VI. Hyperminds

Dexter Jettster: . . . Those analysis droids only focus on symbols. Huh! I should think that you Jedi would have more respect for the difference between knowledge and. . . wisdom!

Obi-Wan Kenobi: Well, if droids could think there'd be none of us here, would there?

From a dialog in the movie *Star Wars Episode II: The Attack of the Clones*, produced and directed by George Lucas.

Is the mind just a computing device, or is it something more? This and similar questions have prompted a number of thinkers and researchers to propose various theories that aim to falsify the general belief that the mind is actually a computing device. In this direction, one may argue that computers are actually "mental prostheses or orthoses, not stand-alone minds" [198]. Indeed, it is not an overstatement to say that computers dully execute commands and deliver results that only a conscious mind can interpret. Obviously, it is not an exaggeration to say that this *naive* remark forms a basis for more rigorous arguments against computationalism, such as the Chinese room argument. Interestingly enough, cognitive science (i.e., "the interdisciplinary study of mind and intelligence, embracing philosophy, psychology, artificial intelligence, neuroscience, linguistics, and anthropology" [201]) began by assuming that the human mind is indeed a Turing machine. Because of its impact on modern thinking, any voice against the kernel of cognitive science is faced with great skepticism.

This chapter is a short presentation of various attacks against computationalism. First, there is a presentation of various arguments based on results from mathematical logic, such as Gödel's incompleteness theorems. Then we present a number of purely philosophical arguments against the idea that the mind is a Turing machine. In addition, there is a more elaborate discussion of the Chinese room argument and related issues. Next,

there is a discussion of the mind from a neurobiological point of view, and we conclude with a discussion of the cognitive aspects of the human mind.

6.1 Mathematics and the Mind

6.1.1 The Pure Gödelian Argument

John Lucas was probably the first to use Gödel's famous incompleteness results to attack computationalism.[1] The essence of his argument [116] is that since *machines* are "concrete instantiations of a formal system," they should not be able to prove a particular proposition that a mind can clearly see to hold true. Thus, minds are not machines. It is rather interesting to note that Paul Benacerraf examined Lucas's argument and concluded that: "If I am a Turing machine, then I am barred by my very nature from obeying Socrates profound philosophic injunction: KNOW THYSELF" [13]. As a side effect, Benacerraf concluded [13] that

> Psychology as we know it is therefore impossible. For, if we are not at best Turing machines, then it is impossible, and if we are, then there are certain things we cannot know about ourselves or any others with the same output as ourselves. I won't take sides.

Lucas's argument was expounded by Roger Penrose in *The Emperor's New Mind* [150] and its sequel *Shadows of the Mind* [151]. In accordance with Lucas, Penrose believes that minds are not machines, and in addition, he believes that computers cannot simulate brain processes. The summary of Penrose's argument that follows is based on Searle's summary that appears in Chapter 4 of [175]:

(i) Classically, the halting problem, which is a specific version of Gödel's incompleteness theorem, cannot be solved. Thus, this can be used to show that our conscious behavior is not computable. In particular, Penrose considers some nonstopping computational processes that cannot be shown to be nonstopping by purely computational methods, but at the same time we can see that the processes are nonstopping.

(ii) The operation of a *neuron* can be simulated by computer. Thus, the behavior of neurons is computable. This implies that neurons cannot be used to explain consciousness, since consciousness has noncomputable features.

1. Actually, as Lucas admits in the first page of [116], the purpose of his work was to attack *mechanism* (i.e., the doctrine that all natural phenomena are explicable by material causes and mechanical principles), which one might say is a forerunner of computationalism.

(iii) A theory of consciousness should be based on noncomputable phenomena that might take place at the level of microtubules in neurons. But in order to understand these phenomena we need a new physics.

Although many thinkers and researchers are convinced that a computational model of the mind is extremely implausible, still they are not convinced by Penrose's argument. For instance, Solomon Feferman [61] shows that there are flaws in Penrose's argument. Interestingly enough, Penrose was "happy to agree with all the technical criticisms and corrections that Feferman refers to in his section discussing" his "treatment of the logical facts" [153]. Feferman also points out that it is misleading to assume that the equivalence between formal systems and Turing machines can be used to derive a general methodology for proving theorems. After all, mathematicians arrive at proofs "through a marvellous combination of heuristic reasoning, insight and inspiration (building, of course, on prior knowledge and experience)" [61]. Another attack on the Gödelian argument has been put forth by Selmer Bringsjord and Michael Zenzen [26].

From the discussion above, it should be clear that Penrose not only rejects "strong AI," but also "weak AI." Quite naturally, he believes that no computer program can have the qualities of *awareness* and *understanding*. Obviously, it is one thing to believe that no computer program can possess these qualities, and another to believe that no machine can possess these qualities. Humans, which may be viewed as biological machines, have both qualities and thus trivially refute the idea that no machine can have awareness and understanding. However, John McCarthy, in his attack against Penrose's ideas [124], supports the idea that computer programs can have awareness and understanding. More specifically, he advocates [123] that interrupts,[2] which are supported by many popular programming languages, might form the basis for the implementation of self-awareness in computer programs. Clearly, we have a situation in which a hardware device sends some unintelligent message that is processed by a computer program, which is dully executed by a CPU. In addition, a conscious biological machine (e.g., Peter) assigns meaning to all of these, and just because of this assignment, the computer program might be self-aware! Although I believe that one day there will be conscious machines, built atop a very different yet to be discovered machine architecture, the current machine architecture is not, at best, a promising direction. And of course, this may explain why space probes landing on other planets rarely survive more than their expected "life" span.

The debate over the Gödelian argument is quite active and recently Michael Redhead [164] presented a simplified version of that argument, presented in the framework of the system Q (a form of arithmetic), which

2. Roughly, an interrupt is a signal created and sent to the CPU, which is caused by some action taken by a hardware device. For example, pressing certain key combinations can cause hardware interrupts.

has the following axioms (recall that $S(x) = x + 1$):

$$\begin{gathered}(y = 0) \vee (\exists x)\Big(y = S(x)\Big),\\ 0 \neq S(x),\\ S(x) = S(y) \Rightarrow x = y,\\ x + 0 = x,\\ x + S(y) = S(x + y),\\ x \cdot 0 = 0,\\ x \cdot S(y) = (x \cdot y) + x.\end{gathered}$$

Observe that in this set of axioms there is no provision for proof by mathematical induction. Now consider the following statement:

$$\text{for all } n, m \in \mathbb{N} \text{ there exists a proof that } m \times n = n \times m. \qquad \text{(A)}$$

First, notice that we cannot switch the order of the quantifiers to get

$$\text{there exists a proof that for all } n, m \in \mathbb{N} : m \times n = n \times m. \qquad \text{(B)}$$

The reason for this deficiency is the lack of an induction axiom. Notice also that the proof of (A) depends on the specific numbers n and m that are chosen, and the length of the proof will increase as the numbers get bigger and bigger. But it should be clear that the length of the proof is finite. On the other hand, the proof of (B) is by no means finite, since it must cover every possible case. It is interesting that we can argue that

$$\text{for all } n, m \in \mathbb{N} : m \times n = n \times m. \qquad \text{(C)}$$

is actually true in system Q. Since the axioms and the theorems of system Q are *analytically* true (i.e., they express defining properties of the natural numbers), we may replace (A) by

$$\text{for all } n, m \in \mathbb{N} \text{ it is true that } m \times n = n \times m. \qquad \text{(D)}$$

But (D) is strictly equivalent to

$$\text{it is true that for all } n, m \in \mathbb{N} : m \times n = n \times m. \qquad \text{(E)}$$

The essence of this statement is that the commutative law of multiplication is true. Notice that truth commutes with the universal quantifier, whereas provability does not. This argument can also be viewed as a first step at showing that system Q is incomplete, since we have found a sentence that we agree is true but not provable in Q. Of course, this is not the only such sentence. For example, other such sentences are the associative laws of addition and multiplication. Since Q is not a recursive theory, one may

say that human mathematical reasoning is stronger than any nonrecursive theory. Figuratively speaking, human mathematical reasoning beats Turing machines.

