P	H	C	O	215	Operator: Wang Hui	Dispatch: 25.03.09	PE: Leah Gourley
Journal Name				Manuscript No.	Proofreader: Liu HongRong	No. of Pages: 18	Copy-editor: Christine Tsai

Philosophy Compass 4 (2009): 10.1111/j.1747-9991.2009.00215.x

Computationalism in the Philosophy of Mind

Gualtiero Piccinini*
University of Missouri – St. Louis

Abstract

Computationalism has been the mainstream view of cognition for decades. There are periodic reports of its demise, but they are greatly exaggerated. This essay surveys some recent literature on computationalism and reaches the following conclusions. Computationalism is a family of theories about the mechanisms of cognition. The main relevant evidence for testing computational theories comes from neuroscience, though psychology and AI are relevant too. Computationalism comes in many versions, which continue to guide competing research programs in philosophy of mind as well as psychology and neuroscience. Although our understanding of computationalism has deepened in recent years, much work in this area remains to be done.

1. Introduction

Computationalism is the view that intelligent behavior is causally explained by computations performed by the agent's cognitive system (or brain).[1] In roughly equivalent terms, computationalism says that cognition is computation. Computationalism has been mainstream in philosophy of mind – as well as psychology and neuroscience – for several decades.

Many aspects of computationalism have been investigated and debated in recent years. Several lessons are being learned: (1) computationalism is consistent with different metaphysical views about the mind, (2) computationalism must be grounded in an adequate account of computation, (3) computationalism provides a mechanistic explanation of behavior, (4) computationalism was originally introduced on the grounds of neurological evidence, (5) all computationalists (yes, even classicists) are connectionists in the most general sense, although not all connectionists are computationalists, and (6) which, if any, variety of computationalism is correct depends on how the brain works.

This essay is organized as follows. First, I will discuss the relationship between computationalism and the metaphysics of mind. Next, I will point out that the logical and explanatory strength of computationalism depends on which notion of computation it employs. Then, after taking a brief look at the origin of computationalism, I will list the main philosophical accounts of computation. With these preliminaries in place, I will introduce some influential varieties of computationalism, followed by a discussion of how computational explanation relates to intentional and mechanistic explanation. I will conclude with a list of arguments against and in favor of computationalism.

2. Computationalism and the Metaphysics of Mind

Computationalism is often said to come with a specific metaphysics of mind – functionalism. Sometimes, computationalism is even conflated with functionalism. Functionalism is a view about the nature of mental states, which says that mental states are functional states. 'Functional' is a term of art that can be explicated in different ways: to a first approximation, a functional state is a state defined in terms of some of its causes and effects. So functionalism maintains that the nature of mental states is given by how they fit in a network of causes and effects. By contrast, computationalism is not committed to any claim about the nature of mental states.

Surely computational theories of cognition are functional, in the broad sense that they explain behavior in terms of functionally defined states, processes, or mechanisms. It doesn't follow that if computationalism is true, functionalism is true; nor does it follow that if functionalism is true, then computationalism is true. To get functionalism from computationalism, we also need an additional assumption, such as that the nature of mental states is (entirely) computational. To get computationalism from functionalism, we also need the independent assumption that all functional states are computational. Both of these assumptions are controversial; neither is especially plausible.

In short, functionalism and computationalism are logically independent. Contrary to a widespread assumption, computationalism is consistent with a wide range of views about the metaphysics of mind, including but not limited to functionalism (Piccinini, 'Mind as Neural Software?').

Computationalism is also sometimes confused with the stronger view that mental states are computational states, in the sense that their nature is wholly (as opposed to partially or not at all) computational. This stronger view, which may be called 'computational functionalism', is not very popular, mostly because of worries about accommodating consciousness. Although there are philosophers who have given computational accounts of consciousness (Lycan; Dennett; Rey), many others doubt that simply being a computational state is enough for being a conscious state (e.g., Block). But rejecting the view that mental states are purely computational is consistent with the acceptance of computationalism: even if consciousness involves more than computation, computation may still explain behavior (in whole or in part).

3. Generic vs. Substantive Computationalism

Computationalism is usually introduced as an empirical hypothesis, open to disconfirmation. Whether computationalism has empirical bite depends on how we construe the notion of computation. The more inclusive a notion of computation, the weaker the version of computationalism formulated in its terms.

At one end of the continuum, some notions of computation are so loose that they encompass virtually everything. For instance, if computation is construed as the production of outputs from inputs and if any state of a system qualifies as an input or output, then every process is a computation.

A somewhat more stringent notion is that of information processing. Sometimes, computation is construed as information processing. The resulting version of computationalism is still weak. There is little doubt that organisms gather and process information about their environment. Processing information is surely an important aspect of cognition. Thus, if computation is information processing, then cognition involves computation. But this doesn't tell us much about how cognition works. In addition, the notions of information and computation in their most important uses are conceptually distinct, have different histories, are associated with different mathematical theories, and have different roles to play in a theory of cognition. It's best to keep them separate (Piccinini and Scarantino).

Computationalism becomes most interesting when it has explanatory power. The most relevant and explanatory notion of computation is that associated with digital computers. Computers perform impressive feats. They solve mathematical problems, play difficult games, prove logical theorems, etc. Perhaps cognitive systems work like computers. To a first approximation, this analogy between computers and cognitive systems is the original motivation behind computationalism. The resulting form of computationalism is a strong hypothesis, one that should be open to empirical testing. To understand it further, it will help to take a historical step back.

