Turing Machines [23]). **There are normally no solutions in closed form. There is not yet any good hierarchy of concepts that express complex behavior in terms of simpler behavior, as is done in physics.**" (Emphasis added)

This is the core of the problem: There is no hierarchy. In physics there is natural encapsulation, so in principle separation between different levels of organization. A meta-language is information compression of the level below. However, discrete automata are all on the same organizational level, even though they show temporal development. That is why a universal automaton, which as an input takes arbitrary machine and executes its algorithm cannot do any better than the machine itself, as it operates on the same information. The way to make it possible for a universal machine to be more powerful is to first separate levels of abstraction between metalevel (universal) and object level (particular

algorithm). That is what is done in physics "for free" by self-organization based on natural laws on different organization levels (spatial scales).

In general it is not necessary for computation belonging to different classes of processes to be universal. Physical processes in quantum mechanics are different from processes in the classical clockwork universe and it is not a big problem if they are modeled differently, by different classes of computers. That is what present day physics does – it produces different theoretical frameworks for different levels of organization – from quarks to galaxies. We have different frameworks executed on the same sort of computer. In the future we can have the same framework executed on different sorts of computers.

7 Questions beyond Present Computational Experiments

Even within the world of cellular automata, there are number of interesting questions for future investigations. [14] propose:

> Going beyond the numerical experiments of Steven Wolfram, it is argued that cellular automata must be considered complex dynamical systems in their own right, requiring appropriate analytical models in order to find precise answers and predictions in the universe of cellular automata. Indeed, eventually we have to ask whether cellular automata can be considered models of the real world and, conversely, whether there are limits to our modern approach of attributing the world, and the universe for that matter, essentially a digital reality.

Instead of exploring cellular patterns from a phenomenological point of view [14] apply analytical methodology like the one used in mathematical physics.

I would add some of questions that came to my mind when reading Wolfram's book. Here are some of them.

Cellular automata get updated synchronously. How about diachronic processes? If they are modeling physical world as we know it, it should be possible to model an event originated in the past (like a photon created in the Big Bang) to interact with the informational structure in the contemporary universe (trigger a detector today). How about non-local systems?

Cellular automata and simple programs have demonstrated surprisingly rich expressive power in modeling self-organization and emergent properties in systems consisting of similar units but there are phenomena in nature that seem to be radically different: How about interactions in totally heterogeneous systems? Can they be reduced to the properties of the underlying grid?

How about evolution? How could evolution and development be implemented in the world of cellular automata? [2] for example proposes constructive mechanisms to explain (the unavoidable) evolution from thermodynamic to anticipatory (teleological) systems that are in agreement with natural computation (physical computation).

It is possible that computation on a mathematical level is "all or nothing" (computational equivalence), but if we want to ascribe computational characteristics to the physical world and explain its full complexity, we must admit that there are hierarchical structures in physical systems that have complex systemic properties. How could the architecture be build up out of form-generating algorithms? In nature there is a hierarchical succession of levels of organization and every higher level can be described by meta-language with respect to previous level. How about second order algorithms, or algorithms changing algorithms?

Could that be that the underlying cellular automaton of the universe expands producing expanding universe which we observe? What would that mean for the properties of the automaton?

8 Conclusion: The Dream of Leibniz Coming True?

> Could it be that someplace out there in the computational universe we might find our physical universe? Will we find the whole of physics? ... I think computation is destined to be the defining idea of our future. (Stephen Wolfram, TED talk, filmed Feb. 2010)

Wolfram's New Kind of Science is one of his closely interconnected projects that can be understood in relation to the Leibniz's quest for automation of reason in a universal science, Mathesis universalis (1695). Leibniz's characteristica universalis was envisaged as algebra expressing conceptual thought by a formal system based on the rules for symbolic manipulation of calculus ratiocinator. There are two opposed interpretations of Leibniz's calculus ratiocinator: the first is analytic view relating calculus to software and "algebra of logic", and the second, synthetic view, found in cybernetics, understands calculus ratiocinator as referring to a "calculating machine". This duality may be seen as reflecting the dichotomy between mathematical and physical view of computation.

The development of formal systems, Hilbert's program and the development of programmable computational machinery all contributed to the gradual realization of the formalization project of Leibniz. However, at the same time the development of human knowledge run into increasing fragmentation and specialization which has reached alarming proportions. So, for example, at present no individual can have general knowledge of physics broad enough to cover all its different fields – from string theory to astrophysics.

Wolfram's project, contrary to the general division into disparate knowledge compartments, runs towards common synthetic framework using tools of formal reasoning and *Mathematica* as calculus ratiocinator, achieving a wide-ranging synthesis of knowledge. Adding Wolfram|Alpha's capability to accumulate and compute general knowledge, this project bears a resemblance to the ambitions of Mathesis universalis, and brings renewed renaissance optimism about the human capability to know the world based on natural laws, with computation as an organizing principle of all knowledge.

References

[1] Davis, M.: The Universal Computer: The Road from Leibniz to Turing. WW Norton (2000)

[2] Deacon, T.: Incomplete Nature: How Mind Emerged from Matter. W.W. Norton & Company, New York (2011)

[3] Dodig-Crnkovic, G.: Investigations into Information Semantics and Ethics of Computing. Mälardalen University Press (2006)

[4] Dodig-Crnkovic, G., Müller, V.: A Dialogue Concerning Two World Systems: Info-Computational vs. Mechanistic. In: Dodig-Crnkovic, G., Burgin, M. (eds.) Information and Computation. Series in Information Studies. World Scientific Publishing Co. (2011)

[5] Dodig-Crnkovic, G.: Info-Computational Philosophy of Nature: An Informational Universe with Computational Dynamics. In: Brier, S., Thellefsen, T., Sørensen, B., Cobley, P. (eds.) From First to Third via Cybersemiotics, the Festschrift for Prof., p. 97. CBS Copenhagen, Denmark (2011)

[6] Dodig-Crnkovic, G.: Info-computationalism and Morphological Computing of Informational Structure. In: Simeonov, P., Smith, L., Ehresmann, A. (eds.) Integral Biomathics, PART II: Mathematics and Computation. Series on Computational Intelligence and Complexity (2012)

[7] Fredkin, E.: Finite Nature. In: Proceedings of the XXVIth Recontre de Moriond, pp. 283–297 (1991)

[8] Fredkin, E.: Digital Mechanics: An Information Process Based on Reversible Universal Cellular Automata. Physica D 45, 254–270 (1990)

[9] Galileo, G.: The Assayer. English Trans. Drake, S.: Discoveries and Opinions of Galileo, pp. 237–238 (1957, 1623)

[10] Hodges, A.: Turing. A Natural philosopher. Phenix, London (1997)

[11] Lesne, A.: The discrete versus continuous controversy in physics. Mathematical Structures in Computer Science 17, 185–223 (2007)

[12] Lloyd, S.: Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos. Knopf, New York (2006)

[13] MacLennan, B.J.: Models and Mechanisms for Artificial Morphogenesis, Natural Computing. In: Peper, F., Umeo, H., Matsui, N., Isokawa, T. (eds.) Proceedings in Information and Communications Technology (PICT). Springer series, vol. 2, pp. 23–33. Springer, Tokyo (2010)

[14] Mainzer, K., Chua, L.O.: The Universe as Automaton: From Simplicity and Symmetry to Complexity. Springer-Verlag Berlin and Heidelberg GmbH & Co. (2011)

[15] Menges, A. (ed.): Material Computation – Higher Integration in Morphogenetic Design, Architectural Design, vol. 82(2). Wiley Academy, London (2012)

[16] Piccinini, G.: Computation in physical systems. The Stanford Encyclopedia of Philosophy, Fall 2010 ed. Stanford University (2010), `http://plato.stanford.edu/`

[17] Piccinini, G.: The Physical Church-Turing Thesis: Modest or Bold? British Journal for the Philosophy of Science 62(4), 733–769 (2011)

[18] Rozenberg, G., Kari, L.: The many facets of natural computing. Communications of the ACM 51, 72–83 (2008)

[19] Shagrir, O.: Why We View the Brain as a Computer. Synthese 153(3), 393–416 (2006)

[20] Svozil, K.: Computational universes. Chaos, Solitons & Fractals 25(4), 845–859 (2005)
[21] Turing, A.M.: The Chemical Basis of Morphogenesis. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 237(641), 37–72 (1952)
[22] Turing, A.M.: Computing Machinery and Intelligence. Mind LIX(236), 433–460 (1950)
[23] Turing, A.M.: On Computable Numbers with an Application to the Entscheidungsproblem. Proceedings London Math. Soc., series 2 43, 544–546 (1936)
[24] Uchii, S.: An Informational Interpretation of monadology (2009) (Preprint), http://philsci-archive.pitt.edu/4635/
[25] Wiener, N.: The Human Use of Human Beings: Cybernetics and Society. Da Capo Press (1988)
[26] Wolfram, S.: A New Kind of Science. Wolfram Media, Champaign (2002)
[27] Zuse, K.: Calculating space. MIT technical translation AZT-70-164-GEMIT. MIT (Proj. MAC), Cambridge, MA (1970)
[28] Zuse, K.: Rechnender Raum. Friedrich Vieweg & Sohn, Braunschweig (1969)

Chapter 22
A New Kind of Philosophy: Manifesto for a Digital Ontology

Jacopo Tagliabue

Department of Philosophy, Università San Raffaele, Milan, Italy, and
Department of A.I., iLabs, Milan, Italy

Abstract. Stephen Wolfram's ambitious "A New Kind of Science" (NKS) re-thinks and re-builds almost every scientific field in the light of the study of cellular automata and their emergent behavior. A little known fact among professional philosophers is that there is plenty of room for philosophy too in NKS: on the one hand, Wolfram's core arguments require sophisticated conceptual analysis to be properly assessed and evaluated; on the other, it is pretty clear that Wolfram himself regards NKS as the obvious premise for a "A New Kind of Philosophy". In this contribution, we shall focus mainly on the second part of NKS philosophical import; in particular, we take Wolfram's own analysis as a starting point to explore the answer to the following question: do philosophers need digital philosophy? Our answer will be "yes", for Wolfram's own reasons and more. First, we argue that philosophy as a whole may benefit from the unorthodox intuitions delivered by the systematic study of CA; second, we outline three promising areas of research for this new kind of philosophy, highlighting that a digital approach to substantial and methodological issues may bring very interesting consequences for many contemporary debates in the discipline. Finally, we place digital philosophy into the wider context of contemporary sciences (computer science, Artificial Intelligence and cognitive sciences), arguing that a genuine interdisciplinary approach would help tackling in new ways the greatest challenges of these fields.