Dale Jacquette [90] has proposed a variant of the Turing test in which the interrogator asks questions about the truth values of Gödel sentences and their negations. Since I do not expect all readers to be familiar with the Turing test, I will briefly explain it.

The Turing test was proposed by Turing [207] as a means to tackle the question whether machines can actually think. The test has the form of a game called the "imitation game." In this game, we have a person, a machine, and an interrogator, who is in a room separated from the person and the machine. The aim of the game is to allow the interrogator to ask questions to both the person and the machine and to determine from their responses which one is the person and which one is the machine. The interrogator knows the person and the machine by labels "X" and "Y," respectively, and at the end of the game she says either "X is a person and Y is a machine" or "X is a machine and Y is a person." The interrogator is allowed to put questions to the person and the machine thus:

> Will X please tell me the length of her hair?

(For a detailed discussion of the Turing test and related issues, see [144].)

According to Jacquette, the mind[3] may use an nonprogrammable nonalgorithmic procedure to judge whether some Gödel sentence is true. The procedure can be characterized as an "intensional conditional in the imperative mood" [90, p. 5]:

> (P) If S says that S is unprovable [relative to some recursively based logic], then answer (print): "S is true."

Jacquette claims that his is a nonalgorithmic implementable procedure, which can be interpreted as the claim that no programming language can be used to implement this procedure. However, this procedure is realizable by a mind simply because a mind understands the meaning of Gödel sentences and, most importantly, the meaning of the negation of Gödel sentences. This crucial information is used by the mind to decide when a sentence S says that it is unprovable. Interestingly enough, if the interrogator decides to explain that $S = \neg\,\mathrm{Thm}(n)$ and $\mathrm{Gn}(\neg\,\mathrm{Thm}(n)) = n$ (recall that $\mathrm{Gn}(A)$ denotes the Gödel number of any well-formed formula A), then both minds and machines can deduce that S is formally undecidable. However, if the interrogator has opted to choose predicates other than $\neg\,\mathrm{Thm}$ to represent unprovability, this information will become useless. Also, one might observe that the appearance of the external negation symbol in $\neg\,\mathrm{Thm}$

3. Clearly, only a mind that has mastered the complexities of mathematical logic can sit next to a machine and play this version of the imitation game. Thus, for the rest of this discussion, the word mind will not just mean any ordinary mind, but a mind well versed in mathematical logic.

could be used to distinguish Gödel sentences from their negations by first translating "problematic" constructions into prenex form, which are then checked for occurrences of an outermost negation symbol. But even this approach will not have the expected results, since, for example, one may replace ¬ Thm with NoThm with the expected semantic meaning. Thus, if both Gödel sentence formulations are tried out by the interrogator, the machine that employs the prenex trick will inevitably confuse Gödel sentences of the first formulation with negations of Gödel sentences in the second formulation.

The crux of Jacquette's argument against mechanism is that the mind's procedure is at the same time intensional and nonalgorithmic. A mind has no problem understanding any Gödel sentence as well as its negation. Thus, it can determine when a sentence says of itself that it is either unprovable or provable. In addition, Jacquette claims that a mind's intensionality and understanding of a sentence's meaning cannot really be simulated by a machine. This means that a machine cannot really fool the interrogator, since she can ask about the truth values of alternatively formulated Gödel sentences and their negations, thus forcing the machine into making mistakes.

Storrs McCall has put forth another argument against computationalism, initially in [121] and later on in [122]. This argument is based on the assumption that Turing machines *know* only what they can *prove* from a set of axioms and a set of well-defined rules of inference. Based on this, McCall tried to show that no Turing machine can know whether the Gödel sentence G of the form, "This statement is unprovable," is true. In a nutshell, the reason why this makes sense is that the truth value of G depends on the consistency of *Peano arithmetic*. Notice that Peano arithmetic, or just PA, is the theory of natural numbers defined by the five Peano axioms (named after the Italian mathematician Giuseppe Peano, who proposed them in 1889):

(i) $0 \in \mathbb{N}$ (zero is a natural number);

(ii) for each $x \in \mathbb{N}$, there exists exactly one $S(x) \in \mathbb{N}$, called the successor of x;

(iii) $S(x) \neq 0$ (zero is not the successor of any natural number);

(iv) $(S(x) = S(y)) \Rightarrow (x = y)$; and

(v) (induction schema) if φ is an arithmetic property such that 0 has this property and if $\varphi(x)$ implies $\varphi(S(x))$ for every x, then every number has the property φ.

Since it is not known whether PA is consistent, it is possible to argue about G by cases:

A_1. If PA is consistent, then G is not provable in PA.

A_2. If PA is consistent, then $\neg G$ is not provable in PA.

McCall assumes that whatever statement holds,[4] truth and provability diverge because if PA is consistent, then G is true and unprovable; but if it is inconsistent, then G is provable and false. Thus, PA contains nontheorems that are true or theorems that are false. According to McCall, the importance of this observation is that humans can distinguish the two entities, but Turing machines fail to do so. Let us assume that statement A_1 holds. In addition, the predicate Prov(Gn(A)) will denote that A is provable (or that A is not a theorem). Now, first note that $G \equiv \neg\,\text{Prov}(\text{Gn}(G))$. If by Cons(PA) we symbolize the statement "PA is consistent," then statement A_1 can be written formally as follows:

$$\text{Cons(PA)} \Rightarrow \neg\,\text{Prov}(\text{Gn}(G)).$$

Equivalently, statement A_1 can be written as

$$\text{Cons(PA)} \Rightarrow G.$$

McCall assumes that this is a theorem that can be proved. However, this may not be correct, since the proof has to be in PA itself and not in the metatheory. Let us now consider the formal version of A_2:

$$\text{Cons(PA)} \Rightarrow \neg\,\text{Prov}(\text{Gn}(\neg G)).$$

According to McCall, it can be shown that this statement is true (see [121] for details). However, one cannot have a formal proof of the "theoremhood" of this statement, and according to McCall, "there are good reasons to believe that [the formal version of statement A_2] is in fact unprovable in PA." The final result is that a Turing machine programmed to enumerate theorems in PA will almost certainly never include the statement above in the set of PA theorems. This, in turn, implies that there is a difference between human and machine thinking. Indeed, no computer program can model all of human reasoning.

Ignoring for the moment the remarks made by Tennant, one would not expect someone to find any flaws in this argument. However, Panu Raatikainen [162] has shown that there is a flaw in McCall's argument. In particular, Raatikainen has derived the formal equivalent of A_1, which implies that machines can make the distinction between true and derivable sentences. More specifically, by assuming Cons(PA), one may get[5]

$$\frac{\text{Cons(PA)} \quad \text{Cons(PA)} \Rightarrow G}{G} \Rightarrow \mathcal{E}\,.$$

4. As Neil Tennant [199] observes, the reasons for claiming that the first sentence is true (it can be proved within PA) are very different from the reasons for claiming that the second sentence is true (even if it is true, it is not provable within PA).

5. Assume that $\circledast$ is a logical operator. Then the symbols $\circledast\mathcal{E}$ and $\circledast\mathcal{I}$ denote an elimination rule for $\circledast$ and an introduction rule for $\circledast$, respectively.