4. Turing vs. McCulloch and Pitts

Contrary to a popular belief, modern computationalism is not due to Alan Turing but to Warren McCulloch and Walter Pitts. This is not just a matter of historical priority. In recent years, our deepening appreciation of the history of both computability theory and computationalism has shed light on what computationalism says and how it may and may not be justified. Unfortunately, here there is only room for a few basic points (for entries into the historical literature, see Cordeschi; Shagrir, 'Gödel on Turing'; Sieg; Aizawa and Schlatter; Abramson).

What Turing did provide (together with Kurt Gödel, Alonzo Church, Emil Post, and Stephen Kleene, among others) is a precise notion of computation, defined in terms of a powerful body of new mathematics called 'computability theory' (Davis). In particular, Turing and Church defended the Church–Turing thesis, according to which the formalisms of computability theory, such as Turing machines, are sufficient to compute any function computable by algorithm. Assuming the Church–Turing thesis, Turing showed that there are universal Turing machines – machines that can compute *any* function computable by algorithm. To a first approximation, modern digital computers are universal in Turing's sense. (Strictly speaking, a universal machine must have an unbounded memory, whereas digital computer memories are not unbounded.)

Computationalism was first proposed in 1943 by McCulloch and Pitts, two neuroscientists of sorts. They were impressed with logical calculi, which could be used to formalize large bodies of knowledge (Whitehead and Russell). As Turing and other logicians had shown, logical calculi could be implemented in digital computing machines. McCulloch and Pitts argued that the brain embodies a logical calculus, which makes the brain a kind of digital computing machine. Their main motivation was to explain cognition. Their main evidence was that neurons send all-or-none signals, somewhat like digital switches. In other words, neurons send signals of fixed strength all of a sudden and in a very short time, as opposed to sending signals of strength that varies and increases or decreases gradually over time.

Although McCulloch and Pitts's theory was justified on neurophysiological grounds, it was not a realistic theory of neural activity, because it was based on severe simplifications and idealizations. But computationalism was the most serious and ambitious mechanistic explanation of cognition and intelligent behavior to date. (It was mechanistic in the sense that it explained behavior in terms of the activity and organization of the system's components.) Furthermore, computationalism came with a promising research program: to design computing machines that perform cognitive tasks. At first, computationalism did not encounter much favor among neuroscientists. Instead, by the 1960s it became popular in artificial intelligence, psychology, and philosophy of mind. By the 1980s, computationalism established itself in neuroscience too, although it's not clear that what most neuroscientists mean by 'computation' is the same as what most psychologists and computer scientists mean by it.

5. What is Computation?

What makes a concrete system into a computing system? Philosophers have offered three main accounts.

The mapping account says, roughly, that a computing system is a concrete system such that there is a computational description that maps onto a physical description of the system. (A computational description is a description that assigns a system different sequences of computational states under different conditions.) If any mapping is acceptable, it can be shown that almost every physical system implements every computation (Putnam, *Representation and Reality*; Searle, *Rediscovery of Mind*). This trivialization result can be avoided by putting appropriate restrictions on acceptable mappings; for instance, legitimate mappings must respect causal relations between physical states (Chrisley; Chalmers; Copeland; Scheutz).

Still, there remain mappings between (many) computational descriptions and any physical system. Under the mapping account, everything performs at least some computations. This still strikes some as a trivialization of computationalism. Furthermore, it doesn't do justice to computer science, where only relatively few systems count as performing computations. Those who want to restrict the notion of computation further have to look beyond the mapping account of computation.

The semantic account is perhaps the most popular in philosophy of mind. It says that computation requires representation: only processes that manipulate the appropriate kind of representation in the appropriate way count as computations (e.g., Pylyshyn). Since cognitive systems and digital computers are generally assumed to manipulate representations, they compute. Since most other systems are generally assumed *not* to manipulate (the relevant kind of) representations, they do not compute. Thus, the semantic account appears to accommodate some common intuitions about what does and does not count as a computing system. For many years, many philosophers of mind took the semantic account for granted. The main debate was between those who defended an externalist semantics of computational states (e.g., Burge; Shapiro) and those who defended an internalist semantics (e.g., Segal; Egan, 'In Defence of Narrow Mindedness').[2]

And yet, semantic accounts have problems too. First, computation is discussed within physical theory. For instance, some physicists debate whether all physical processes are computational and what the computational power of physical systems is (e.g., Penrose; Wolfram). Since most physical systems do not manipulate representations, the semantic account of computation seems ill-suited for that debate. Second, specifying the kind of representation and representational manipulation that is relevant to computation seems to require a non-semantic way of individuating computations (Piccinini, 'Functionalism, Computationalism, and Mental Contents'). Finally, representation does not seem to be presupposed by the notion of computation employed in computability theory and computer science (Piccinini, 'Computation without Representation'). Undaunted by these objections, many philosophers of mind are busy defending and developing semantic accounts of computation (Shagrir, 'Why We View the Brain as a Computer'; Horowitz; Sprevak).

Finally, there is the 'mechanistic account' (Piccinini, 'Computing Mechanisms'). It says that computations are a special kind of process, defined in part by the kind of vehicle being manipulated (i.e., strings of discrete states). Only the relatively few systems that manipulate the appropriate vehicles according to rules defined over them count as computing systems.[3]

A common objection is that the mechanistic account leaves out so called 'analog computation'. 'Analog computation' is another term of art that can mean a number of things. In its most precise sense, 'analog computation' is the manipulation of continuous variables to solve differential equations (Pour-El, 'Abstract Computability'). The objection from analog computation begs the question of whether analog computation and digital computation have enough in common to be given a single account. Since they are quite different kinds of processes, this is doubtful. At any rate, analog computation has been given its own mechanistic account in terms of the appropriate kind of vehicle (i.e., continuous variables) (Piccinini, 'Computers').