1 Introduction

Stephen Wolfram's ambitious "A New Kind of Science" (hence NKS) explicitly calls for a radical revision of pretty much every scientific field[1]: the details of the story (and arguably the strength of the argument) vary from field to field, but the discoveries made by Wolfram by studying cellular automata (hence CA) are the core of the whole book. While CA are well-known among scientists since the Eighties (invented by John Von Neumann in the Fifties [28], though it is fair to acknowledge Wolfram himself[2] for much of the popularity of the field

[1] For an overview, see the introductory section of [31].
[2] See for example [29] and [30].

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 325–339.
DOI: 10.1007/978-3-642-35482-3_22

thereafter), they are not as popular among professional philosophers, even if many computability concepts are now part of any serious practitioner's toolkit. This is very unfortunate, since there is plenty of room for philosophy in NKS: on the one hand, NKS's boldest conclusions and claims rest on fairly sophisticate notions from epistemology (what can we know?), philosophy of science (how should we judge a scientific paradigm?) and even metaphysics (what is the nature of computational phenomena?); on the other, it is pretty clear that Wolfram himself regards "A New Kind of Science" as the obvious premise for a "A New Kind of Philosophy":

> Among them [the fundamental issues philosophers address] are questions about the ultimate limits of knowledge, free will, the uniqueness of the human condition and the inevitability of mathematics. Much has been said over the course of philosophical history about each of these. Yet inevitably it has been informed only by current intuitions about how things are supposed to work. But my discoveries in this book lead to radically new intuitions.[3]

In *this* contribution, we shall focus mainly on the second aspect of NKS' philosophical import[4]. In particular, we take Wolfram's own analysis as a starting point to explore the answer to the following question: do philosophers need digital, CA-inspired philosophy? (Spoiler alert: yes, they do, for Wolfram's own reasons and some more).

The paper is organized as follows: in *Section I* we sketch a theory of systematic philosophy and in its relations with other forms of rational enquiry (mathematics and empirical science). In *Section II* we introduce NKS discoveries ("the new intuitions") and see how they can be fruitfully applied to existing debates in contemporary philosophy. Finally, in *Section III* we shall argue that NKS and CA provide much more than just new intuitions: they provide philosophers with new tools that may help shape the discipline and change its place in the world.

2 What Is Philosophy?

To outsiders - and even to insiders, from time to time - philosophical disputes seem puzzling: sometimes what is at issue is not that clear (what is a possible world, anyway?[5]), sometimes it is hard to understand why we should favor one

[3] [31, p. 10].

[4] The first - together with a philosophically friendly introduction to CA - is addressed at length in [3].

[5] Actually, the seemingly sci-fi notion of "possible world" is a fascinating theoretical concept, useful in many branches of logic and philosophy. The *locus classicus* is [21]; for an up-to-date review, see [13].

among many competing explanations (after all, there are supposedly crucial "thought experiments" in philosophy - but they are not experiments in the scientific sense, right?). To further complicate the matter, philosophers do not agree on a definition, since the scope and method of the discipline are themselves matter of dispute within the discipline. Without the ambition of settling the issue once and for all, I shall propose a working definition (which I take to be relatively uncontroversial) that will be used in what follows to better understand the importance of Wolfram's remarks and CA for philosophy.

As philosophers[6], we are committed to come up with a list of basic entities and simple rules out of which everything we "see" – atoms, people, galaxies, mathematical objects, moral values, mental states – can be built. Non-philosophers[7] can easily imagine the work as some sort of reverse-LEGO: you start with the whole model in front of you and the task is to compile the list of items that were in the LEGO kit in the first place, together with the assembly manual (the world we live in is a hell of a LEGO model, so it is no wonder that philosophers are fighting all the time about which items should make the final list). As it is stated, our game raises an immediate objection: isn't it sciences (and especially physics) that tell us what the fundamental features of reality are? if so, why bother with philosophy at all? To understand why we need philosophy and sciences alike, it is important to understand what kind of LEGO bricks philosophers are after. While it is physics that tells us, say, that particle X's behavior is causing event E, it is philosophy that tells us what "causing" means: sure, the fact that E is caused by X is grounded in some physical features of our world, but unfortunately this does not account for the concept of causation in its full generality[8]. In other words, sciences presuppose a concept - *causation* - whose fundamental structure is not explained by science: upon reflection, this is not strange at all, since science and physics also presuppose summation, whose fundamental structure is explained by mathematics, and implications, whose fundamental structure is explained by logic.

Philosopher's LEGO model is thus made by the most important qualitative features of our world (like causation, identity, morality, rationality, and so on), the concepts we use in science and everyday reasoning to make sense of what happens around us[9]. So the question becomes: among these concepts, can we single out a minimal set of fundamental ones to which we can reduce all the

[6] *Caveat*: this is especially true for ontology, which is in some sense the most fundamental branch of philosophy (a view that can be traced back to Aristotle) and the one - as we shall see - which may benefit the most from CA (not that I am partial, but incidentally, it is also my area of specialization).

[7] Non-philosophers may wish to consult [8] which is a gentle introduction to the field written for non-specialists.

[8] As a small illustration, consider: "Inflation caused unemployment". How would you even *begin* to explain this relation using physics?

[9] Saying that an account of causation is not that important will not do for obvious reasons - for example, the fact that causation is at the heart of responsibility, a founding notion of our society.

others? To further drive this point home, consider the following screenshots from Conway's famous CA[10], the *Game of Life*:

t_0 t_1 t_2

CA practitioners will recognize a so-called *glider* floating in the space. A software may surely help us calculate where the glider will be after n time-steps in the universe evolution; but what is the correct description of the situation from an ontological perspective? Take the two following characters:

A is a philosopher who agrees with a commonsensical account: we are seeing one object moving in the space.

B is a philosopher who says that no object is moving: she insists that there are no composite objects, just atomic cells. Since cells do not move, movement is impossible.

Of course, both philosophers can agree with the software calculation, since the LEGO model they see is exactly the same; however, the qualitative notions they employ in their descriptions differ a lot[11]: only for A cells can be combined to form further "emergent" objects, so that only in A's universe things may be created and destroyed. If this seems a rather unimportant fact concerning *Life*, just substitute "gliders in a CA" with *"persons* in our universe": are *we* real (or, which is the same, can we die)[12]?

Now that we hopefully have some idea of the scope of philosophy, it is time to address two crucial questions regarding its method: how did A and B reach their conclusion? How should we decide between competing theories?

To answer these questions, imagine playing the reverse-LEGO game again (say, with a LEGO cathedral). You have the model in front of you - you cannot break it, you cannot touch it - and all you can observe are the model's macroscopic features: you see the church's spires and gothic gargoyles, but you

[10] See [1]. For a demonstration of *Life* amazing capabilities, a quick Internet search will give the desired result.

[11] Of course, another philosopher, C, may argue that this is just a "semantical dispute", a problem with words, not with the world's fundamental structure. This is a respectable position, drawing on complex background in the methodology of ontology and metaphysics, but we shall not pursue it here (see [7] for a collection of works in metametaphysics). It is worth noting that metaphysical skepticism is itself a philosophical position - and one whose practitioners usually end up explaining in philosophy journals.

[12] In the example several philosophical topics are intertwined: for an introduction to the themes of composition see [6]; for the metaphysics of change see [25].

can't see how they have been built. There are countless LEGO kits compatible with the model (i.e. they are such that you can have countless indistinguishable cathedrals starting with different kits), so how do you choose[13]?

A first, obvious desideratum is conceptual economy. "Less is more", "Occam's razor", "Lex parsimoniae", "Simplex sigillum veri"[14] are all effective slogans for this principle, stating that we should favor simpler theories (theories with fewer primitive or undefined notions) over more complex ones when the explicative power is the same. In our case, we could try to solve the LEGO model by reproducing the cathedral with the fewest possible brick types: we start with just one type (say, the smaller LEGO brick) and then we introduce others only if we cannot continue with what we have already included. If we do the job with rigor, we should end up with the smallest possible list of fundamental brick types.

Unfortunately, while this makes much sense in our example, it overlooks two crucial points in solving the game: first, we do not know the "smaller LEGO brick" to begin with; second, it may be very hard to judge when we cannot continue with what we already have in the list anymore. To appreciate the former difficulty, it is enough to note that our evidence is compatible with a very simple assumption: the model is built from a unique giant brick cathedral-shaped; if we did not know that LEGO does not produce cathedral-shaped bricks, we could not rule out this hypothesis. To appreciate the latter problem, we point out that even in the presence of a complete kit, we may still be unable to assemble the cathedral: the pieces are all there, but without enough ingenuity, we simply cannot make them work - so we may be tempted to introduce ad-hoc pieces to complete the model.