As was noted previously, $\neg \text{Prov}(\text{Gn}(G)) \equiv G$, which can be written as

$$(\neg \text{Prov}(\text{Gn}(G)) \Rightarrow G) \wedge (G \Rightarrow \neg \text{Prov}(\text{Gn}(G))).$$

This, in turn, is used as a premise in the following deduction:

$$\frac{(\neg \text{Prov}(\text{Gn}(G)) \Rightarrow G) \wedge (G \Rightarrow \neg \text{Prov}(\text{Gn}(G)))}{G \Rightarrow \neg \text{Prov}(\text{Gn}(G))} \ \wedge 2\mathcal{E}\,.$$

Since G holds, we get

$$\frac{G \quad G \Rightarrow \neg \text{Prov}(\text{Gn}(G))}{\neg \text{Prov}(\text{Gn}(G))} \ \Rightarrow \mathcal{E}\,.$$

There is a small problem here: G cannot be proved inside PA. If we now apply $\wedge\mathcal{I}$, we get

$$\frac{G \quad \neg \text{Prov}(\text{Gn}(G))}{G \wedge \neg \text{Prov}(\text{Gn}(G))} \ \wedge\mathcal{I}\,.$$

And finally, by applying $\Rightarrow \mathcal{I}$ we get

$$\frac{\begin{array}{c}[\text{Cons(PA)}] \\ \vdots \\ G \wedge \neg \text{Prov}(\text{Gn}(G))\end{array}}{\text{Cons(PA)} \Rightarrow (G \wedge \neg \text{Prov}(\text{Gn}(G)))} \ \Rightarrow \mathcal{I}\,.$$

The conclusion is just the formal counterpart of A_1. Raatikainen finishes his paper by saying that although McCall's argument is not valid, this does not mean that computationalism is actually correct.

Bringsjord and his colleagues at the Rensselaer Artificial Intelligence and Reasoning (RAIR) Laboratory [25] reported their Gödelian argument for minds whose computational capabilities transcend the capabilities of the Turing machine. The members of the RAIR lab were involved in an effort to devise a (partial) solution to the busy beaver problem, and their efforts led to the formulation of their argument. Before going on, it is necessary to explain what this problem is about. The description of the problem that follows is from the busy beaver section of RAIR's lab web page:[6]

> Consider a binary-alphabet Turing Machine which is given an infinite, blank tape as input. If this machine halts, we define its productivity as the number of 1's left on the tape after the machine is run to completion. If it does not halt, the machine is given a productivity value of zero. Now consider all of the binary-alphabet Turing Machines that have n states. The machine in this set which has the highest productivity is called a Busy Beaver, and its productivity is the result of the Busy Beaver function $\Sigma(n)$. Alternatively, the productivity score can be defined as the number of transitions made before halting.

6. http://www.cs.rpi.edu/~kelleo/busybeaver/.

For reasons of brevity, the solution will not be discussed. Interested readers should point their web browsers to RAIR's busy-beaver web page for details.

The argument's goal is to refute computationalism, when it is understood as the supporting theory of the thesis that people are computers, which, in turn, are realizations of Turing machines. Assuming that p ranges over persons and m over Turing machines, this thesis can be stated as follows:

$$\forall p \exists m \; p \backsim m, \tag{$\mathscr{C}$}$$

where $\backsim$ is pronounced "are." This means that $p \backsim m$ can be interpreted as *p instantiates* (or *realizes*) *m*. Assume that each person is a realization of some Turing machine. If a measure of the mental capabilities of any person is equal to the measure of the complexity of a Turing machine (e.g., the number of states plus the number of transitions used), then all people are Turing machines whose measure of complexity is at or below some threshold. More specifically, if we assume that C is a function that has a Turing machine as argument and returns a number that characterizes its complexity, then the idea just presented can be written formally as follows:

$$\forall p \exists m \; (\; p \backsim m \wedge C(m) \leq k), \tag{$\mathscr{C}'$}$$

where $k \in \mathbb{N}$ is the threshold. The goal of Bringsjord's team was to devise an argument (not a proof) to refute the thesis that people are computers. The argument goes as follows:

- There are persons who have managed to determine the productivity of the initial segment of Turing machines (e.g., such persons are members of the RAIR lab; see [25] for details):

$$\exists p \Big(D(\; p, \Sigma(1)) \wedge \cdots \wedge D(\; p, \Sigma(6)) \Big). \tag{1}$$

- There is a natural number at and beyond which Turing machines with measure of complexity less than or equal to k fail to determine productivity:

$$\exists n \; \forall m \Big(\; C(m) \leq k \Rightarrow \neg D(m, \Sigma(n)) \wedge \neg D(m, \Sigma(n+1)) \wedge \cdots \Big). \tag{2}$$

- If a person can determine the productivity for n, then this same person can determine the productivity for $n + 1$:

$$\forall n \; \forall p \Big(D(p, \Sigma(n) \Rightarrow D(p, \Sigma(n+1)) \Big). \tag{3}$$

- Assume that computationalism, as expressed by ($\mathscr{C}'$), actually holds. Also, suppose that p^*, who is an arbitrary person, determines the initial segment of the busy-beaver problem, that is,

$$D(p^*, \Sigma(1)) \wedge \cdots \wedge D(p^*, \Sigma(n)). \tag{3'}$$

Since (C') holds for any person, it must hold true for p^*, that is,

$$\exists m\Big(p^* \backsim m \wedge C(m) \leq k\Big). \tag{4}$$

Let us randomly choose an m^* and an n^* such that

$$\Big(p^* \backsim m^* \wedge C(m^*) \leq k\Big) \tag{5}$$

and such that

$$\forall m\Big(C(m) \leq k \Rightarrow \neg D(m, \Sigma(n^*)) \wedge \neg D(m, \Sigma(n^* + 1)) \wedge \cdots\Big). \tag{6}$$

Clearly, (6) holds for m^*:

$$\Big(C(m^*) \leq k \Rightarrow \neg D(m^*, \Sigma(n^*)) \wedge \neg D(m^*, \Sigma(n^* + 1)) \wedge \cdots\Big). \tag{7}$$

From (5) and (7) we can deduce

$$\neg D(m^*, \Sigma(n^*)) \wedge \neg D(m^*, \Sigma(n^* + 1)) \wedge \neg D(m^*, \Sigma(n^* + 2)) \wedge \cdots .$$

By identity elimination and induction using (3), (5), and (3′), we can infer $\forall n\, D(m^*, \Sigma(n))$, which is a contradiction. From this it follows that since humans are information processors with capabilities lying somewhere in the arithmetic hierarchy and if humans are ordinary Turing machines they have a certain fixed size k, humans are hypercomputers.

Clearly, no one expects such an argument to win critical acclaim without any objection. On the contrary, there are issues that even Bringsjord et al. have spotted. For example, for skeptics, premise (3) practically implies that sooner or later people will be able to solve any problem. First of all, Bringsjord et al. respond by saying that what they claim does not mean that given enough time, anything is possible. They note that there are problems that even infinite-time Turing machines cannot solve, and such problems cannot be solved by any human. The essence of their argument is that if humanity "gets to n in the Σ problem, it can get to $n + 1$." And this is exactly the difference between Turing machines and humans: Turing machines cannot solve the problem for $n+1$ if they have successfully solved the problem for n; while it is also true that there is a limit to what humans can do, it is just above the limit of what machines can achieve.

Stewart Shapiro [179] has given an interesting account of the battle between computationalists and the Lucas–Penrose side over the Gödelian argument. Shapiro starts by exploring the meaning of the words "machine" and "human" in the context of this battle. Generally speaking, one may assume that when computationalists speak of machines they actually mean Turing machines, and when the Lucas–Penrose side speaks of humans they

actually mean creatures that have unlimited lifetimes, attention spans and energy, as well as unlimited resources at their disposal. In addition, another crucial assumption concerning these idealized human beings is that they do not make any mistakes! Both parties assume that there exists a set $\mathbf{K}$ consisting of "all and only the analogues of arithmetic theorems, sentences in the language of first-order arithmetic that can be known with unassailable, mathematical certainty" [179, p. 277]. This set is called the set of *knowable* arithmetic sentences. Since each element of $\mathbf{K}$ can be identified with its Gödel number, one may assume that $\mathbf{K} \subset \mathbb{N}$. Computationalists, quite expectedly, take it for granted that the Church–Turing thesis is valid and thus assume that $\mathbf{K}$ is recursively enumerable. Of course, the Lucas–Penrose side does not agree with this conclusion and argues that that there are procedures employed by humans that cannot be simulated by a Turing machine. Interestingly, its seems that hypercomputation has no place in this battle: computationalists completely deny it and the Lucas–Penrose side assumes that noncomputable processes are necessarily nonmechanical. Obviously, in the eyes of a proponent of hypercomputation both views are wrong: since the Church–Turing thesis is not valid, $\mathbf{K}$ is not recursively enumerable, while there are processes that transcend the Church–Turing barrier and that are purely mechanical. In spite of this, let us continue with Shapiro's analysis.