The mechanistic account can be used to distinguish different kinds of computing systems – ordinary calculators, programmable and non-programmable computers, etc. – based on the components they have and the kind of string manipulations they can perform. The account can then be used to classify different versions of computationalism depending on which type of computing mechanism they postulate cognitive systems to be. There are versions of computationalism according to which cognitive systems are networks of simple processing units (Rumelhart and McClelland; Churchland and Sejnowski), finite state automata (Nelson), programmable computers (Devitt and Sterelny), programmable computers that store programs in memory (Fodor, *Language of Thought*), computers that are universal in Turing's sense (Newell and Simon), and computers that are even more powerful than universal Turing machines (Copeland, 'Wide Versus Narrow Mechanism').

6. Classicism, Connectionism, and Beyond

Until the early 1980s, it was commonly assumed that computationalism (i) is committed to the existence of a language of thought (i.e., the idea that cognition is the manipulation of linguistic, or sentence-like, structures), and (ii) has little or nothing to do with the brain (i.e., computationalism tells us little or nothing about how the brain works and how the brain works tells us little or nothing about whether computationalism is true: neural descriptions and computational descriptions are at 'different levels'). During the 1980s, connectionism re-emerged as an influential approach to psychology. Most connectionists deny that cognition is based on a language of thought and affirm that a theory of cognition should be at least 'inspired' by the way the brain works (Rumelhart and

McClelland). While classical computationalists explain cognition by reference to linguistic structures, connectionists explain cognition in terms of neural networks (i.e., networks of simple processing units, somewhat like the networks of neurons found in the brain). The resulting debate (e.g., Fodor and Pylyshyn; Smolensky) has been somewhat confusing.

Part of what's confusing is that different authors employ different notions of computation, which vary in both their degree of precision and their inclusiveness. But even after we factor out differences in notions of computation, clarification is needed.

Given the apparent conflict between (i) and (ii) on one side and claims by connectionists on the other, many get the impression that computationalism and connectionism are mutually exclusive. But many connectionists also maintain that neural networks perform *computations* and such computations explain behavior, so these connectionists should be counted among the computationalists. To make matters worse, other connectionists reject computationalism – they maintain that their neural networks, while explaining behavior, do not do so by performing computations. Furthermore, we have already seen that (ii) is false: computationalism was initially introduced as a theory of the brain, and it (or at least something that sounds like it) is now a working assumption of many neuroscientists. In addition, (i) is overly restrictive: computations need not manipulate linguistic structures.

To clarify this debate, we need two separate distinctions. One is the distinction between computationalism ('cognition is computation') and its denial ('cognition is something other than computation'). The other is the distinction between classicism ('cognition is computation over linguistic structures') and connectionism ('cognition is what neural networks do'). We then have two versions of computationalism – the classical one and the connectionist one ('cognition is neural network computation') – standing opposite to the denial of computationalism, which includes the anti-computationalist version of connectionism ('cognition is neural network processing, which is something other than computation'). This may be enough to accommodate most views in the current debate. But it still doesn't do justice to the relationship between classicism and connectionism.

Part of the difficulty derives from the ambiguity of the term 'connectionism'. In its original sense, connectionism says that behavior is explained by the changing connections between stimuli and responses, which are biologically mediated by changing connections between neurons (Thorndike; Hebb). This original connectionism – which is similar to associationism but adds to it a biological mechanism to explain the associations – influenced contemporary connectionism, which is often confused with it (and with associationism). But contemporary connectionism is a different view. In its most general form, contemporary connectionism simply says that behavior is explained (at some level) by neural network activity. But this is a truism – or at least it should be. The brain is the organ of cognition, the cells that perform cognitive functions are (mostly) the neurons, and neurons perform their cognitive labor by organizing themselves in networks. Modern connectionism is a platitude.

The confusion arises because many contemporary connectionists are also connectionists in the original, associationist sense. So classicists object to connectionism in the associationist sense, while connectionists insist that the brain is made out of neural networks. Of course it is, but this does not refute classicism. What remains to be determined is which neural networks, organized in which way, actually explain cognition.

To sum up, everyone is (or should be) a connectionist in the most general contemporary sense, though not everyone is a connectionist in the associationist sense. Some people are classicists, believing that in order to explain cognition, neural networks must amount to manipulators of linguistic structures. This view is sometimes called 'implementational connectionism'. Some people are non-classicist (but still computationalist) connectionists, believing that cognition is explained by non-classical neural network computation. Finally, some people are anti-computationalist connectionists, believing that cognition is explained by neural network processes, but these do not amount to computation (e.g., because they process the wrong kind of vehicles). To find out which view is correct, in the long run the only effective way is to study nervous systems at all levels of organization and find out how they produce behavior (Fig. 1; for more details and references, see Piccinini, 'Some Neural Networks').

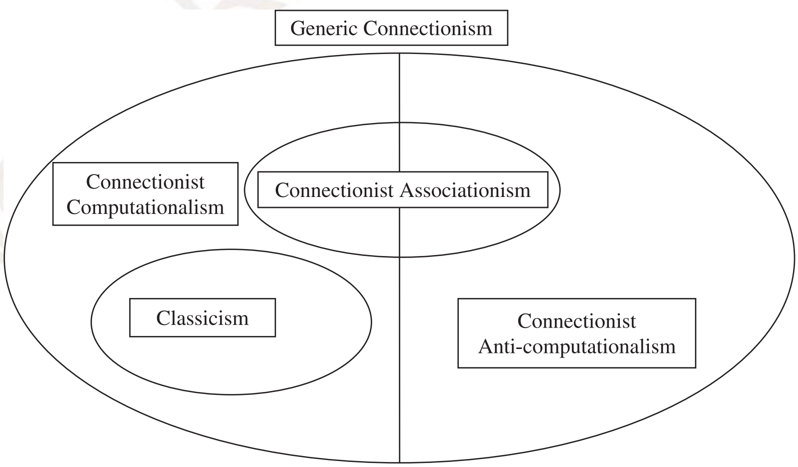

Fig. 1. Some prominent forms of computationalism and anti-computationalism and their relations.