In both cases, our search for a solution heavily relies on *intuitions*[15]: we have a strong intuition that the cathedral is a composite object and we have a strong intuition that the smaller LEGO brick, by itself, is not enough to produce the church's spires and gothic gargoyles[16]. But where do intuitions come from? For the most part from common sense and science. This is why philosophy may strongly benefit from a detour in other fundamental disciplines: physics and mathematics, in particular, provide us with new intuitions that often correct commonsensical arguments. To take a famous example, for centuries the infinite was considered to be without any interesting structure: thanks to Cantor, many

[13] As we shall remark, this is also a problem for different scientific theories explaining the same set of facts (the so-called "underdetermination problem", see [26]).

[14] Literally, "Simplicity is the mark of Truth".

[15] See also the wise words in [20, pp. ix-xi], and [15, pp. 13-15].

[16] In many respects, this image is an oversimplification: in particular, no two theories in general explain (equally well) exactly the same set of phenomena, so the final judgment is a trade-off between theoretical simplicity and explanatory power - a balance which may be itself matter of philosophical debate (that is why we will never run out of work). It is worth emphasizing that almost any "fundamental" discipline shares the problem: for example, competing "theories of everything" in physics are not judged over experiments, but over theoretical considerations of conceptual economy and intuitions of what may be a possible fundamental layer of reality.

of us have now changed our intuitions into a more robust conception of infinite, allowing a very rich and interesting structure (some may still believe that there are strictly more integers than even numbers; but we know better); in other words, things we believed possible turn out to be impossible, and vice versa. Of course this is not to say that physics and mathematics may solve philosophical puzzles by themselves, since, as we have seen, the structures investigated by philosophy are somewhat presupposed by both. However, while philosophers sometimes declare from the armchair the impossibility of X just because our imagination does not yet cover X, good science has always been doing a great job in broadening our horizons. To the extent that CA are good science, it should not be surprising we have so much to learn from them.

3 A New Kind of Intuitions

As we have seen philosophy is, in a strong sense, dependent upon intuitions (together with theoretical considerations of simplicity, explanatory power, etc.) Therefore, when science challenges previously held intuitions, we should expect our philosophical perspectives to be challenged as well. In a nutshell, this is also Stephen Wolfram's idea when he claims that the "discoveries in this book [NKS] lead to radically new intuitions"[31, p. 10]. What these discoveries and these intuitions are is obvious to any reader of NKS, however a quick review of the book's main themes may be useful as a reference[17]:

NKS$_1$) Simple rules may produce very complex behavior: contrary to what we think, complex behavior does not require a rich ontology to begin with (pretty much as with LEGO, it is astonishing the complexity you can achieve with a tiny set of primitives).

NKS$_2$) Computation is everywhere: computational phenomena are not limited to computers; in some sense, computation is the most fundamental process in reality, one to which everything else is to be reduced.

NKS$_3$) Universal computation is much more widespread than what we think: many systems, when investigated through the lens of computability, turn out to be equivalent in a precise mathematical sense (i.e. they all can compute the same set of functions and this set comprises every possible computable function).

NKS4) The ability to predict the world is severely constrained by (NKS$_3$): once a system is proved equivalent to a Universal Turing Machine (UTM), it can be shown that no algorithmic procedure can accurately predict its behavior – we can only run a simulation and wait for the result to be computed step by step. This, in turn, highlights the importance of computer-assisted discovery in any area of rational enquiry.

[17] *Caveat*: the list is not meant to be exhaustive nor to be the list Wolfram himself would have written; however it nicely summarizes the most important points for what follows. Moreover, while some claims can be held independently from the other, they are most compelling when presented as a single world view.

NKS_5) Digital universes' behavior is phenomenologically rich: in particular, digital universes may be indistinguishable from continuous counterparts due to their ability of hosting an incredible variety of patterns at different levels of granularity.

Taken together, (NKS_1)-(NKS_5) are a strong challenge to our intuitions. Granted, some of these claims are not new to philosophers around the globe: see, for example, the survey in [23] regarding (NKS_2), or consider the following, striking declaration from Dan Dennett:

> Every philosophy student should be held responsible for an intimate acquaintance with the Game of Life. It should be considered an essential tool in every thought-experimenter's kit, a prodigiously versatile generator of philosophical important examples and thought experiments of admirably clarity and vividness.[18]

However, it is fair to say that NKS as a whole constitutes an exceptionally strong case for what Ed Fredkin called the "Finite Nature Hypothesis":

> Finite Nature is a hypothesis that ultimately every quantity of physics, including space and time, will turn out to be discrete and finite; that the amount of information in any small volume of space-time will be finite and equal to one of a small number of possibilities. (...) We take the position that Finite Nature implies that the basic substrate of physics operates in a manner similar to the workings of certain specialized computers called cellular automata.[19]

To appreciate an original contribution of NKS to philosophy, let us analyze Wolfram's approach to the problem of free will. What is the riddle of free will? We would like all of the following statements to be true, but (here is the trick) it seems they cannot be possibly all true:

1. The physical world is deterministic[20] and causally closed[21] (i.e.: only physical events may cause physical events).
2. The world physical state at t_n together with the laws of nature is sufficient, for each m, for any physical state at t_{n+m}.
3. If we have free will, for each action A we have freely deliberated and performed, we could have done otherwise.
4. We have free will.

[18] [11, p. 37].
[19] [14, p. 116].
[20] At the level relevant to explain our actions – say, the level of biological processes in the brain. Of course, physics may not be deterministic at its bottom level (say, quantum reality), but it is obscure how this fact would vindicate freedom of the will.
[21] See for example [17] for a discussion of this claim and its relation to the mind/matter debate.

(1) is an empirical statement: if tomorrow we discover that biology is not deterministic or that non-physical entities (like ghosts or Greek goddesses) may influence physical events, we will find that (1) is indeed false. However, for scientists and philosophers holding (1) is tantamount as holding a naturalistic perspective on the world – so, it will better be true. (2) is a conceptual truth, the definition of determinism: if the world is deterministic, the big bang and the physical laws necessitate the evolution of the entire universe. (3) is basic conceptual analysis: part of what it means to have free will is to not be necessitated when one is acting. Finally, (4) is an intuitive truth: we do feel we are free – moreover, we arguably assign moral responsibility based on this assumption (i.e., you're a good person because you made a good action when instead you could have made a bad one).

The crucial question is therefore: how is the tension between (2) and (3) to be explained? Some believes that the intuition is correct: there is a tension and it cannot be resolved: a deterministic world is not compatible with us being free[22]; so, if (1) stays, it turns out that free will is like unicorns and chimeras, i.e. a fictional entity. Many philosophers, however, contend that the tension can be explained away, giving us all the free will we desire; interestingly enough, Daniel C. Dennett, one of the most important contemporary thinkers in this tradition, employs CA to argue that you cannot infer that your actions are unavoidable from the fact that the basic physics is deterministic[23]. Stephen Wolfram attempts his own analysis starting from (NKS_1)-(NKS_5):

> From the discoveries in this book it finally now seems possible to give an explanation for this [free will]. And the key, I believe, is the phenomenon of computational irreducibility.[24]

Computational irreducibility is precisely captured by (NKS_4): although a system follows definite underlying laws, "its overall behavior can still have aspects that fundamentally cannot be described by reasonable laws"[25]. Pretty much as atoms in a CA, our neurons follow simple rules; however, whenever a system reaches a threshold in its computational complexity, we simply "cannot readily make predictions about the behavior of the system"[26]: we attribute free will to humans because there is no way to accurate predict their behavior, so that the concept of freedom fills the explanatory gap between the micro-level and the macro-phenomena[27]. According to Wolfram, CA play a leading role in providing a

[22] See [27] for an excellent starting point to the contemporary discussion.
[23] See [12, pp. 40-44]. For a philosophical discussion in the context of CA, see [3].
[24] [31, p. 750].
[25] [31, p. 750].
[26] [31, p. 751].
[27] Fellow philosophers may indeed argue that Wolfram's claim is that there is no real free will, just an epistemic limitation we call "free will". We leave to the informed reader to decide whether this position is a version of "compatibilism" or some form of "eliminativism".

new framework to understand the phenomenon: while explanations from chaos theory and quantum randomness have been recently proposed, "nothing like this is actually needed"[28]: thanks to the new kind of science, we have now a precise model of how the explanatory gap works. Good science has shown us once more how to crack a riddle (how unpredictability arises from micro-determinism) we previously didn't understand[29].

4 Philosophy in a Digital World

Notwithstanding the importance of other NKS themes for existing philosophical discussions[30], we wish to further pursue Wolfram's claim about the global philosophical import of CA. As Andrew Ilachinski points out, CA can be fruitfully used as "conceptual vehicles for studying pattern formation and complexity"[31]: if philosophy can be thought of some kind of reverse-engineering of reality's structure, CA are a most precious conceptual lab to explore emergent behavior in its purest form. In particular, we shall explore three different ways in which CA may improve the method of philosophy as it is actually practiced – all of them are expansions of Wolfram's core idea on the import of CA-based intuitions for knowledge in general.

4.1 Reverse-Engineering Reverse-LEGO

A first obvious point in the "New kind of philosophy" Manifesto is the use of digital universes (say, *Rule 110* or *The Game of Life*) to model existing philosophical theories: what happens, in other words, if we let ourselves apply philosophical reasoning in a world whose starting kit is known a priori? As we have seen before[32], CA may provide the purest environment in which different intuitions may be vividly tested and theories may be compared: while CA *per se* (just as mathematics and physics) cannot solve, say, the metaphysical problem of change, observing the arguments involved in the dispute in the context of a digital universe may help bring some clarity to the whole debate.