If $\mathbf{T}$ is the set of truths of first-order arithmetic, then by assumption $\mathbf{K} \subseteq \mathbf{T}$. However, let us suppose that $\mathbf{K} = \mathbf{T}$. Assume that Φ is an arithmetic proposition. Then either $\Phi \in \mathbf{T}$ or $(\neg\Phi) \in \mathbf{T}$. If $\Phi \in \mathbf{T}$, then $\Phi \in \mathbf{K}$ and so Φ is knowable. Otherwise, $(\neg\Phi) \in \mathbf{T}$ and $(\neg\Phi) \in \mathbf{K}$ and so it is knowable in principle that Φ is false. Let us recapitulate: if in the language of first-order arithmetic $\mathbf{T} = \mathbf{K}$, then for every arithmetic proposition Φ, an idealized human can determine whether Φ is true or false; that is, every arithmetic proposition can be *decided* by an idealized human being. Now, by Tarski's theorem on truth in arithmetic, no program can output a correct true or false value for every statement of number theory, which implies that $\mathbf{T}$ is not recursively enumerable. Thus, if $\mathbf{T} = \mathbf{K}$ and every arithmetic truth can be proved by an idealized human being, the set $\mathbf{K}$ is not recursively enumerable and the computationalists are wrong.

In order to defend their own belief, computationalists demand that $\mathbf{T} \neq \mathbf{K}$. Assume that $\Phi \in \mathbf{T}$ and $\Phi \notin \mathbf{K}$. Then Φ is an *unknowable* truth. This implies that both Φ and $\neg\Phi$ are *absolutely undecidable*, and so even an idealized human being cannot decide whether Φ is true or false. In other words, if what computationalists believe is true, there are absolutely undecidable arithmetic propositions.

In conclusion, this battle will be over once we know whether $\mathbf{T} = \mathbf{K}$. However, computationalists can easily avoid losing this battle, since "[they] are having trouble coming up with a reasonable mechanistic thesis for Lucas and Penrose to attack" [179, p. 300]. However, it seems that the whole battle is like trying to convince Alfred Square, resident of the

two-dimensional Edwin A. Abbott's Flatland [2], that there is a three-dimensional world. Clearly, this is almost impossible unless Alfred is able to enter the three-dimensional world in order to realize that his world is just part of this brave new world!

6.1.2 The Argument from Infinitary Logic

Another mathematically oriented argument, which is based on the isolation and exploitation of mathematical reasoning, is the argument from infinitary logic. Mathematical reasoning seems to be infinitary in nature and, consequently, one may argue that it is also irreducible to language usage. However, this seems to be a side issue irrelevant to the present discussion. The argument from infinitary logic aims at showing that the infinitary nature of mathematical reasoning is in general part of what makes a mind a hypermind. Our presentation is based on the exposition of the argument that is included in [26].

In order to apprehend the argument, it is necessary to be familiar with infinitary logic. The brief, rough exposition that follows is based on [11]. Assume that μ and λ are two infinite cardinals such that $\lambda \leq \mu$ and that $\mathcal{L}$ is a fixed first-order language. Also, suppose that Φ is a set of formulas of $\mathcal{L}$ such that $\text{card}(\Phi) < \mu$. Then $\bigwedge \Phi$ and $\bigvee \Phi$ will denote infinite conjunctions and disjunctions with $\text{card}(\Phi)$ conjuncts or disjuncts, respectively. In addition, if X is a set of individual variables such that $\text{card}(X) < \lambda$ and φ is an $\mathcal{L}$-formula, then $\exists X\varphi$ and $\forall X\varphi$ are formulas. Moreover, if φ and ψ are $\mathcal{L}$-formulas, then $\varphi \wedge \psi$ and $\neg\varphi$ are formulas. More generally, all $\mathcal{L}$-formulas are formulas. A language having these characteristics is an infinitary language, denoted by $\mathcal{L}_{(\mu,\lambda)}$. In particular, the language $\mathcal{L}_{(\omega_1,\omega)}$, where ω_1 denotes the set of countable ordinals, is one that allows countably infinite conjunctions but only finite quantifications. Now we can proceed with the argument from infinitary reasoning as presented in [26]:

(i) All reasoning is computable.

(ii) For every case of reasoning R there exists a Turing machine (or any equally powerful device) M such that some computation C of M is such that $R = C$ [from (i)].

(iii) For every computation C of every Turing machine M there is an equivalent deduction D in some instantiation of $\mathcal{L}_I$ (i.e., first-order logic).

(iv) For every case of reasoning R there exists a deduction D in some instantiation of $\mathcal{L}_I$ such that $R = D$ [from (ii), (iii); universal elimination, hypothetical syllogism, and universal introduction].

(v) There exists a case of reasoning R^*, namely reasoning with $\mathcal{L}_{(\omega_1,\omega)}$, that is such that for every deduction D in some instantiation of the first-order logic $\mathcal{L}_I$, $R^* \neq D$.

(vi) It is not the case that all reasoning is *computable* [*reductio ad absurdum*; (iv), (v) contradictory].

The designers of this argument claim that it is valid because the inferences are formally correct. In addition, they discuss a number of objections to this argument. The first objection is that this argument is not really *convincing*. Their response to this objection is simple: it is one thing to have a convincing argument and another thing to have a *sound* argument. Furthermore, it is important to notice that the history of science is full of *unconvincing* but sound theories, such as the theory that the Earth moves around the Sun.

Another objection concerns reasoning in and about $\mathcal{L}_{(\omega_1,\omega)}$ that is simply manipulation of finite expressions that are clearly computable, such as the following expression borrowed from [26, p. 108]:

$$\bigvee_{n<\omega} \exists x_1 \cdots \exists x_n \forall y (y = x_1 \vee \cdots \vee y = x_n).$$

The essence of the response to this objection is that although Hilbert noticed that proofs are presented as finite strings on finite sheets of paper and consequently put forward the ideas we presented in the introductory chapter, Gödel managed to abolish Hilbert's ideas. In addition, Gödel proved that "human mathematical reasoning is *not* always limited to Hilbertian reasoning: some form of infinitistic reasoning must be employed for some proofs of formulas about $\mathbb{N}$" [26, p. 109].

6.1.3 The Modal Argument

According to Selmer Bringsjord and Konstantine Arkoudas [23], there are basically two methods for attacking computationalism when starting from mathematical results in the realm of incompleteness. The first method is the one described in the previous section, while the second method is the one that will be presented in this section. The proof that minds are not Turing machines is a two-stage process. First, it is necessary to make suitable idealizations of minds and machines, and then one must prove a formally valid modal argument.

Like Shapiro, Bringsjord and Arkoudas believe that idealized computers can be identified with ordinary Turing machines. Unlike Shapiro's idealized humans with unlimited capacities, the idealized humans of Bringsjord and Arkoudas take input and yield output that reflects decisions based on the inputs taken. Also, they assume that (part) of the human mind is actually an information-processing device.

Let us consider the following decision problem: Given a Turing machine $\mathscr{M}_0$ and an input string w, does $\mathscr{M}_0$ halt on input w? It has been proved that there is no algorithm (i.e., no Turing machine) that can decide this problem. Assume that $D(\mathscr{M}, \mathscr{M}', i)$ is a predicate that stands for the sentence, "Turing machine $\mathscr{M}$ determines whether Turing machine $\mathscr{M}'$ halts on input i." Using this predicate we can formally specify the undecidability of the problem above in quantified modal logic as follows:

$$\forall \mathscr{M} \exists i \neg \Diamond D(\mathscr{M}, \mathscr{M}_0, i). \tag{M1}$$

Notice that the modality $\Diamond$ is associated to logical or mathematical possibility, that is, $\Diamond\varphi$ if and only if it is logically or mathematically possible that φ. Assume that $M(x)$ stands for "x is a Turing machine," $P(x)$ for "x is a person," and $I(x)$ for "x is input for a Turing machine." Then (M1) can be written as follows

$$\exists x \Big(M(x) \wedge \forall y \Big(M(y) \rightarrow \exists u \big(I(u) \wedge \neg \Diamond D(y, x, u) \big) \Big) \Big).$$

For the sake of argument let us assume that persons are indeed Turing machines, or, more accurately, that persons are physically realized Turing machines. This assumption can be specified in the following way:

$$\forall \mathcal{P} \exists \mathscr{M} \; \mathcal{P} \backsim \mathscr{M}. \tag{M2}$$

From (M1) and (M2) we deduce that

$$\forall \mathcal{P} \exists i \neg \Diamond D(\mathcal{P}, \mathscr{M}_0, i). \tag{M3}$$

Bringsjord and Arkoudas conclude that since there are persons, (M3) is inconsistent with

$$\forall \mathcal{P} \forall i \Diamond D(\mathcal{P}, \mathscr{M}_0, i). \tag{M4}$$

And so if we can prove (M4), we have an indirect proof of ¬(M1), which means that computationalism is false.