7. *Levels of Explanation*

In our everyday life, we explain behavior in terms of our attitudes towards the world: what we believe, desire, etc. These attitudes are partially individuated by what they are about – their intentional content. Thus, the explanatory style employed in our folk psychology is often called 'intentional explanation'. For example, you might say that 'Lori went to the store *because she was craving chocolate and she believed the store was the source of chocolate nearest to her*'.

Intentional explanation is common in psychology and cognitive neuroscience too. When faced with a cognitive explanandum, e.g., 'why did Lori go to the store?' scientists generally postulate a system of internal states – mental representations – individuated in part by what they represent. For example, Lori's brain contains something that represents chocolate, which under the circumstances guides Lori's behavior so that she seeks chocolate. How this 'scientific' version of intentional explanation relates to its folk psychological counterpart is under dispute.

Some philosophers maintain that folk psychology postulates internal representations too: it's a proto-scientific explanation (Sellars). The question then arises as to what these representations amount to. Are the mental representations postulated by scientists going to vindicate those postulated by the folk? Some say yes (e.g., Fodor, *Psychosemantics*), others say no (e.g., Churchland).

Yet other philosophers maintain that folk psychological intentional explanations do not, in fact, postulate internal representations. Simulation theorists claim that intentional explanations are based on the simulation of one mind by another. According to *some* simulation theorists, simulation does *not* involve ascribing mental representations (Gordon, 'Radical Simulation'; 'Simulation and Reason Explanation'). If this alternative account of folk psychology is correct, then the vindication of folk psychology by science is beside the point, because folk psychological intentional explanation is not a proto-scientific explanation.

Regardless of folk psychology, there is a further question about intentional explanation: what is the status of *scientific* intentional explanation? How does it relate to computational explanation? Computational explanations explain behavior in terms of a computational process. Such a computational process is generally assumed to manipulate internal representations with the relevant intentional content – the content postulated by intentional explanations. If so, scientific intentional explanations can be embedded within computational ones – the computations add a physically implementable process such that the system can exhibit the desired behavior by manipulating the relevant representations. But where does the intentionality of the representations come from?

On intentionality, there is a wide range of opinions. According to anti-naturalists, intentionality needs no (naturalistic) explanation – the intentional character of computational explanation simply shows that psychology and perhaps neuroscience are sui generis sciences. By contrast, anti-realists maintain that intentionality is not a real feature of the mind, although ascriptions of intentional states may be heuristically useful (Egan, 'Modest Role for Content'). Finally, naturalists look for explanations of intentionality in non-intentional terms (see Rupert for a review).

A further question pertains to the relationship between computational and mechanistic explanation. The traditional view is that these two types of explanation are pitched at two independent levels: computations are the domain of psychologists, while the implementing neural mechanisms are the business of neuroscientists (cf. Marr). This view has been criticized as unfaithful to scientific practices. It's been pointed out that (i) both psychologists and neuroscientists offer computational explanations (Piccinini, 'Computational Explanation'), (ii) far from being independent, different levels of explanation constrain one another (e.g., Feest), and (iii) both computational explanations (Churchland and Sejonowski) and mechanistic explanations (Craver) can be given at different levels.

One alternative to the traditional view is the mechanistic account of computation. According to it, computational explanation is just one type of mechanistic explanation. Mechanistic explanations provide components with such properties and organization that they produce the explanandum. Computational explanation, then, is explanation in terms of computing mechanisms and components – mechanisms and components that perform computations. (Computation, in turn, is manipulation of appropriate vehicles – strings of digits.) Mechanistic explanations come with many levels of mechanisms, where each level is constituted by its components and the way they are organized. If a mechanistic level produces its behavior by the action of computing components, it counts as a computational level. Thus, a mechanism may contain zero, one, or many computational levels depending on what components it has and what they do.

More work remains to be done to spell out in detail how different levels of explanation relate to one another, and how computational explanations in psychology and neuroscience relate to one another as well as other kinds of explanation.

8. *Objections*

Computationalism has accumulated a large number of objections, none of which are conclusive. I will mention some prominent ones.

8.1. THE MATHEMATICAL OBJECTION

As Turing showed, there is a precise limit to the range of theorems that any (fixed) computing machine can prove. But as Turing also pointed out, mathematicians are capable of inventing new methods of proof, so they can prove more theorems than any (fixed) computing machine. Therefore, the objection concludes, the cognitive systems of human mathematicians are not computing machines (Turing).

Some people have taken the mathematical objection a step further. Rather than limiting themselves to the claim that cognition involves something beyond computation, they argue that cognition involves hypercomputations – that is, computations more powerful than those carried out by Turing machines (Siegelmann; Copeland, 'Wide Versus Narrow Mechanism'; Bringsjord and Arkoudas). In a sense, this is still a version of computationalism – it just says that the computations performed by cognitive systems are more powerful than those of Turing machines. But at any rate, there is no hard evidence that human beings can hypercompute: no one has shown how to solve a genuinely Turing-uncomputable problem, such as the halting problem,[4] by using human cognitive faculties. Therefore, for now, all we need to answer is Turing's original mathematical objection.