Moreover, philosophical proposals have been developed in connection with our everyday reality, but it is not always clear how to assess them due to the intricate network of concepts and intuitions that come in play when we observe the complex world we live in - maybe, simpler yet rich worlds, may help. As an example, consider the following two sentences:

[28] [31, p. 752].

[29] Of course, Wolfram's argument hardly settles the issue once and for all. However, it's very important to recognize how the creative, yet rigorous application of a scientific discovery to a conceptual problem may help us see previously unnoticed connections.

[30] For example, the nature of space and time [31, pp. 481-496] and the metaphysics of computation [31, pp. 637-714].

[31] [16, p. 7].

[32] The reader may also wish to recall Daniel Dennett's remarks in the previous section.

L_1 All gold spheres are less than a mile in diameter.
L_2 All uranium spheres are less than a mile in diameter.

They are both true generalizations about our physical world, yet they differ in an important respect: (L_2), but not (L_1), is what we would regard (as scientists) as a true law of nature, something connected to the deep structure of the world and not the result of a sum of accidental facts. What is this difference grounded upon? Or - which is pretty the same - what is, generally speaking, a law of nature? Obviously, it is not the logical form of a sentence that makes it a law: (L_1) and (L_2) are both universally quantified statements involving physical features of our world. So what is it[33]?

Under a popular account, natural laws are the true generalizations contained in the best deductive system describing our world[34]. What does that mean? Take the long and somewhat confusing Book-Of-The-World, containing all the facts about the evolution of our Universe (say, things like the Big Bang, the Halley comet, the Solar System, exhaustively described in the appropriate scientific language). We can hope to systematize this vast knowledge of particular facts with several possible axiomatic systems, competing for the best trade-off between simplicity, strength, predictive power, etc.: some are conceptually simple, but not enough powerful to deliver the content of the Book-Of-The-World, some are powerful, but at the cost of many axioms and primitive notions[35]. Suppose the system with the best trade-off is S: then, the natural laws of the world are all and only the true generalizations contained in S's axioms or theorems. Going back to our earlier case, we can see how the account elegantly manages to solve the problem: since, presumably, S contains axioms concerning elements' stability and decay but not axioms concerning an upper bound to gold spheres' diameter, just (L2) can be derived from S - and that is why it is a law of nature.

Apart from purely conceptual considerations[36], a striking difficulty with such an account is that, textbook examples aside, no one has the slightest idea of how this system S may look like. Actually, the situation is even worse than that: no one has the slightest idea of how competing systems may look like, how the alleged trade-off may present itself, what strategies we may use to judge them. So, apart from purely conceptual considerations, we may still have the doubt that the theory is not (as we philosophers love to say) extensionally adequate, i.e., sometimes it fails to recognize a law in a lawful generalization and/or sometimes it says that an unlawful generalization is a law.

[33] See [5] for a general introduction.
[34] See [19], [22], drawing from ideas presented in [24].
[35] To avoid problems of language specificity, it is usually added the constraint that the predicates in the system stand for "natural properties" (which is another technical term that may benefit from a CA modeling). For the purpose of the present discussion, it is just important to note that systems are in some way normalized before the comparison.
[36] Such as the fact that, generally, the world is supposed to be the way it is *because* of the laws governing its evolution, not the other way around.

Digital universes such as *Life* may be very helpful: we can easily (so to speak) compile the relevant Book-Of-The-World and then ask ourselves: What is the best system that can generate this amazingly rich universe? Arguably, if we take as axioms the initial conditions of a *Life*'s run plus *Life*'s fundamental dynamics, we should obtain a simple and strong system capable of generating all the particular facts contained in the Book-Of-The-World. However, the system as it is will not generate any easily recognizable laws about, say, the movement of gliders under several conditions. Does this mean there are no laws concerning gliders? Even if we acknowledge their "derivative", emergent nature, this does not mean we cannot exploit solid regularities to understand their behavior: why should not these be bona fide laws? Again, if this seems a rather unimportant fact concerning a CA, just consider the analogous question about our world: are there laws in the special sciences (from biology, to psychology, to economics) or are physical laws the only laws of nature, while anything else is, at best, a bunch of ceteris paribus conditions, or plain accidental generalizations?[37]

Of course, there may well be theoretical merits in the view that *Life*'s laws are just the basic facts about its micro-dynamics: this is not the place to further discuss the issue, nor it is CA job to settle it. However, it is crucial to see how rephrasing a theory in terms of CA may be helpful in seeing clearly the bold consequences of some moves and the potential problems involved.

4.2 Developing and Debugging Philosophical Theories

As a second point in our Manifesto, we want to stress the importance of building a rigorous philosophy of CA. Consider again a classical *Life* dynamics:

t_1 t_2 t_4

In this case we have an *eater* (south-west) devouring a *glider* (traversing the region in the lattice from north-east). It may seem perfectly legitimate to say that this is an instance of *causation*, i.e. to assert that "The eater caused the glider to disappear". The corresponding counterfactual, in fact, seems true: hadn't the eater being present, the glider would have not disappeared (but would have reached the south-west corner in a few time-steps). According to many philosophers, this is good evidence that we have witnessed a causal interaction.

The concept of causation is one among the most crucial in understanding our world - think for example about the importance of causal processes within science. Of course, many philosophical accounts are available in the market to

[37] For a more extended discussion of this point, see [3].

explain this phenomenon: some[38] take the causal relation as somehow primitive, a brute, basic ingredient of our world. But here a rigorous ontology of the digital universe may be very helpful, highlighting that the above instance of causation is produced by the simple elements defining the *Life* universe - nothing more than that, no strange "causal powers", no magical relations (how do we know that? Well, we built the very thing)[39]. All the concepts we have when we describe *Life* dynamics are somehow sufficient to deterministically generate causal phenomena: investigating how exactly this is possible is a challenging task that goes beyond the scope of this contribution, yet this is a clear, paradigmatic example of how an important philosophical concept may benefit from a systematic investigation of CA philosophical features[40].

Closely related to the idea of a "CA philosophy" is the idea of "philosophical debugging". "Debugging" is a familiar activity to practitioners in Computer Science: in a nutshell, the idea is that, when an algorithm produces unexpected results, we test it step by step to understand where (and why) things start to go wrong. Consider a simple algorithm to calculate the factorial of some number, say 10:

```
for (i = 1; i =< 10; i++)
{
   int n = 1;
   n = n * i;
}
return n;
```

If we run the algorithm, we would get 10 as the result of 10!, which is clearly wrong. So where is the mistake? If we follow step by step the algorithm, we notice that each time in our loop the variable n (the one storing the partial result of the multiplications) is restarted to 1; obviously, we should declare it outside the loop to get the intended computation:

```
int n = 1;
for (i = 1; i =< 10; i++)
{
   n = n * i;
}
return n;
```

This is very nice, but, as philosophy is usually practiced, *none* of these beautiful practices of error-checking is possible: we start with some assumptions, we

[38] See for example the now very popular "manipulability account" defended in [32].

[39] Recall one of our earlier lesson: we should not declare the impossibility of X just because our imagination does not yet cover X.

[40] As a pioneering example of this attitude, see [2, Ch. 3].

somehow get weird outcomes, but there is not always an easy way to tell when we made a mistake in between.

CA are very precise mathematical objects: as such, they can support a very precise philosophical structure, one which will be easy enough to translate into your favorite programming language so that it may run on universe's simulations. In that way, we can use simulations as benchmark of our philosophical theory, by letting the computer compute all the consequences of our assumptions (by teaching it, so to speak, to see that reality through the lens of our theory). So, if the simulation produces an eater devouring a glider and our theory does not detect causation, we may stop the universe and check the reasoning as applied to the situation to understand what is wrong - and fix it.

Probably, not all debates in contemporary metaphysics may be adequately represented in a digital universe, but surely for many of them CA would provide a new, exciting way to develop and test original accounts: so much for those who think we cannot make experiments in philosophy.

4.3 Improving Ontology-Based A.I.

As a last area of interest, we mention briefly the potential benefit of a CA philosophy to computer science, in particular to Artificial Intelligence. While, of course, it is not known how exactly we can make a computer truly intelligent, we have a pretty good idea of why it is, sometime, truly stupid - as anyone who uses a smartphone to send messages well knows, the software sometimes suggests weird words since it does not understand the context of the word, nor its meaning (i.e. its relation to the actual world and other words). Information, *per se*, is quite useless: the key to knowledge representation is structured ("semantic") information[41].

One way to improve computers' understanding is to teach them concepts and relations, modeled in some formal language they can easily manipulate: that is why over recent years formal ontology has been increasingly useful to solve real business challenges[42]. Back to our philosophical debugging, we see that by teaching lessons in ontology to our laptop we are arguably making it capable of assigning (proto)-meaning to the words it uses to describe the digital universe. If this is true, any progress we can make in the small, yet qualitatively rich world of CA should be a potential contribution to the ongoing development of applied ontologies. Moreover, since CA phenomena are particularly abstract, a thorough understand of their philosophical significance may turn out to be of great interest to a wider audience, comprising practitioners in computer science, Artificial Intelligence and cognitive science.

If artificial minds are the next big thing[43], a CA-inspired philosophy may bring the field closer than ever to a great scientific revolution.

[41] See [10].
[42] See for example the works collected in [4].
[43] As argued for example in [18].

5 Conclusion

NKS – published ten years ago – was the exciting, groundbreaking exposition of a new scientific paradigm, whose scope and success it is still too early to judge. However, we argued at length that, independently of the specific claims made in that book, the conceptual framework put forward by Wolfram is a lively, engaging, challenging proposal to change (sometimes, radically) our habits of thought.