The crucial question is whether (M4) is actually true. Before going on, it is necessary to clarify that the "modal argument is not inseparably linked to a particular formal derivation or a particular proof theory." This means that one may present this argument even in first-order logic. However, the authors have presented their argument in this manner because they happen to be comfortable with it. Clearly, this book is about hypercomputation, and so far we have presented a good number of conceptual devices that transcend the capabilities of the Turing machine that are eventually realizable. Assume that $\mathscr{H}$ stands for any hypermachine. Then it follows that

$$\forall \mathscr{H} \forall i \Diamond D(\mathscr{H}, \mathscr{M}_0, i). \tag{M5}$$

If we are inclined to assume that a person may be a hypermachine and not just a Turing machine, we can formally express this as follows:

$$\Big(\forall\mathscr{H}\forall i\Diamond D(\mathscr{H},\mathscr{M}_0,i)\Big)\rightarrow\Big(\forall\mathcal{P}\forall i\Diamond D(\mathcal{P},\mathscr{M}_0,i)\Big). \tag{M6}$$

Proposition (M4) follows by *modus ponens* from (M5) and (M6).

No argument remains unchallenged, and this argument is no exception. In the remainder of this section I will present two objections to the modal argument as well as the responses offered by the designers of the argument.

If one assumes that computationalism is the belief that people are physical computers, then one may hope to refute the modal argument. In particular, if we assume that $\mathcal{C}$ ranges over embodied computers, then the formal expression describing computationalism takes the following form:

$$\forall\mathcal{P}\exists\mathcal{C}\ \mathcal{P}\backsim\mathcal{C}. \tag{M2$'$}$$

Based on this, proposition (M1) must be replaced with

$$\forall\mathcal{C}\exists i\neg\Diamond D(\mathcal{C},\mathscr{M}_0,i). \tag{M1$'$}$$

But this proposition is false, since there is some machine $\mathcal{C}_0$ (e.g., an oracle Turing machine, a trial-and-error machine) that can solve the halting problem for $\mathscr{M}_0$. Computationalism is the doctrine that advocates that persons are just symbol processing "machines" and not hypermachines, which implies that (M1$'$) cannot possibly be true.

Let us now discuss the second objection, which is based on the common belief that modern physical computers running some program P are physically instantiated Turing machines. Obviously, at this point we should pretend that there is no *empirical* evidence for the view that modern digital computers are not Turing machines. Suppose that $\mathscr{B}$ ranges over modern digital computers running some program P. Then proposition (M2) takes the following form:

$$\forall\mathscr{B}\exists\mathscr{M}\ \mathscr{B}\backsim\mathscr{M}. \tag{M2$''$}$$

It follows from (M1) and (M2$''$) that

$$\forall\mathscr{B}\exists i\neg\Diamond D(\mathscr{B},\mathscr{M}_0,i). \tag{M3$''$}$$

But this proposition is inconsistent with

$$\forall\mathscr{B}\forall i\Diamond D(\mathscr{B},\mathscr{M}_0,i). \tag{M4$''$}$$

This means that digital computers running some program are not computers! The problem is that proposition (M4$''$) is true only if every digital computer is actually a hypercomputer, while on the other hand, proposition (M3$''$) is true only if modern digital computers are instantiations of Turing machines.

6.2 Philosophy and the Mind

The mind as an object of philosophical inquiry has been a very attractive subject of study for several thousand years. Almost every philosopher has had something to say about the mind, which in many cases has affected people's lives in quite unexpected ways. In particular, various prejudices and folk beliefs have deeply affected the formation of philosophical doctrines, which, in turn, reflect these prejudices and beliefs. For instance, as Searle notes in [176], Cartesian dualism gave the material world to scientists and the mental world to theologians. Thus the new scientific discoveries of the time posed no threat to traditional religion. Although the philosophy of the mind is a very interesting subject, we will concentrate on arguments against computationalism. The reader with a general interest in the subject should consult any textbook on the philosophy of mind.

6.2.1 Arguments Against Computationalism

Let us now present a number of important arguments against computationalism. The presentation of these arguments is highly influenced by Searle's presentation in [176].

The term *qualia* (singular: quale) refers to the ways things seem to us. In particular, qualia describe the qualitative character of conscious experiences. To make things clear, imagine that you and a friend are staring at a landscape at sunset. The way it looks to you–the particular, personal, subjective visual quality of the landscape–is the quale of your visual experience at the moment. Perhaps, that is why no color model (i.e., a mechanism by which we can describe the color formation process in a predictable way) can accurately describe colors. Since qualia really exist and computationalism does not take them into account, one may conclude that computationalism is false. Note that we assume here that computationalism and *functionalism* are being conflated. Functionalism, which is a doctrine quite similar to computationalism, argues that what it takes to be a mind is independent of its physical realization.

Thomas Nagel [139] argues that although one may have perfect knowledge of a bat's neurophysiology, she will not be able to say what it is like to be a bat. Even if she could by gradual degrees be transformed into a bat, she could not imagine the way she would feel when, eventually, she would be metamorphosed into a bat. The argument is based on the observation that bats have a sensory apparatus considerably different from ours, and it aims to show that having complete knowledge of everything that goes on inside the body of an animal is still insufficient to explain consciousness. Yujin Nagasawa [137] has put forth an interesting objection to Nagel's argument. More specifically, Nagasawa claims that if we have a vivid imagination

or a sophisticated simulation system, there is no problem for us to know what it is like to be bat without being a bat like creature. However, an immediate response is that one cannot really say how it feels like to "enjoy" smoking a cigarette when one has never smoked one. Imagination is simply not enough!

A similar argument is the one that Frank Jackson published in [89]. Assume that it is possible to create a dome inside of which everything is black and white. Maria grows up in this dome and she is educated by watching distant-learning programs on a black-and-white television set and by reading black-and-white books and magazines. In this way Maria learns everything other pupils learn about the physical world that surrounds us. Thus, she knows there are objects that are red, but she has never seen any red object in her life. Now, if Maria knows everything she should know, she should have no problem recognizing the red Ferrari in a full-color photo of sport cars. But this is not true, since the very moment she sees the red Ferrari in the photo, she will learn what it is like to sense red. In a nutshell, knowledge is not enough to know what it is like to sense colors.

Another argument against computationalism has been put forward by Ned Block. This argument is considered by many as an immediate antecedent to the Chinese room argument. Block's argument goes like this: Assume that the brain of a typical human being consists of around 1.5 billion neurons.[7] Also, assume that each Chinese citizen plays the role of a neuron. For instance, neuron firing can be *simulated* by the act of calling another person using a cellular phone. This "artificial" brain lacks mental states (e.g., one cannot claim that it "feels" wrath), and thus it cannot be classified as a real brain. A similar argument was advanced by Searle [174]. This argument has been dubbed the "Chinese gym" argument, while Block's argument is known as the "Chinese nation" argument. Searle's argument goes like this. Imagine that there is a hall containing many monolingual English-speaking men. These men would carry out the same operations as the neurons of a connectionist architecture (i.e., neural networks) that models the brain process that take place on the brain of the human in the Chinese room argument. No one in the gym speaks any Chinese, and there is no way to imagine that the system considered as a single entity, understands Chinese. Yet the system gives the impression that it understands Chinese.