As Turing pointed out, computing machines that change their programs over time and are allowed to make mistakes are not limited in the way that fixed machines are. They can enlarge the range of methods they can employ and the theorems they can prove. Therefore, the cognitive systems of human mathematicians can still be computing machines, so long as they change over time (in a non-computable way) and can make mistakes. Since this last claim is independently plausible, computationalism escapes the mathematical objection (Piccinini, 'Alan Turing and the Mathematical Objection').

8.2. REPRESENTATION/INTENTIONALITY

Cognition involves intentionality (i.e., cognitive states are *about* something), but computation is insufficient for intentionality. Therefore, cognition is not computation. A well known variant of this objection claims that computation is insufficient for 'understanding' (Searle, 'Minds, Brains, and Programs'). One response is that cognition does not involve intentionality. According to this response, explaining behavior does not require the postulation of mental representations or similarly intentional notions. Another response is that computation does suffice for intentionality. The semantic view of computation may seem to help here. For if computation presupposes representation in a robust enough sense ('original intentionality'), then computation is automatically intentional. The problem with this second response is that computation per se hardly presupposes the right notion of representation ('original intentionality'). Most intentional realists prefer a third response: they accept that computation is insufficient for intentionality but maintain that computation is still the process that explains behavior. According to them, cognition is *intentional* computation. Where does the intentionality come from? As we saw above, some of them take intentionality as a primitive or give it a non-reductive account, while others offer naturalistic accounts of intentionality. Most of today's computationalists seem to accept that a complete explanation of cognitive phenomena requires not only a computational explanation of behavior, but also an account of intentionality.

8.3. ANTI-REPRESENTATIONALISM

Computationalism presupposes that cognition manipulates representations, but cognition does no such thing. Therefore, cognition is not computation (cf. van Gelder, 'What Might Cognition Be'). This objection reverses the previous one, and it's a nonstarter. There are two reasons. First, computationalism can be formulated without *presupposing* representationalism, and in my opinion it should be. Second, representations are a staple of mainstream psychology and neuroscience. They are not likely to be eliminated from our explanations of cognition (pace Ramsey).

8.4. CONSCIOUSNESS

Cognition involves consciousness, but computation is insufficient for consciousness. Therefore, cognition is not computation. Actually, whether cognition involves consciousness is controversial. Some philosophers maintain that consciousness is epiphenomenal and that cognition can be explained without involving consciousness. If so, then the consciousness objection is defused. But even if cognition involves consciousness, there are several attempts to explain consciousness in broadly computational terms (e.g., Lycan; Dennett; Rey). Finally, even if cognition involves consciousness and consciousness requires something beyond computation, it doesn't follow that computation is not an important part of the explanation of cognition.

8.5. EMBODIED AND EMBEDDED COGNITION

Cognition is embodied (i.e., coupled with a body) and embedded (i.e., coupled with the environment), while computation is disembodied and unembedded. Therefore, cognition is not computation (cf. Thompson). This objection is based on a false premise. Computation *can* be disembodied and unembedded, but it *need not* be. Computing systems can be coupled with a body, an environment, or both. For any substantive version of the thesis that cognition is embodied or embedded (i.e., any version that does not build anti-computationalism into its definition), it is possible to postulate that computations are at least part of the cognitive processes. And for those with a taste for extended cognition (i.e., cognitive systems some of whose parts are in the agent's environment), Robert Wilson has proposed a version of computationalism according to which the computations themselves extend into the environment.

8.6. Dynamics

Cognition is dynamical, not computational (cf. van Gelder, 'Dynamical Hypothesis'). This is a false contrast. A dynamical system is a system that changes over time in a way that depends on its state at any given time. Dynamical systems are usefully modeled using systems of differential equations or difference equations, and the study of such equations is called 'dynamical systems theory'. Cognitive systems, of course, are dynamical systems – they change over time in a way that depends on their state at any given time. Thus, some scientists employ differential equations to study cognitive systems. By the same token, computational systems are dynamical systems too, and they are often studied using differential or difference equations. Thus, there is no opposition between the claim that cognition is dynamical and the claim that cognition is computation. From this, however, it doesn't follow that cognition is computation. It remains to be seen whether the dynamics of cognition (or at least the dynamics of natural cognizers) are computational.

9. Arguments for Computationalism

Many arguments for computationalism have been offered, none of which are conclusive. Here are some important ones.

9.1. From Functionalism

It is sometimes assumed that computationalism is a consequence of functionalism. As per Section 2, this is not the case.

9.2. From Pancomputationalism

Some authors maintain that the cognitive system is computational because everything can be described computationally (Putnam, 'Psychological Predicates'). In my opinion, this view trivializes computationalism. Its conclusion is not the explanatory version of computationalism that most computationalists intend. In addition to being irrelevant to the philosophy of mind, pancomputationalism is true only in a fairly trivial and uninteresting sense (Piccinini, 'Computational Modeling vs. Computational Explanation').

9.3. FROM THE CHURCH–TURING THESIS

Several authors have argued that computationalism is a consequence of the Church–Turing thesis (e.g., Baum). These arguments are based either on a misunderstanding of the Church–Turing thesis or on fallacious reasoning (Tamburrini; Copeland, 'Wide Versus Narrow Mechanism'; Piccinini, 'Computationalism, the Church–Turing Thesis, and the Church–Turing Fallacy'). The Church–Turing thesis is about the boundaries of what can be computed by algorithm. It does not entail anything about cognition unless it's independently established that all cognitive processes are algorithmic computations.