Unfortunately, many philosophical issues directly connected with NKS could not be discussed - first and foremost, a comparison between the philosophy of emergent properties and the science of emergent computation[44], which strikes me as the most promising area for a new synthesis of tools and ideas. Hopefully, the presented material would help philosophers understand the importance of a direct acquaintance with NKS and digital universes in general; indirectly, we also suggested that science would benefit from a better acquaintance with philosophy, but a solid defense of this latter point need to be the focus of another work.

Up until the beginning of the 20th century, philosophers were often at the very frontier of their time's science, something which became less and less common in the course of the century[45]. A New Kind of Philosophy may help them regain that place in this century[46].

References

[1] Berlekamp, E., Conway, J., Guy, R.: Winning Ways for Your Mathematical Plays, vol. II. Academic Press (1982)
[2] Berto, F., Rossi, G., Tagliabue, J.: The Mathematics of Models of Reference. College Publications (2010)
[3] Berto, F., Tagliabue, J.: Cellular Automata. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy (2012)
[4] Borgo, S., Lesmo, L. (eds.): Formal Ontologies Meet Industry. IOS Press (2008)
[5] Carroll, J.W.: Laws of Nature. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy (2010)
[6] Casati, R., Varzi, A.: Parts and Places. MIT Press (1999)
[7] Chalmers, D., Manley, D., Wasserman, R. (eds.): Metametaphysics: New Essays on the Foundations of Ontology. Oxford University Press (2009)
[8] Conee, E., Sider, T.: Riddles of Existence: A Guided Tour of Metaphysics. Clarendon Press (2007)

[44] See, for example, [9].

[45] While the facts are uncontroversial, it is hard to find a satisfying explanation. The increasing specialization of science and the rise of "analytic" philosophy – socially organized as a science – for sure contributed to the unfortunate separation.

[46] Many thanks to Francesco Berto (who, incidentally, taught me half the philosophy I know), Giulia Livio, Massimo Mastrangeli for comments, suggestions and technical help on earlier drafts of this work, and to Hector Zenil, for his kind assistance through all the editorial process.

[9] Crutchfield, J.P.: The Calculi of Emergence. Physica D 75, 11–54 (1994)
[10] Davies, J., Fensel, D., Van Harmelen, F. (eds.): Towards the Semantic Web. Ontology-Driven Knowledge Management. John Wiley & Sons (2003)
[11] Dennett, D.: Real Patterns. Journal of Philosophy 88, 27–51 (1991)
[12] Dennet, D.: Freedom Evolves. Viking Penguin (2003)
[13] Divers, J.: Possible Worlds. Routledge (2002)
[14] Fredkin, E.: A New Cosmogony. In: PhysComp 1992: Proceedings of the Workshop on Physics and Computation, pp. 116–121. IEEE Computer Society Press (1993)
[15] Hudson, H.: The Metaphysics of Hyperspace. Oxford University Press (2006)
[16] Ilachinski, A.: Cellular Automata. World Scientific Publishing (2001)
[17] Kim, J.: Supervenience and Mind. Cambridge University Press (1993)
[18] Kurzweil, R.: The Singularity is Near. Viking Penguin (2005)
[19] Lewis, D.: Counterfactuals. Harvard University Press (1973)
[20] Lewis, D.: Philosophical Papers, vol. 1. Oxford University Press (1983)
[21] Lewis, D.: On the Plurality of Worlds. Blackwell Publishers (1986)
[22] Lewis, D.: Humean Supervenience Debugged. Mind 103, 473–490 (1994)
[23] Piccinini, G.: Computation in Physical Systems. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy (2010)
[24] Ramsey, F.: Foundations. Routledge and Kegan Paul (1978)
[25] Sider, T.: Four-Dimensionalism: An Ontology of Persistence and Time. Oxford University Press (2005)
[26] Stanford, K.: Underdetermination of Scientific Theory. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy (2009)
[27] Van Inwagen, P.: An Essay on Free Will. Oxford University Press (1983)
[28] Von Neumann, J.: The General and Logical Theory of Automata. In: Cerebral Mechanisms in Behavior: The Hixon Symposium. John Wiley & Sons (1951)
[29] Wolfram, S.: Statistical Mechanics of Cellular Automata. Reviews of Modern Physics 55, 601–644 (1983)
[30] Wolfram, S.: Universality and Complexity in Cellular Automata. Physica D 10, 1–35 (1984)
[31] Wolfram, S.: A New Kind Of Science. Wolfram Media (2002)
[32] Woodward, J.: Making Things Happen. Oxford University Press (2003)

Chapter 23
Free Will and *A New Kind of Science*

Selmer Bringsjord

Department of Cognitive Science
Lally School of Management & Technology
Rensselaer Polytechnic Institute (RPI), USA
Selmer.Bringsjord@gmail.com

Abstract. What does Wolfram's new kind of science (nks_w) imply about the decidedly *non*-new topic of free will versus determinism? I answer this question herein. More specifically, I point out that Wolfram's nks_w-based position on free will is centered on the nature of physical laws, rather than formal logic; briefly rehearse the longstanding ontology of main positions on free will versus determinism in the context of physical laws; after a more detailed look at Wolfram's position, register agreement with him that in light of nks_w, the *belief* that free will is real and robust is to be expected, and is rational; but explain that nks_w provides no cogent rationalist basis for believing that we are *in fact* free. I conclude by pointing out that in light of the foregoing, if we are free, and can know that we are on the strength of what rationalists demand (e.g., proof), nks_w, while perhaps truly new, is truly incomplete. In short, I show that Wolfram, on free will, is epistemologically insightful, but metaphysically deficient.

1 Introduction

What does Wolfram's (2002) [11] new kind of science (nks_w) imply about the decidedly *non*-new topic of free will versus determinism? I answer this question herein. More specifically, I begin by briefly explaining that Wolfram's nks_w-based position on free will is centered in physics and agentless computation, not formal logic (§2); rapidly rehearse the immemorial, main positions on free will versus determinism in connection with physical law (§3); sketch out in section 4 what it takes to provide a rationalist basis for a position on free will (or on any subject, for that matter); after a more detailed look at Wolfram's position (§5), register agreement with him that in light of nks_w, the *belief* that free will is real and robust is to be expected, and is quite rational (§6); but explain that nks_w provides no cogent rationalist basis for believing that we are *in fact* free (§7). I conclude (in §8) by pointing out that in light of the foregoing, if we are free, and can know that we are on the strength of what rationalists demand (e.g., proof), nks_w, while perhaps truly new, is truly incomplete. In short, I show that Wolfram, on free will, is epistemologically right, but metaphysically deficient.

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 341–350.
DOI: 10.1007/978-3-642-35482-3_23

2 Wolfram on Free Will: A Physics-Based Orientation

Wolfram writes:

> Ever since antiquity it has been a great mystery how the universe can follow definite laws while we as humans still often manage to make decisions about how to act in ways that **seem** quite free of obvious laws. (Wolfram (2002) 750; bolded text here and in quotes hereafter due to me, to serve subsequent purposes)

Here Wolfram is pointing to a version of the free-will problem that involves physical laws and causation. There are other versions of the problem that are more abstract, and which steer clear of physical laws in favor of *a priori* reflection from the armchair (or its correlate in ancient Greece), and some of these were also discussed in the distant past that Wolfram points to. A seminal example is Aristotle's famous consideration of a future sea battle, given in his *De Interpretatione*, Chapter 9 (which can be found in [10]). Aristotle reflects on whether *tertium non datur* (TND) holds, and brings that issue into focus by asking whether TND holds with respect to

1. There will be a sea-battle tomorrow.
2. There will not be a sea-battle tomorrow.

Aristotle's reasoning, which we needn't assess, runs essentially as follows. If we assume for the sake of argument that 1. is true, then clearly the proposition expressed by this statement was true a week back, and a month back, and indeed 10,000 years back, *ad indefinitum*. But this is to say that it has always been the case that there will be a sea-battle tomorrow — and hence it immediately follows that all those human actions commonly associated with fighting a sea-battle (including the decision to launch an attack in the first place), commonly regarded to be up to us (= free), aren't. Exactly parallel reasoning can be carried out if the starting point is the assumption that 2. holds.

Aristotle's discussion falls under the topic of logic and the mind, definitely not, for instance, *physics* and the mind, and certainly nks_w falls into the latter domain. (Aristotle is read by some scholars as recommending rejection of TND in favor of a three-valued logic, an idea that certainly had legs: some contemporary extensional logics, e.g. the heterogeneous logic underlying Barwise and Etchemendy's [1] Hyperproof system, add to TRUE and FALSE such values as UNKNOWN.) After all, the central-to-nks_w doctrine of computational irreducibility ranges over the behavior, through time, of *physical* processes.

We turn now to consideration of the free-will problem from the standpoint not of armchair reflection and abstract logic, but physical, or natural, laws.

3 The Ontology of Free Will vs. Determinism

The classical expression of the "physics-relevant" "free will problem" is given by Chisholm's [7]. Encapsulated, the problem as portrayed by him is as follows; we shall call it 'The Dilemma.'

The Dilemma

(1) If determinism is true, then free will is an illusion; and yet on the other hand, (2) if *in*determinism is true, free will is an illusion. But since (3) either determinism or indeterminism is true, it follows that (4) free will is indeed chimerical.

To ease exposition, let's use '$\mathcal{D}$' to denote determinism, and '$\mathcal{I}$' to denote indeterminism. Indeterminism is understood to simply be the negation of determinism; that is, $\mathcal{I}$ if and only if not-$\mathcal{D}$. We thus see that (3) is an instance of a theorem in elementary deductive logic (viz., TND in either the propositional or predicate calculi; perhaps Boole never considered the sea-battle!), and is hence unassailable.