7. Actually, the brain of an adult human has more than 100 billion neurons [181], but the core of Block's argument is that the entire population of China will implement the functions of neurons in the brain. Thus, we cannot actually use the real figure for the presentation of the argument.

6.2.2 The Chinese Room Argument Revisited

In 2004, the Chinese room argument (CRA) turned 25 years old, but age is not slowing the CRA down. It is still a subject of debate as well as a source of inspiration for members of the scientific community. And the recent collection of selected papers on the CRA [159], which was edited by John Preston and Mark Bishop, as well as Jerome Wakefield's recent paper on the CRA [214], which was the most viewed paper on the "Minds and Machines" web site in 2003, are clear proofs of this. Apart from its popularity, the real question is whether the CRA is still valid. And that is exactly the subject of this section.

An interesting idea concerning the validity of the CRA was put forth by Bruce MacLennan [118], who rightly claims that if one accepts the CRA in the digital setting, then one should also accept it in the analog setting, and conversely, if one does not accept it in the digital setting, then one should not accept it in the analog setting. Here the term "analog setting" refers to analog computation (see Chapter 9). MacLennan does not believe in the validity of the CRA. He has proposed the "granny room argument," that is, the analogue of the CRA in an analog computing setting, in order to refute the CRA based on his view that one has to accept or reject the applicability of the CRA in both the digital and analog settings. In the granny room there is a person who is exposed to a continuous visual image and produces a continuous auditory output. By making use of various analog computational aids the person in the room "implements the analog computation by performing a complicated, ritualized sensorimotor procedure." When the system sees an image of MacLennan's grandmother it will respond, "Hi, Granny!" MacLennan believes that his argument refutes the CRA, but the truth is that this argument does not actually do so. In fact, one may question the point of substituting symbol recognition with face recognition. Assume that the person inside the room has photos for each face that can possibly appear, and depending on the face seen, she produces what MacLennan calls a *(continuous) auditory image*. For example, when she sees face A, she has to say, "Hello, Stella!" Practically, this argument does not differ from the classical "digital" version of the CRA. The point is that facial recognition is no different from symbol recognition, and thus the CRA remains immune from attack even in the "analog" setting.

Jerome C. Wakefield [214] presents some interesting ideas concerning the CRA. In particular, he criticizes the formulation of the CRA as presented by Searle in [175, pp 11–12]:

i. Programs are entirely syntactic.

ii. Minds have a semantics.

iii. Syntax is not the same as, nor by itself sufficient for, semantics.

$\therefore$ Programs are not minds.

According to Wakefield, this syllogism is problematic because the third premise is a "straightforward denial of computationalism," since "the no-semantics-from-syntax intuition is precisely what strong AI proponents are challenging with their computationalistic theory of content." However, we feel that it is necessary to see first what Searle has to say in [175] about the third step:

> [T]he general principle that the Chinese Room thought experiment illustrates: merely manipulating formal symbols is not in and of itself constitutive of having semantic contents, nor is it sufficient by itself to guarantee the presence of semantic contents. It does not matter how well the system can imitate the behavior of someone who really does understand, nor how complex the symbol manipulations are; you cannot milk semantics out of syntactical processes alone.

To claim that 2 is a set is clearly counterintuitive. In addition, depending on which sets one identifies with the natural numbers, there are many other things that are equally counterintuitive (e.g., $2 \in 3$). However, the existence of such counterintuitive results does not mean that the reduction of numbers to sets is problematic. Similarly, one cannot claim that since there some counterintuitive results in the thought experiment associated with the CRA, one can object to the claim that certain computer states are beliefs, which, in a nutshell, is the essentialist objection to the CRA. Wakefield claims that this is a valid objection to the CRA and argues that the CRA can be reinterpreted in such a way as to make it immune to the essentialist attack. In particular, if we explain the meaning of the CRA in an indeterminate way, the new argument still poses a challenge to computationalism and strong AI. This new formulation of the CRA has been dubbed the *Chinese Room Indeterminacy Argument*, or just CRIA.[8] Wakefield's CRIA goes as follows [214]:

i. There are determinate meanings of thoughts and intentions-in-action. In addition, a thought about a syntactic shape is different from any thoughts that possess the semantic content that is expressed by the syntactic shapes.

ii. Any syntactic fact underdetermines, and at the same time leaves indeterminate, the contents of thoughts and intentions-in-action.

$\therefore$ The content of thoughts and intentions-in-action cannot be constituted by syntactic facts.

And as Wakefield notes, "[t]his indeterminacy argument provides the needed support for Searle's crucial third premise."

8. An argument is called *indeterminate* when it is open to multiple incompatible interpretations consistent with all the possible evidence.

If one could have demonstrated that syntax is indeed the same as semantics, then she would have managed to refute the CRA. And the easiest way to achieve this goal would be to show or at least to provide evidence that some computer program understands. By following this line of thinking, Herbert Alexander Simon and Stuart A. Eisenstadt [184] describe three programs they believe provide evidence that computer programs can understand and thus falsify the CRA. For instance, they present a program called ZBIE that *simulates* human language learning. The program has as inputs sentences in any natural language and description lists that represent simple scenes (e.g., "The boy pulls on the oar under the lash."). After some time, the program acquires a vocabulary of words related to the scenes it has as input and a vocabulary of relational structures. In addition, using sentences in two languages as inputs, instead of sentences and scenes, ZBIE can learn to translate from one language to another. However, there are some issues with these "astonishing" capabilities. First, notice that even modern specialized programs fail to provide meaningful translations. For instance, the author used a mechanical language translator to translate the sentence above to Greek and the resulting text back to English only to get back the completely different sentence, "the boy pulls in the oar under the whip." And of course, if a modern professional tool does this kind of work, what should one expect from a tool of the early 1970s? On the other hand, when someone learns a new word, she tries to associate this new word with her own experiences so as to grasp its real meaning. For example, when a juvenile learns the word "orgasm," she will not really understand the real meaning of the word until the day she first experiences an orgasm. So syntax is simply not enough to understand what an orgasm is. In other words, Jaak Panksepp's question, "Could you compute me an orgasm?" has a negative answer. More generally, it is meaningless to say that a computer program understands just because some talented computer programmer has figured out a number of cases that make a computer program appear as if it really understands. A clever set of rewriting rules cannot possibly be equated with understanding.

6.3 Neurobiology and the Mind

The brain is part of the central nervous system and includes all the higher nervous centers. It is also the center of the nervous system, and the seat of consciousness and volition. As such, it is of great importance to neurobiologists. Until recently, most biologists employed *reductionism* (i.e., the idea that the nature of complex things can always be explained by simpler or more fundamental things) to explain biological phenomena (e.g., the discovery of the structural and chemical basis of living processes is a result of the application of reductionism to biology). However, it is quite surprising,

particularly for nonspecialists, that biologists are gradually abandoning reductionism in favor of *emergence*, which is roughly the idea that some properties of a system are irreducible. Indeed, as Marc Van Regenmortel notes [210]:

> Complex systems are defined as systems that possess emergent properties and which, therefore, cannot be explained by the properties of their component parts. Since the constituents of a complex system interact in a non-linear manner, the behaviour of the system cannot be analysed by classical mathematical methods that do not incorporate cooperativity and non-additive effects.

And he concludes by stating that "reductionism is not the panacea for understanding the mind." Interestingly enough, biological naturalism is an explanation of the so-called *mind-body problem* (i.e., "How can a decision in my soul cause a movement of a physical object in the world such as my body?" [176, p. 17]) that is based on exactly these principles. More specifically, biological naturalism is based on the following theses [176, pp. 113–114]:

(i) Conscious states, with their subjective, first-person ontology, are real phenomena in the real world. It is impossible to do an eliminative reduction of consciousness in order to show that it is just an illusion. In addition, it is not possible to reduce consciousness to its neurobiological basis, because such a third-person reduction would leave out the first-person ontology of consciousness.