9.4. FROM THE ALL-OR-NONE CHARACTER OF NERVOUS ACTIVITY

McCulloch and Pitts argued that since neuronal signals are all-or-none, similar to those of digital devices, the brain is a computing device. But the analogy between neurons and digital switches is far from exact. The all-or-none character of nervous activity alone does not establish that the brain is a computing system (Piccinini, 'First Computational Theory').

9.5. FROM INFORMATION PROCESSING

Computing systems manipulate complex combinatorial structures. Such structures can be semantically interpreted, and computations can be set up so that they respect the semantic properties of computational states. For instance, there are computations that derive valid logical inferences. Accordingly, it's been argued that computationalism explains how human beings (and perhaps other animals) can manipulate representations so as to respect their semantic properties. This is one flavor of the argument from information processing.

The argument from information processing comes in different flavors, which support different versions of computationalism. An especially weak version of the argument simply defines computation as information processing: since it is independently plausible that cognition involves the processing of information, it then follows that cognition involves computation in this sense. But as we saw in Section 3, this does nothing to explain how such information processing is carried out. By contrast, a particularly strong version of the argument from information processing proceeds from the premise that cognition involves the processing of linguistic representations. Such an argument can then be used to defend a classical version of computationalism against not only anti-computationalism, but also non-classical computationalism (Fodor and Pylyshyn; Aizawa). Whether any argument from information processing is successful depends on how information processing is ultimately to be explained – a complex question with many ramifications in philosophy as well as the sciences of mind.

9.6. FROM COGNITIVE FLEXIBILITY

Human beings are cognitively flexible: they can solve an indefinite number of problems and learn an indefinite range of behaviors. How do they do it? Consider computers. Computers are the most flexible artifacts by far. They can do mathematical calculations, derive logical theorems, play board games, recognize objects, control robots, and even engage in somewhat crude conversations. They can do this because they can execute different sets of instructions designed for different tasks. In Turing's terms, they approximate universal machines. Perhaps human beings are cognitively flexible because, like computers, they possess a general purpose processing mechanism that executes different instructions for different tasks (Fodor, 'Appeal to Tacit Knowledge'; Newell; Samuels). The argument from cognitive flexibility is one of the most powerful, because there is no well worked-out alternative explanation of cognitive flexibility, at least for high level cognitive skills such as problem solving.[5] Notice that this argument supports not computationalism in general, but more specifically a strong classical version of computationalism – one according to which there is a strong analogy between cognitive systems and digital computers.

10. Conclusion

Computationalism is a family of theories about the mechanisms of cognition. The main relevant evidence for testing it comes from neuroscience, though psychology and AI are relevant too. Computationalism comes in many versions, which continue to guide competing research programs in philosophy of mind as well as psychology and neuroscience. Although our understanding of computationalism has deepened in recent years, much work in this area remains to be done.

Acknowledgments

Thanks to several people – especially Ken Aizawa, Brie Gertler, Corey Maley, Eric Thomson, and Anna-Mari Rusanen – for comments and suggestions, and to the University of Missouri Research Board for research support.

Short Biography

Gualtiero Piccinini works primarily in philosophy of mind, with an eye to psychology, neuroscience, and computer science. His main current interests include computationalism, the relation between mind and brain, intentionality, and consciousness. His articles have been published in *Philosophy and Phenomenological Research*, *Philosophy of Science*, *Australasian Journal of Philosophy*, *Philosophical Studies*, *Neural Networks*, *Synthese*, *Canadian Journal of Philosophy*, *Studies in the History and Philosophy of Science*, *Journal of Consciousness Studies*, and *Minds and Machines*. In 2003, he graduated from the department of History and Philosophy of Science at the University of Pittsburgh. Between 2003 and 2005, he was a James S. McDonnell Post Doctoral Research Fellow in the Philosophy-Neuroscience-Psychology Program at Washington University in St. Louis. Since 2005, he is a member of the Philosophy Department at the University of Missouri – St. Louis.

Notes

* Correspondence address: 599 Lucas Hall (MC 73), 1 University Blvd., St. Louis, Missouri, United States, 63121. Email: piccininig@umsl.edu.

[1] Opinions differ as to how cognition relates to the brain. For simplicity, I will drop most references to the brain in what follows. I do not mean to take a stance on the mind-brain relation.

[2] Egan's view is different from standard semantic accounts, because she individuates computational states in terms of mathematical contents rather than cognitive contents.

[3] Mechanistic and mapping accounts are consistent with (some) computations (including cognitive ones) processing representations, but, unlike semantic accounts, they don't *define* computation as the processing of representations.

[4] The halting problem is the problem of determining whether any given Turing machine will ever halt while computing on any given input. There is no Turing machine that solves the halting problem.

[5] Even animals exhibit a degree of cognitive flexibility that has been argued to require at least some features of a digital computer, such as a read-write memory (Gallistel).

Works Cited

Abramson, Darren. 'Turing's Responses to Two Objections'. *Minds and Machines* 18 (2008): 147–67.

Aizawa, Kenneth. *The Systematicity Arguments*. Boston, MA: Kluwer, 1993.

—— and Mark Schlatter. 'Walter Pitts and "A Logical Calculus" '. *Synthese* 162 (2007): 235–50.

Baum, Eric B. *What is Thought?* Cambridge, MA: MIT Press, 2004.

Bringsjord, Selmer and Konstantine Arkoudas. 'On the Provability, Veracity, and AI-Relevance of the Church-Turing Thesis'. *Church's Thesis After 70 Years*. Eds. Adam Olszewski, Jan Wolenski, and Robert Janusz. Heusenstamm: Ontos, 2007.