But we can be clearer. For Chisholm:[1]

$\mathcal{D}$: Every event is caused (by the conjunction of physical laws and prior and simultaneous events).

Hence, by elementary quantifier reasoning from the negation of determinism, we have:

$\mathcal{I}$: At least one event is uncaused.

There can be no doubt that the reasoning in The Dilemma is formally valid; indeed, an obvious symbolization in the propositional calculus, and a formal proof, effortlessly obtained, would quickly confirm that $\{(1), (2), (3)\}$ deductively entails (4). In addition, given even garden-variety accounts of event-causation, it's not hard to see that both (1) and (2) in The Dilemma are quite plausible.[2]

Take (1) first, and understand an event e to be caused just in case prior events, combined with the relevant laws of nature, logically necessitate that e occurs. Suppose now that e is an event that many would regard to be a strong candidate for something humans freely bring about; for example, *Smith's raising his hand to signal the launching of a battle*. Suppose that this event happens at t_n, and that $\mathcal{D}$ is true. Then, given events holding before t_n, at t_{n-1} let's say, and laws of nature that are of course completely beyond the control of Smith, it's logically necessary at t_{n-1} that Smith raise his hand to vote. Since this reasoning can be iterated indefinitely, we will reach a snapshot of the universe at a time $t^\star$ eons before Smith's existence which is such that, long into the future from that timepoint, Smith absolutely must send the signal he does at t_n. This fact is

[1] And for others also seeking a rigorous statement of the free will problem; e.g., for Zimmerman's [12] and Bringsjord's [2].

[2] In a Newtonian framework, e.g., (1) and (2) are provable on axiomatizations of Newtonian mechanics. It's beyond scope for the present chapter to discuss the status of such formalizations, or formalizations of (1) and (2) in, say, quantum-mechanical frameworks. Both (1) and (2) do seem quite plausible on Wolfram's physico-computational framework.

inconsistent with the proposition that it's up to Smith as to whether he raises his hand or not, under any reasonable understanding of up-to-us-ness.

Now, the ontology of the immemorial free-will debate is derived from stances on the truth or falsity of (1), (2), $\mathcal{D}$, and $\mathcal{I}$, and runs as follows:

The Ontology of the Free-Will Debate

Incompatibilism (1): $\mathcal{D}$ is compatible with our having free will; i.e., the reasoning given above in favor of (1) is regarded to be compelling.
Compatibilism not-(1): $\mathcal{D}$ is compatible with our having free will.
Hard Determinism : $\mathcal{D}$ conjoined with incompatibilism.
Soft Determinism : $\mathcal{D}$ conjoined with compatibilism.

Not that it matters for the present essay, but Chisholm was a libertarian, as am I. A defense of libertarianism is provided in [2]. In general, it's safe to say that incompatibilism is aligned with the common-sense and ubiquitous laic notion, unabashedly affirmed herein, that the concept of up-to-us-ness is at the heart of what it means to be free. If free will consists in our ability to perform actions that are entirely up to us, then compatibilism, which must accept that free will requires only that we do what we want to do, doesn't seem to be tenable. This is so because if our desires were pre-programmed into us by some other agent, our acting in accord with our desires wouldn't be up to us, but rather up to that other agent.

4 Rationalism Encapsulated

We turn now again to Chisholm, who has provided a discrete continuum of epistemic "strength" [8]. Chisholm's spectrum of the strength of a proposition for a rational human mind is a nine-point one, and ranges from 'certainly false' at the negative end, to 'certain' at the positive end. At the halfway point are propositions said to be *counterbalanced*. There are then four positive strength factors working up from there: first *probable*, then *beyond reasonable doubt*, then *evident*, and finally the aforementioned *certain*. Certain propositions include the indubitable truths of formal logic (e.g., *modus ponens*, $0 \neq 1$, Peano Arithmetic, etc.), and presumably "Cartesian" truths such as "I exist," and "It seems to me that I'm sad." What kind of thing is evident? For the most part, the evident would be populated by those propositions we affirm on the strength of direct sense perception. For example, that there is a computer screen in front of me when I'm typing out a sentence such as the present one is evident. This proposition isn't certain: you might be hallucinating, after all; but it's — as we might say — *close* to certain. You wouldn't want to say, for example, while spying a coffee cup in front of you, in perfect health and having not ingested recently any mind-altering drugs ..., that the proposition that there's a cup in front of you is merely beyond reasonable doubt: you want to say, instead, that you are well within your epistemic "rights" in holding that it's *extremely* likely that there's a cup before you. This, again, is the category of the evident.

But moving down another Chisholmian notch in strength, we do in fact hit *beyond reasonable doubt* — which of course famously coincides roughly with what it takes in certain legal systems (e.g., that of the U.S.) to legally convict someone of murder. That is, to convict someone of this kind of crime, the evidence must make some such proposition as *Jones is guilty* beyond reasonable doubt. Finally, note that to convict on this standard, it's not sufficient to know that it's merely *probable* that Jones did it. Some proposition P being probable is the last notch before we reach *counterbalanced*, which as you've no doubt anticipated entails that a purely rational agent wouldn't bet in favor of P, and wouldn't bet against it. A perfectly rational agent who is agnostic about some proposition P would regard P to be counterbalanced.[3]

Armed with Chisholm's spectrum, we can now offer a tolerably clear encapsulation of the rationalist standard for belief in positions on free will:

> **Rationalism** The view that belief in weighty, philosophical proposition P must be supported by deductive proofs or arguments, where the inferences in this reasoning are each formally valid, and the premises are at least probable.

This doctrine can be partitioned into at least a *strong*, *moderate*, and *weak* subforms. *Strong rationalism* is the view (and as it happens, *my* view) that any human person believing some weighty, philosophical P ought to have on hand at least one outright proof of P; that is, have on hand a formally valid chain of deductive inference originating from premises that are each certain.[4] The doctrine of *moderate rationalism* holds that if Jones abides by this doctrine and believes P, then Jones must have on hand at least one formally valid argument for P whose premises $P_1, P_2, \ldots, P_n$ are each at least evident. And following suit we can say that *weak rationalism* requires only that the premises involved in deductive reasoning for the P in question are at least probable. Readers will no doubt get the driving idea from the foregoing; the story would continue on, all the way through an exceedingly fine-grained ontology of rationalism.[5]

[3] What about the "negative" side of Chisholm's continuum? Since neither the empiricist nor the rationalist, if abiding by their respective programs for belief fixation, would assent to propositions on the negative side of *counterbalanced*, we have no need here to explore this epistemic terrain. Interested readers can consult [8], and a recent "AI-ish" exploitation of Chisholm's framework in [5].

[4] Some readers will inevitably ask: "Is there any such thing?!" I'm well aware of the fact that even some axioms in some axiomatic set theories are controversial, and hence perhaps not certain. (Even the power-set axiom in ZFC has its detractors, e.g.) Nonetheless, whatever one can deduce in deductively valid fashion from, say, $1 = 1$, would be certain, and one would be well-advised to believe such a consequence. For instance, $1 = 1 \vee Q$, for any proposition Q, would be an acceptable disjunction for even a strong rationalist to believe.

[5] For example, we could distinguish between the strength of inferential links in the argument for P.

5 Wolfram on Free Will: A More Careful Look

It's now time to look in more detail at Wolfram's treatment of free will in *A New Kind of Science*. To do so, let's pick up right after the short quote from this book presented earlier in section 2. We read:

> [F]rom the discoveries in this book it finally now seems possible to give an explanation for [how the universe can follow definite laws while we as humans still often manage to make decisions ... in ways that **seem** quite free of obvious laws]. And the key, I believe, is the phenomenon of computational irreducibility. For what this phenomenon implies is that even though a system may follow definite underlying laws its overall behavior can still have aspects that fundamentally cannot be described by reasonable laws. For if the evolution of the system corresponds to an irreducible computation then this means that the only way to work out how the system will behave is essentially to perform this computation—with the result that there can fundamentally be no laws that allow one to work out the behavior more directly. And it is this, I believe, that is the ultimate origin of the **apparent** freedom of human will. For even though all the components of our brains presumably follow definite laws, I strongly suspect that their overall behavior corresponds to an irreducible computation whose outcome can never in effect be found by reasonable laws. (Wolfram (2002) 750; bolded text due to me, to serve subsequent purposes)

We can quickly erect a modicum of logico-computational machinery to demonstrate that Wolfram here is entirely correct.

Consider two human persons, Alice and Bob (M_b). We'll assume that M_b is a deterministic Turing machine (TM) based on the binary alphabet $\{0, 1\}$ and having two one-way tapes t_1 and t_2, one read/write head operating on each.[6] Tape t_1 enables perception for M_b: a symbol appearing on t_1, and read, indicates that that symbol is perceived by M_b. The other tape, t_2, is used for "internal thinking" on the part of Bob. To further fix our context, we assume that Bob's life unfolds in discrete time steps

$$t_1, t_2, t_3, \ldots$$

into the future, in accordance with the following pattern: Bob thinks for four steps, then perceives (either 0 or 1, and the head on t_1 then moves one square to the right, and awaits the next datum from the external world) in one time step, and then four in a row for thinking, and so on *ad indefinitum*. We write 'C_i^M' to

[6] Wolfram is generally fond of depicting cellular automata rather than TMs (though he does spend appreciable time on Register machines), and indeed he gives a fascinating example of an "unpredictable" one in his principal discussion of free will: see the graphic on p. 750 (2002). But no loss of generality or insight results from restricting our attention to TMs. In addition, while I claim to have proved that human persons can't possibly be TMs, or indeed anything of the sort (e.g., see [3, 4]), for the sake of exposition and argument we here ignore such reasoning, which makes the identification of Bob with M_b palatable.

refer to a *configuration* of TM M.[7] Consider an equation schema **E** designed to yield a configuration of Bob for any timepoint given as input; that is, consider:

$$f(t_k) = C_k,$$

where the function f provides the "meat" of this equation, and is itself a Turing-computable function.[8]

We are now in position to see that Wolfram, in the quote immediately above, is right.