(ii) Conscious states are entirely caused by lower-level neurobiological processes in the brain. Conscious states are thus *causally reducible* to neurobiological processes. However, they have absolutely no life of their own independent of the neurobiology. Causally speaking, they are not something "over and above" neurobiological processes.

(iii) Conscious states are realized in the brain as features of the brain system, and thus exist at a level higher than that of neurons and synapses. Individual neurons are not conscious, but portions of the brain system composed of neurons are conscious.

(iv) Because conscious states are real features of the real world, they function causally. For instance, the reader's conscious thirst causes him or her to drink water.

As a direct consequence, one can surely simulate in principle the functioning either of parts of the brain or of the whole brain in a computer. However, it is impossible for the computer simulation to become conscious. In order to make things clear, let us give a somewhat trivial argument. Many people are aware that water is the chemical compound H_2O and that

ethanol is the chemical compound CH_3CH_2OH. Each water molecule consists of atoms, which, in turn, consist of electrons, neutrons and protons. And of course the same applies to the ethanol molecules. The question is, since both water and ethanol consist of exactly the same basic building blocks, why do they taste different, and more generally, why do they have different properties? Certainly, the answer is that their molecules consist of different numbers of electrons, neutrons, and protons and that these elementary particles are arranged in different ways. So it is not enough to know the constituents of a compound to have a complete image of its properties. Analogously, one may say that it is not enough to study the properties of neurons and how they are connected in order to (fully) understand the brain and its operations.

If we suppose that the computational theory of the mind is indeed true, then we should expect that the brain operates in a discrete manner. Indeed, according to "modern" computationalism, the brain operates in discrete manner in a discrete universe. However, to the disappointment of many computationalists, Michael Spivey and his colleagues Marc Grosjean and Günther Knoblich [188] reported that there is compelling evidence that language comprehension is a continuous process. In their experiment, Spivey and his colleagues had at their disposal forty-two volunteers, who were Cornell University undergraduate students who took psychology courses. Each volunteer was presented with color images of two objects on a screen, and a prerecorded audio file instructed them to click one of the images with a mouse. One of the objects had the role of a distractor object and the other the role of a target object. When the students were instructed to click one of the two objects and the names of the objects did not sound alike, such as apple and jacket, the trajectories of their mouse movements were straight and direct to the objects they were instructed to click on. On the other hand, when the students were instructed to click on an "apple" and were presented with two objects with similar sounding names (e.g., "apple" and "maple"), they were slower to click on the correct object, and in addition, their mouse trajectories were much more curved.

This experiment provided powerful support for models of continuous comprehension of acoustic-phonetic input during spoken-word recognition. In addition, the data gathered from this experiment provide support to the claim that the continuous temporal dynamics of motor output reflect continuous temporal dynamics of lexical activation in the brain. In other words, one may say that cognition does not operate by entering and leaving states (e.g., like a state machine or automaton) but rather can have values in between (e.g., it may be partially in one state or another) and eventually stabilizes to a unique interpretation, which, for example, can be the recognition of a certain word.

Panksepp is the father of the emerging field of *affective neuroscience*, which supports the idea that affective and cognitive mental processes are distinct. A summary of "recent conceptual and empirical advances in

understanding basic affective process of the mammalian brain and how we might distinguish affective from cognitive processes" was presented in [146]. The following short presentation of affective neuroscience is based on this paper.

It is a common belief, shared particularly among nonscientists, that emotional processes have both *cognitive* and *affective* attributes. In addition, these attributes rank highest among a number of other attributes emotional processes may have. However, because of the difficulty unambiguously distinguishing the two attributes in the laboratory, many scientists have begun to question the utility of this distinction. In spite of this skepticism, Panksepp believes that this very distinction may prove helpful in deciphering the neurobiological nature of the basic affective quality of conscious actuality. Panksepp advances this idea because affective feelings are, not completely but to a considerable degree, distinct neurobiological processes from an anatomical and a neurochemical point of view. Also, this distinction is evident to a similar degree with respect to peripheral bodily interactions. Emotional and motivational feelings "push" organisms to make cognitive choices (e.g., to find food when hungry, water when thirsty, companionship when lonely). If this idea is indeed true, then it is necessary to develop special techniques to understand affective organic processes in neural terms, which, in turn, may provide a solid basis for the construction of a coherent science of the mind. As a side effect of such a development, new psychiatric therapeutics will be advanced. Interestingly, the foundation on top of which emotional and motivational processes are built is analog in nature. In addition, this foundation is to a large degree the result of evolutionary process. Let us now see why Panksepp advocates the distinction between affects and cognitions.

First of all, emotional states are inherently characterized by valence. In other words, they are characterized by either aversive or attractive feelings that do not accompany pure cognitions. It is not entirely unreasonable to suppose that various basic emotional and motivational responses and the accompanying types of valence have their origin in inherently evolutionarily controlled states of the nervous system. These mental abilities of the brain are not built just from the perceptions of external events and the cognition that follows. Instead, they have an intrinsic structure of their own. However, emotions are not just disturbances of the physical setting in which they occur. In addition, they help control the way we perceive the world around us.

Although many forms of brain damage severely impair cognitions, still emotional responses and many basic affective tendencies are not affected. This dictum is based on the fact that early decortication (i.e., removal of the outer covering of the brain) of neonatal rats affects the ability of these animals to learn while their emotional and motivational behaviors remain almost intact. Ralph Adolphs, Daniel Tranel, and Antonio Damasio [3] reported the results of their study to test the hypothesis that the recognition

of emotions is probably "composed" in different brain regions, which depends on the nature of the stimuli that have caused these emotions. Adolphs et al. studied a person who had suffered extensive bilateral brain lesions, and their findings support the dictum above. These and similar observations have led Panksepp to conclude that, "*Cognitions are largely cortical while affects are largely subcortical.*"

It is an everyday observation that children are very emotionally alive, which suggests that "affective competence *is* elaborated more by earlier maturing medial brain systems than more rostrally and laterally situated cognitive systems" [146, p. 10]. These remarks affirm that affects are more likely to be evolutionary "givens." The higher cortico-cognitive processes that keep in check emotionality appear gradually as the organisms mature.

Processes that resemble discrete computational processes may generate cognitions, while neurochemical processes that resemble analog computational processes may be responsible for the generation of affects. A direct consequence of this observation is that in the case of long-term emotional learning, the conditioning of holistic "state" responses plays an important role, while in the case of cognitive learning, the temporal resolution of formal operations and propositions plays an important role. Probably, this is the reason why it is hard to activate cognitions by directly stimulating the brain, while this does not hold true for affects.

Cognitions do not generate facial or bodily expressions and do not have any effect on the tone of our voice, while emotions generate such expressions and changes in tone. Although the importance of facial expressions in the study of emotional feelings has not remained unchallenged, still it is clear that these emotional actions can cause congruent feelings. And in cases in which someone has suffered cortical damage, full emotional expressions cannot be generated by cognitive means, while they can be aroused by spontaneous emotional states.

Over the past 15 or more years, various studies have revealed emotional asymmetry and asymmetries in motor output (for instance, see [46, 78, 91, 185]). There are two general theories of emotional asymmetry: the right-hemisphere hypothesis and the valence hypothesis. According to the right-hemisphere hypothesis, the right hemisphere is the center of all forms of emotional expression and perception. On the other hand, the valence hypothesis posits that emotional valence deeply affects hemispheric asymmetry for expression and perception of emotions. More specifically, the right hemisphere is dominant for negative emotions and the left hemisphere is dominant for positive emotions. Both hypotheses have received empirical support.

It is an unfortunate fact that our way of thinking and perceiving the world around us is constrained by prevailing cultural and scientific assumptions. And this is why affective issues have been confronted with great skepticism. However, this attitude is changing, and a growing number of researchers now recognize the importance of affects. One of the main

reasons for this turnaround is that by understanding what affects really are, we may hope to understand what consciousness really is.

6.4 Cognition and the Mind

The arguments and ideas presented in this section have appeared in periodicals whose scope is marginally related to the philosophy of mind.