Burge, Tyler. 'Individualism and Psychology'. *Philosophical Review* 95 (1986): 3–45.

Chalmers, David J. 'Does a Rock Implement Every Finite-State Automaton?'. *Synthese* 108 (1996): 310–33.

Churchland, Patricia S. and Terrence J. Sejnowski. *The Computational Brain*. Cambridge, MA: MIT Press, 1992.

Churchland, Paul M. *Neurophilosophy at Work*. Cambridge: Cambridge UP, 2007.

Copeland, B. Jack. 'Narrow Versus Wide Mechanism: Including a Re-Examination of Turing's Views on the Mind-Machine Issue'. *The Journal of Philosophy* 96 (2000): 5–32.

——. 'What is Computation?'. *Synthese* 108 (1996): 224–359.

Cordeschi, Roberto. *The Discovery of the Artificial: Behavior, Mind and Machines Before and Beyond Cybernetics*. Dordrecht: Kluwer, 2002.

Craver, Carl F. *Explaining the Brain*. Oxford: Oxford UP, 2007.
Davis, Martin. *The Undecidable*. Ewlett, NY: Raven, 1965.
Dennett, Daniel C. *Consciousness Explained*. Boston, MA: Little, Brown and Company, 1991.
Devitt, Michael and Kim Sterelny. *Language and Reality: An Introduction to the Philosophy of Language*. Cambridge, MA: MIT Press, 1999.
[4] Egan, Frances. 'In Defence of Narrow Mindedness'. *Mind and Language* 14 (1999): 177–94.
——. 'A Modest Role for Content'. *Studies in the History and Philosophy of Science* (forthcoming).
Feest, Uljana. 'Functional Analysis and the Autonomy of Psychology'. *Philosophy of Science* 70 (2003): 937–48.
Fodor, Jerry A. 'The Appeal to Tacit Knowledge in Psychological Explanation'. *Journal of Philosophy* 65 (1986): 627–40.
——. *The Language of Thought*. Cambridge, MA: Harvard UP, 1975.
——. *Psychosemantics*. Cambridge, MA: MIT Press, 1987.
—— and Zenon W. Pylyshyn. 'Connectionism and Cognitive Architecture'. *Cognition* 28 (1988): 3–71.
[5] Gallistel, Charles G. 'Learning and Representation'. *Learning and Memory: A Comprehensive Reference*. Eds. R. Menzel and J. Byrne. New York, NY: Elsevier, forthcoming.
van Gelder, Timothy. 'The Dynamical Hypothesis in Cognitive Science'. *Behavioral and Brain Sciences* 21 (1998): 615–65.
——. 'What Might Cognition Be, if not Computation?' *The Journal of Philosophy* 92 (1995): 345–81.
[6] Gordon, Robert. 'Radical Simulation'. *Theories of Theories of Mind*. Eds. P. Carruthers and P. Smith. Cambridge: Cambridge UP, 1996.
——. (2001). 'Simulation and Reason Explanation: The Radical View'. *Philosophical Topics* 29 (2001): 175–92.
Hebb, Donald O. *The Organization of Behavior: A Neuropsychological Theory*. New York, NY: Wiley, 1949.
Horowitz, Amir. 'Computation, External Factors, and Cognitive Explanations'. *Philosophical Psychology* 20 (2007): 65–80.
Lycan, William. *Consciousness*. Cambridge, MA: MIT Press, 1987.
Marr, David. *Vision*. New York, NY: Freeman, 1982.
Nelson, Raymond J. *The Logic of Mind*. Dordrecht: Reidel, 1982.
Newell, Allen. *Unified Theories of Cognition*. Cambridge, MA: Harvard UP, 1990.
—— and Herbert A. Simon. 'Computer Science as an Empirical Enquiry: Symbols and Search'. *Communications of the ACM* 19 (1976): 113–26.
Penrose, Roger. *Shadows of the Mind*. Oxford: Oxford UP, 1994.
Piccinini, Gualtiero. 'Alan Turing and the Mathematical Objection'. *Minds and Machines* 13 (2003): 23–48.
——. 'Computation without Representation'. *Philosophical Studies* 137 (2008): 205–41.
——. 'Computational Explanation in Neuroscience'. *Synthese* 153 (2006): 343–53.
——. 'Computational Modeling vs. Computational Explanation: Is Everything a Turing Machine, and Does It Matter to the Philosophy of Mind?'. *Australasian Journal of Philosophy* 85 (2007): 93–115.
——. 'Computationalism, the Church-Turing Thesis, and the Church-Turing Fallacy'. *Synthese* 154 (2007): 97–120.
——. 'Computers'. *Pacific Philosophical Quarterly* 89 (2008): 32–73.
——. 'Computing Mechanisms'. *Philosophy of Science* 74 (2007): 501–26.
——. 'The First Computational Theory of Mind and Brain: A Close Look at McCulloch and Pitts's "Logical Calculus of Ideas Immanent in Nervous Activity" '. *Synthese* 141 (2004): 175–215.
——. 'Functionalism, Computationalism, and Mental Contents'. *Canadian Journal of Philosophy* 34 (2004): 375–410.
[7] ——. 'The Mind as Neural Software? Understanding Functionalism, Computationalism, and Computational Functionalism'. *Philosophy and Phenomenological Research* (forthcoming).
——. 'Some Neural Networks Compute, Others Don't'. *Neural Networks* 21 (2008): 311–21.
[8] —— and Andrea Scarantino. 'Computation vs. Information Processing: How They Are Different and Why It Matters'. *Studies in the History and Philosophy of Science* (forthcoming).