6 Wolfram Is Correct — Epistemologically

Suppose that we are interested in whether Alice believes Bob to have free will. Not unreasonably, we shall stipulate the following epistemic principle $\mathcal{E}$: a sufficient condition for such a belief on the part of x about TM y is that despite x's having complete knowledge about the transition rules that determine the state of y at t_k given the state of y at t_{k-1}, the "overall behavior" of y cannot be anticipated by x. More precisely, we stipulate that despite knowledge of transition rules, x does not, indeed *cannot*, predict, on the strength of an equation of the form of **E**, the configuration that y will be in for some future timepoint. Next, we shall agree with Wolfram that if x is in this position of ignorance about the future states of y, then x will ascribe free will to y; that is, x will believe that y has free will.

We can now prove that Wolfram is right with respect to Bob, as long as we assume that Bob, *qua* TM, is for instance as complex as the impenetrable, unpredictable 6-state machines which have never been predictable within the confines of the Busy-Beaver Problem.[9] Needless to say, Bob's mind is unquestionably more complex that such TMs! The proof is trivial once we realize that the Wolframian setup we have established implies that the following proposition is now an easy lemma: $\neg\exists f f(t_k) = C_k^{M_b}$,

From this lemma it follows directly by *modus ponens* on $\mathcal{E}$ that Alice believes that Bob has free will. Since we have here fleshed out computational irreducibility with respect to Bob, Wolfram's reasoning is certified. Moreover, his reasoning, given the account supplied above, is without question rationalist in nature — since the inferences are deductively valid, and all premises appear to be at least evident. In particular, given the framework set out in section 4, we can declare that Wolfram has provided a case for the belief in free will that accords with the standards of moderate rationalism.

[7] The concept of a *snapshot* or *configuration* is standard in presentations of TMs. E.g., see [9].

[8] I have shown that human mentation includes information-processing more powerful than what a TM can reach [6], but I leave this aside in the present essay.

[9] Wolfram provides an elegant, succinct description of the Busy-Beaver Problem: ((2002) 889 & 1144).

7 Wolfram Is Wrong — Metaphysically

But there is a hitch here, a very serious one. In general, the hitch is that it doesn't follow from the fact that x believes some proposition that that proposition is true. Some humans still believe that Earth is flat, after all. But how does this specifically relate to the case at hand? If you look back to all the bolded parts of the quotes from Wolfram's *ANKS*, you'll see, clear as day, that Wolfram has proffered only an explanation for why humans, in general, *believe* that they have free will. For example, we earlier saw this:

> And it is this, I believe, that is the ultimate origin of the **apparent** freedom of human will. For even though all the components of our brains presumably follow definite laws, I strongly suspect that their overall behavior corresponds to an irreducible computation whose outcome can never in effect be found by reasonable laws. (Wolfram (2002) 750; again, bolded text due to me, to serve present purposes)

This is just one example from many that I've pinpointed via bolded text, but the situation, especially given the many other bolded words, should be clear as day. If someone's will is *apparently* free, it hardly follows that that will is *in fact* free. Nowhere in *ANKS* does Wolfram even intimate that he maintains that our decisions are in fact free.

But what we are ultimately concerned with is whether, *in fact*, at least some of our decisions are truly up to us. On this issue, which is the real one, Wolfram is deafeningly silent. Moreover, it would seem to be implausible that free will, or up-to-usness, is in fact in place in the universe as conceptualized under nks_w. Why?

Well, think back to Alice and Bob. But we are now concerned not with whether Alice, under reasonable physico-computational and epistemic assumptions, believes that Bob $= M_b$ has free will; rather, we are interested in whether or not Bob is *in fact* free. We have only two general factors that are relevant to this question. And neither factor is of help to Wolfram on the question before us.

To see this, consider first the first factor: the one that served to support Alice's belief that Bob is free: namely, that relevant instantiations of equation **E** for Bob's future behavior are simply unavailable. But the unavailability of equations of this form in no way rationalistically entails that Bob in fact is free. For just because we can't predict, for some future timepoint, what state Bob will be in at it, doesn't ensure that the state he is in at this timepoint is due to the free operation of his own will. You may be unable to predict what general configuration a puppet will be in to start the next act of a puppet show (because, among other reasons, you are unfamiliar with the relevant choreography), but it hardly follows from this inability that a puppet has free will.

And now what is the second factor? It's that by all accounts the Wolframian world-view seems to be inconsistent with up-to-us-ness. Notice that I don't *assert* this inconsistency; I claim only that there *seems* to be outright inconsistency. The reason for my claim can be seen by turning yet again to the Alice-Bob scenario;

specifically, to the fact that while Alice can't predict what Bob's configuration will be in at an arbitrary future timepoint, Alice *can* predict the configuration Bob will be in at t_n if she knows the configuration he's in at t_{n-1} (since, as we legislated, she knows Bob's transition rules). In short, nothing seems to be up to Bob whatsoever: the state he is in at any given moment appears to be entirely necessitated by the co-operation of transition rules (which are of course directly analogous to causation in The Dilemma) and the input to them (i.e., the configuration at the moment immediately preceding). In sum, The Dilemma could be recast within the Wolframian pan-computational framework, and would thereby lose none of its original force; in fact it would *gain* in force.

8 Conclusion

To sum up, the situation is clear: A rational person (i.e., for us herein, a rationalist), having open-mindedly studied *A New Kind of Science*, and assumed to have an understanding of the longstanding ontology of the free-will debate, including specifically The Dilemma, will not be enlightened as to what the solution to that dilemma is. However, on the bright side, Wolfram can be credited for his commendable scholarship, for the elegance of his discussion of free will, and for a compelling argument in support of the proposition that if our world is indeed computationally irreducible, human persons, when assumed to have sufficiently refined cognitive capacities for perception and reasoning, will indeed believe themselves to be free. The downside is that if we are free, and can know that we are on the strength of supporting argumentation and/or proof of the sort required by moderate rationalism, it follows that nks_w is at best incomplete, and at worst — if in fact the computationalism in nks_w rules out up-to-us-ness — incorrect.

References

[1] Barwise, J., Etchemendy, J.: Hyperproof. CSLI, Stanford (1994)
[2] Bringsjord, S.: Free Will. In: What Robots Can and Can't Be, pp. 266–327. Kluwer, Dordrecht (1992)
[3] Bringsjord, S.: What Robots Can and Can't Be. Kluwer, Dordrecht (1992)
[4] Bringsjord, S., Arkoudas, K.: The modal argument for hypercomputing minds. Theoretical Computer Science 317, 167–190 (2004)
[5] Bringsjord, S., Taylor, J., Shilliday, A., Clark, M., Arkoudas, K.: Slate: An Argument- Centered Intelligent Assistant to Human Reasoners. In: Grasso, F., Green, N., Kibble, R., Reed, C. (eds.) Proceedings of the 8th. International Workshop on Computational Models of Natural Argument, (CMNA 2008), Patras, Greece, pp. 1–10 (2008), `http://kryten.mm.rpi.edu/Bringsjord_etal_Slate_cmna_crc_061708.pdf`
[6] Bringsjord, S., Zenzen, M.: Superminds: People Harness Hypercomputation, and More. Kluwer Academic Publishers, Dordrecht (2003)

[7] Chisholm, R.: Freedom and Action. In: Lehrer, K. (ed.) Freedom and Determinism, pp. 11–44. Random House, New York (1964)
[8] Chisholm, R.: Theory of Knowledge, 2nd edn. Prentice-Hall, Englewood Cliffs (1977)
[9] Lewis, H., Papadimitriou, C.: Elements of the Theory of Computation. Prentice Hall (1981)
[10] McKeon, R. (ed.): The Basic Works of Aristotle. Random House, New York (1941)
[11] Wolfram, S.: A New Kind of Science. Wolfram Media, Champaign (2002)
[12] Zimmerman, M.: An Essay on Human Action. P. Lang, New York (1984)

Erratum: Symmetry and Complexity of Cellular Automata: Towards an Analytical Theory of Dynamical System

Klaus Mainzer and Carl von Linde-Akademie

Technische Universität München, München, Germany
mainzer@cvl-a.tum.de

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 47–65.
DOI: 10.1007/978-3-642-35482-3_5

DOI 10.1007/978-3-642-35482-3_24

In the original version, the second author name has to be deleted i.e. “Carl von Linde-Akademie”.

The affiliation part was wrongly captured as second author name. It should be read as:

Carl von Linde-Akademie,
Technische Universität München, München, Germany
mainzer@cvl-a.tum.de

The original online version for this chapter can be found at
http://dx.doi.org/10.1007/978-3-642-35482-3_5

Afterword: Is All Computation?

The main goal of NKS is to understand nature through computation, and, indeed, mining the computational universe has produced remarkable results.

There are a number of indisputable facts about NKS. The book and its companions (its website `http://www.wolframscience.com/` which includes the free online and the iPad versions) have stimulated and motivated much research and debate not only in computability and complexity, but also in mathematics, natural and social sciences and arts. More than 10,000 publications have cited NKS, cf. Wolfram's blog[1]. Most notable developments are contained in the books [10, 16, 14, 20] and the articles in this present book. In science—where new results supersede previous ones—this is arguably one of the most important symptoms of a significant impact.