In Section 3.1.2 we presented a model of the mind based on the assumption that the mind is a trial-and-error machine. Here we follow a different path by assuming that the mind is a machine that has semantic content.

People are definitely not computers, but people are definitely (some sort of) *machines*, since they can calculate, memorize, etc. And naturally the question is, What kind of machines are people? James H. Fetzer presents some interesting ideas on this matter in [62]. A sign is a generalization of the concept of a symbol. Charles Sanders Peirce divides signs into three categories: *icons*, *indices*, and *symbols*. Here is how Peirce explains the difference among these three categories:[9]

> There are three kinds of signs. Firstly, there are *likenesses*, or icons; which serve to convey ideas of the things they represent simply by imitating them. Secondly, there are *indications*, or indices; which show something about things, on account of their being physically connected with them. Such is a guidepost, which points down the road to be taken, or a relative pronoun, which is placed just after the name of the thing intended to be denoted, or a vocative exclamation, as "Hi! there," which acts upon the nerves of the person addressed and forces his attention. Thirdly, there are *symbols*, or general signs, which have become associated with their meanings by usage. Such are most words, and phrases, and speeches, and books, and libraries.

One may say that an icon is a thing that resembles that for which it stands, an index is a cause or an effect of that for which it stands, and a symbol is merely habitually or conventionally associated with that for which it stands. Based on this division, Fetzer suggests that there should be at least three kinds of minds. More specifically, Type I minds that can process icons, Type II minds that can process icons and indices, and Type III minds that can process icons, indices, and symbols. Although Fetzer stopped here, we can go on and introduce Type IV minds as minds that manipulate only indices and Type V minds as minds that manipulate only symbols. However, computers process symbols according to their form and not the meaning

9. The excerpt is from Peirce's paper entitled *What Is a Sign?* which is available online from `http://www.iupui.edu/~peirce/ep/ep2/ep2book/ch02/ep2ch2.htm`.

that may be associated to them by usage. On the other hand, a mind capable of processing symbols, while aware of the meaning associated with their "meaning," is clearly different from a Type V mind. Let us call these minds Type VI minds. It seems that the sci-fi androids are Type VI minds, but I will not argue about this idea.

A Type III mind is actually a *semiotic* system; while a modern computing system (i.e., a Type V "mind") is a symbolic system. There are two differences between semiotic and symbolic systems. First, a symbolic system is able to process syntax (i.e., it is able to manipulate meaningless marks), while a semiotic system is able to process signs that are meaningful for this system. Second, a symbolic system manipulates marks by executing some computational procedure, but a semiotic system manipulates signs by noncomputational procedures. Human thought processes cannot be described by symbol systems, but they can be described by semiotic systems. Another important difference between semiotic and symbolic systems is that in the case of semiotic systems there is a "grounding" relationship between signs and what they stand for, while in the case of symbolic systems, such a relation does not exist.

These observations have led Fetzer to propose that the mind is actually a semiotic engine. As such, the mind processes information in a nonalgorithmic way.

Quite recently, Chris Eliasmith discovered a major flaw in functionalism and reported it in [57]. Recall that the Turing machine is a conceptual device, and as such, its properties are independent of any particular realization. In addition, it is easy to characterize a Turing machine from its input, the state of the machine, and the program being executed. Functionalists believe that what makes something a mental state depends on its function in the cognitive system of which it is a part. More specifically, mental states are functional relations between sensory stimulations (input), behavior (state of the machine), and their mental states (the program being executed). Thus, cognitive functions can be completely characterized by high-level descriptions abstracted from their implementation. Also, "two systems are functionally isomorphic if *there is a correspondence between the states of one and the states of the other that preserves functional relations*" [161]. This implies that any system isomorphic to a mind is a mind. Assume that there are two functionally isomorphic systems having different implementations. Then these will have the same mental states (if any). This is the thesis of multiple or universal realizability (i.e., the fallacious claim that anything can be described as implementing a computer program), which Eliasmith refutes in [57].

The argument against the multiple realizability thesis is based on the idea that two computing devices that are equivalent (or isomorphic if you prefer this term) are not equal. In particular, machine equivalence provides little information regarding the way a machine actually computes something, and it is this way that is cognitively relevant. For instance, although

a modern CISC machine is equivalent to a RISC machine, in the sense that one can compile and execute exactly the same programs on both machines, a RISC machine is faster (e.g., consider operating systems, such as OpenSolaris and GNU/Linux, that are available for both architectures and think about their performance). Clearly, if we compile the same program under the same operating system running on two different architectures the resulting binary files will be completely different. Obviously, both binaries will produce the same results, but one will be executed much faster than the other. The reason for this difference in performance is due to the simplicity of the RISC architecture or to the complexity of the CISC architecture. Clearly, this means that the implementation, contrary to the functional belief, really matters. In other words, a system that is functionally isomorphic to a mind is not necessarily a mind.

By having as a starting point "the cognitive study of science," Roland Giere [64] shows that only "distributed cognition" can be employed to understand cognition as it occurs in modern science. Giere uses an example to demonstrate the validity of his ideas. In particular, he considers the large hadron collider of the European Center for Nuclear Research (known as CERN), which is coupled with a very large detector called ATLAS. The ATLAS project involves many scientists, technicians, and support personnel and aims to obtain direct experimental evidence of the existence of the Higgs boson.[10] Since there is no reason to explain all the details involved, it sufficies to say that the experiment involves the acceleration of certain elementary particles to very high energies and their subsequent collision in the detector. Depending on what goes on in the detector, one may decide whether the Higgs boson actually exists.

When finished, the ATLAS project will produce some knowledge, which is actually a cognitive product. Thus, one may view the ATLAS project as a cognitive process. Clearly, one may wonder about the nature of scientific cognition starting with this particular example. As expected, the "standard" answer to this problem is that a *cognitive agent*, which is a human or artificial *individual*, acquires a *symbolic representation* that is *computationally* processed according to a set of syntactic rules. This answer is problematic for a number of reasons. First of all, it is not clear who or what this cognitive agent is. A typical answer to this question is that the cognitive agent is the person who interprets the final output. There are two problems with this response. First, if we assume that such a person indeed exists, then this person "operates" by manipulating and thus is incapable of understanding anything. Second, there is actually no such person, since the final output is the result of a complex interaction among people with different kinds of expertise who consult sophisticated equipment. Thus, we cannot find a single person who has the required property.

A partial solution to these problems emerges if we consider the notion

10. The Higgs boson is a hypothetical particle whose very existence would validate the "standard" mechanism by which particles acquire mass.

of *collective cognition*. In this setting, we assume that each individual participating in the project is actually a computational system. We may therefore say that the final output is the conjunction of the outputs produced by each individual. Clearly, this solution insists that humans are computers and as such is simply unacceptable. Apart from that, it does not take into consideration the artifacts that play a crucial role in the project.

A better idea is to use the notion of *distributed cognition* (i.e., a cognitive system that is collective but includes not only persons but also instruments and other artifacts as parts of the cognitive system). In this new setting, scientists, technicians, machines, sensors, etc., interact harmoniously to achieve a final result. Obviously, this does not mean that machines and sensors are conscious. Instead, when the cognitive system is viewed as a whole, one may easily say that it is a computational system. But does it make sense to say that the ATLAS project is actually a computational project?

The whole project is not computational at all. First of all, when elementary particles interact, no symbolic representation is transformed by syntactic-like operations. And since computation is identified with the transformation of symbolic representations by syntax-manipulation operations, one easily deduces that elementary-particle interaction is not a computational process. Unfortunately, not everybody shares this idea. For instance, there are those who believe that even the whole universe is a gigantic computer that computes its next state. However, such beliefs are based on unjustified assumptions (see Section 8.5 for a more detailed discussion of these issues). But it is equally interesting to say that it is the beauty of computation in general and the "desire for a single, overarching explanation for everything" that has compelled many thinkers and researchers to support the idea that the universe is a computer. Nevertheless, the project is partially computational in the sense that there are computers that do actually compute. Thus, Project ATLAS is a hybrid system. There are some further questions related to the very nature of knowledge, but a proper treatment of such questions falls outside the scope of this book.