Many ideas and much material of NKS are still as cogent and relevant today as they were ten years ago; see, for example, the experimental trend in mathematics discussed in [6].

The book has attracted both vivid criticism and praise. Many 2002 reviews were written in a quite emotional tone, mixing opinions about the scientific merit of the ideas, results and methodology of the book, and S. Wolfram himself; see, for example, the archives [11, 21]. Little relevant issues like the self-publication of NKS and the apparent isolation of the author from the scientific community have been discussed. Of course, there have been important exceptions: one is Aaronson's review [1]. Later, more scientific approaches prevailed, e.g. Neske's analysis of the Principle of Computational Equivalence, [17].

Last but not least, NKS motivated the design of two unique computation-powered entities: Wolfram|Alpha[2], the engine which can answer questions, do mathematics and statistics, compute scientific data and facts[3], and the computable document format (cdf), an embedded knowledge container document[4] which injects interactivity into the standard flat and lifeless document (processed by the free Wolfram CDF player[5], cdf documents promise to change the way research, business, education, business and technical development are done).

Wolfram's 'Principle of Computational Equivalence' (PCE) states that almost all computational systems that are not obviously simple are equivalent to a universal Turing machine. As the Church-Turing Thesis, PCE cannot be proved,

[1] `http://blog.stephenwolfram.com/2012/05/its-been-10-years-whats-happened-with-a-new-kind-of-science/`.

[2] `http://www.wolframalpha.com`

[3] New York Times reports that Apple's voice-controlled personal assistant Siri accounts for about 25% of the traffic handled by Wolfram|Alpha, `http://www.nytimes.com/2012/02/07/technology/wolfram-a-search-engine-finds-answers-within-itself.html?_r=2`.

[4] `http://www.wolfram.com/cdf`

[5] `http://www.wolfram.com/cdf-player`

but can be possibly disproved (see also [5]). One way to collect evidence in favour of PCE is to find more examples of *s*imple programs that exhibit complex behaviour.

The last statement has different meaning depending upon the way we interpret its terms: simple program and complex behaviour. 'Complex behaviour' can refer to computational power: a program is complex if it is computational universal, that is, it can simulate any other program. An example of universal simple program is the "Rule 110" cellular automaton defined by the following simple binary recurrence

$$p_{i+1,j} = p_{i,j} + p_{i,j+1} - (1 + p_{i,j-1})\, p_{i,j}\, p_{i,j+1}, i \geq 0,$$

with an initial condition at $i = 0$, cf. [12]. If $p_{i,j} = 1$ when cell (i, j) is coloured black, and $p_{i,j} = 0$ when it is white, then the 'Rule 110' cellular automaton can generate complicated patterns even when run on obviously simple initial conditions, such as a single black cell.

The quest for the smallest universal Turing machine has stimulated generalisations of Turing original notion: various types of weak universality have been proposed. Using non-periodic initial configurations (having the complexity of a context free language) Smith [18] proved the weak universality of Wolfram's 2-state 3-symbol Turing machine (the universality of 'Rule 110' implies the universality of a 2-state, 5-symbol Turing machine, a weaker result). Margenstern [15] has constructed a family of weakly universal two-state cellular automata in the hyperbolic plane.

'Simple programs with complex behaviour' can be illustrated by characterisations in terms of small-size programs of complicated mathematical open problems. For example, the Riemann Hypothesis and the P = NP problem, two of the seven open problems in the list of the Millennium Prize Problems stated by the Clay Mathematics Institute in 2000, can be fully described by programs with less than 3 kbits and 7 kbits, respectively (see [8]).

According to [7] three other definitions of simplicity of universality are possible. A prefix-free universal machine U is

- *simple for Peano Arithmetic* (PA) if PA can prove (given U's full description) that "U is prefix-free and universal": some but not all prefix-free universal machines are simple for PA;
- *n–simple for Zermelo–Fraenkel set theory with the axiom of choice* (ZFC) if ZFC can compute n digits and no more of the binary expansion of its halting probability: for every $n \geq 1$ there exists a prefix-free universal machine which is *n–simple for* ZFC and there exists a universal machine which is not 1–simple for ZFC;
- PA–*simple for randomness* if PA can prove that U's halting probability is algorithmically random: every simple for PA prefix-free universal machine is PA–simple for randomness, but the converse implication is an open problem.

Can PCE be disproved? Any argument refuting the Church-Turing Thesis will disprove PCE. But are there reasons to doubt the validity of the Church-Turing Thesis? According to NKS, p. 1125, Note (d):

> ... starting in the 1950s a few physicists, notably Richard Feynman, asked about fundamental comparisons between computational and physical processes. But it was not until the 1980s–perhaps particularly following some of my work–that it began to be more widely realized that Church's Thesis should best be considered a statement about nature and about the kinds of computations that can be done in our universe. The validity of Church's Thesis has long been taken more or less for granted by computer scientists, but among physicists there are still nagging doubts, mostly revolving around the perfect continua assumed in space and quantum mechanics in the traditional formalism of theoretical physics (see page 730).

During the same time Turing [19] noted that

> 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".

A Turing machine augmented with a random oracle can trespass Turing's barrier, i.e. it can compute an incomputable function if the oracle is incomputable, [2]. These machines have been studied theoretically for a long time (see [13]), but are there ways to actually build any of them? Quantum randomness certified by Kochen-Specker theorem is *s*trongly incomputable (see [9, 4]), so a Turing machine working with a finite but unbounded supply of quantum random bits generated in this way trespasses Turing's barrier. Such a computational system may disprove both the Church-Turing Thesis and PCE, hence the question is: Can such a computational system be built? Blueprints for such systems have been designed in [3, 4], but more is necessary to reach a conclusive answer.

Would the refutation of the Church-Turing Thesis and/or PCE falsify the more philosophical statement: "all is computation"? Probably not.

Acknowledgement. We thank A. A. Abbott and H. Zenil for his comments which improved the presentation.

References

[1] Aaronson, S.: Book Review: 'A New Kind of Science', p. 11 (2002) arXiv:quant-ph/0206089v2

[2] Abbott, A.A., Calude, C.S., Svozil, K.: On demons and oracles. Asia Pacific Mathematics Newsletter 2(1), 9–15 (2012)

[3] Abbott, A.A., Calude, C.S., Svozil, K.: A quantum random number generator certified by value indefiniteness. Mathematical Structures in Computer Science (accepted, 2012)
[4] Abbott, A.A., Calude, C.S., Conder, J., Svozil, K.: Kochen-Specker Theorem Revisited and Strong Incomputability of Quantum Randomness (2012) (in preparation)
[5] Arrighi, P., Dowek, G.: The Principle of a Finite Density of Information. In: Zenil, H. (ed.) Irreducibility and Computational Equivalence. ECC, vol. 2, pp. 127–134. Springer, Heidelberg (2013)
[6] Bailey, D.H.: A New Kind of Science: Ten Years Later. In: Zenil, H. (ed.) Irreducibility and Computational Equivalence. ECC, vol. 2, pp. 67–76. Springer, Heidelberg (2013), http://crd-legacy.lbl.gov/~dhbailey/dhbpapers/dhb-wolfram-2012.pdf
[7] Calude, C.S.: Simplicity via provability for universal prefix-free Turing machines. Theoretical Comput. Sci. 412, 178–182 (2010)
[8] Calude, C.S., Calude, E.: The Complexity of Mathematical Problems: An Overview of Results and Open Problems. CDMTCS Research Report 410, 12 (2011)
[9] Calude, C.S., Svozil, K.: Quantum randomness and value indefiniteness. Advanced Science Letters 1, 165–168 (2008)
[10] Chua, L.: A Nonlinear Dynamics Perspective of Wolfram's New Kind of Science, vol. 1 & 2. World Scientific, Singapore (2007)
[11] Clark, W.E.: A Collection of Reviews of ANKOS and Links to Related Work, http://shell.cas.usf.edu/~wclark/ANKOS_reviews.html
[12] Cook, M.: Constructive methods for one-dimensional cellular automata. Constructive Cellular Automata Theory 1, 1–40 (1998), http://psoup.math.wisc.edu/CA98/index.html, Published in NKS, p. 1115, and in Universality in elementary cellular automata. Complex Systems 15(1), 1–40 (2004)
[13] Cooper, S.B.: Computability Theory. Chapman & Hall/CRC, London (2004)
[14] Downey, A.: Think Complexity: Complexity Science and Computational Modeling. O'Reilly, Sebastopol (2012)
[15] Margenstern, M.: A family of weakly universal cellular automata in the hyperbolic plane with two states, CoRR, abs/1202.1709 (2012), http://arxiv.org/abs/1202.1709
[16] McIntosh, H.: One Dimensional Cellular Automata. Luniver Press (2009)
[17] Neske, G.: Whatever happened to Stephen Wolfram? A New Kind of Science revisited. Skeptic 14(3), 52–56 (2008), https://files.nyu.edu/gtn206/public/Neske_NKSEssay_2008.pdf
[18] Smith, A.: http://www.wolframscience.com/prizes/tm23/TM23Proof.pdf
[19] Turing, A.M.: Computing machinery and intelligence. Mind 59, 433–460 (1950)
[20] Zenil, H. (ed.): A Computable Universe. Understanding & Exploring Nature as Computation. World Scientific, Singapore (2012)
[21] Archived Media Coverage of NKS, http://www.wolframscience.com/coverage.html

—Cristian S. Calude
The University of Auckland, New Zealand

Index