Pour-El, Marian B. ‘Abstract Computability and Its Relation to the General Purpose Analog Computer (Some Connections between Logic, Differential Equations and Analog Computers)’. *Transactions of the American Mathematical Society* 199 (1974): 1–28.

9 Putnam, Hilary. ‘Psychological Predicates’. *Art, Philosophy, and Religion*. Pittsburgh, PA: U of Pittsburgh P, 1967.

——. *Representation and Reality*. Cambridge, MA: MIT Press.

Pylyshyn, Zenon W. *Computation and Cognition*. Cambridge, MA: MIT Press, 1984.

Rey, Georges. ‘Mind, Intentionality and Inexistence: An Overview of My Work’. *Croatian Journal of Philosophy* 5 (2005): 389–415.

Rumelhart, David E., James M. McClelland, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press, 1986.

Rupert, Robert D. ‘Causal Theories of Mental Content’. *Philosophy Compass* 3 (2008): 353–80.

10 Samuels, Richard. ‘Classical Computationalism and the Many Problems of Cognitive Relevance’. *Studies in the History and Philosophy of Science* (forthcoming).

Searle, John R. ‘Minds, Brains, and Programs’. *The Behavioral and Brain Sciences* 3 (1980): 417–57.

——. *The Rediscovery of the Mind*. Cambridge, MA: MIT Press, 1992.

Segal, Gabriel. ‘Defence of a Reasonable Individualism’. *Mind* 100 (1991): 485–93.

11 Sellars, Wilfrid. ‘Empiricism and the Philosophy of Mind’. *Minnesota Studies in the Philosophy of Science, Vol. 1, The Foundations of Science and the Concepts of Psychology and Psychoanalysis*. Eds. Herbert Feigl and Michael Scriven. Minneapolis: U of Minnesota Press, 1956.

Shagrir, Oron. ‘Why We View the Brain as a Computer’. *Synthese* 153 (2006): 393–416.

——. ‘Gödel on Turing on Computability’. *Church’s Thesis after 70 Years*. Eds. Adam Olszewski, Jan Wolenski, and Robert Janusz. Heusenstamm: Ontos, 2007.

Shapiro, Lawrence A. ‘Content, Kinds, and Individualism in Marr’s Theory of Vision’. *The Philosophical Review* 102 (1993): 489–513.

12 Sieg, Wilfried. ‘Computability Theory’. *Handbook of the Philosophy of Science: Philosophy of Mathematics*. Ed. Andrew Irvine. Elsevier, 2006.

Siegelmann, Hava T. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Boston, MA: Birkhäuser, 1999.

Smolensky, Paul. ‘On the Proper Treatment of Connectionism’. *Behavioral and Brain Sciences* 11 (1988): 1–23.

13 Sprevak, Mark. ‘Computation, Individuation, and the Representation Condition’. *Studies in the History and Philosophy of Science* (forthcoming).

14 Tamburrini, Guglielmo. ‘Mechanistic Theories in Cognitive Science: The Import of Turing’s Thesis’. *Logic and Scientific Methods*. Eds. Maria Luisa Dalla Chiara, K. Doets, Daniele Mundici, and J. van Benthem. Kluwer, 1997. 239–57.

Thompson, Evan. *Mind in Life: Biology, Phenomenology, and the Sciences of Mind*. Cambridge, MA: Harvard UP, 2007.

Thorndike, E. L. *The Fundamentals of Learning*. New York, NY: Teachers College, Columbia U, 1932.

Turing, Alan M. ‘Computing Machinery and Intelligence’. *Mind* 59 (1950): 433–60.

Whitehead, Alfred N. and Bertrand Russell. *Principia Mathematica*. Cambridge: Cambridge UP, 1910–13.

Wilson, Robert A. ‘Wide Computationalism’. *Mind* 103 (1994): 351–72.

Wolfram, Stephen. *A New Kind of Science*. Champaign, IL: Wolfram Media, 2002.

Author Query Form

Journal: Philosophy Compass

Article: phco_215.fm

Dear Author,

During the copy-editing of your paper, the following queries arose. Please respond to these by marking up your proofs with the necessary changes/additions. Please write your answers on the query sheet if there is insufficient space on the page proofs. Please write clearly and follow the conventions shown on the attached corrections sheet. If returning the proof by fax do not write too close to the paper's edge. Please remember that illegible mark-ups may delay publication.

Many thanks for your assistance.

No.	**Query**	**Remarks**
1	Author: Please provide the reference of Block in the Works Cited.	
2	Author: Please provide the reference of Chrisley in the Works Cited.	
3	Author: Please provide the reference of Scheutz in the Works Cited.	
4	Author: Please provide update (if any).	
5	Author: Please provide update (if any).	
6	Author: Please provide page range.	
7	Author: Please provide update (if any).	
8	Author: Please provide update (if any).	

No.	Query	Remarks
9	Author: Please provide page range.	
10	Author: Please provide update (if any).	
11	Author: Please provide page range.	
12	Author: Please provide place and page range.	
13	Author: Please provide update (if any).	
14	Author: Please provide place.	

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	• • • under matter to remain	Ⓥ
Insert in text the matter indicated in the margin	⋏	New matter followed by ⋏ or ⋏⊗
Delete	/ through single character, rule or underline or ⊢—⊣ through all characters to be deleted	ơ/ or ơ/⊗
Substitute character or substitute part of one or more word(s)	/ through letter or ⊢—⊣ through characters	new character / or new characters /
Change to italics	— under matter to be changed	⌣
Change to capitals	≡ under matter to be changed	≡
Change to small capitals	= under matter to be changed	=
Change to bold type	∼ under matter to be changed	∼
Change to bold italic	≂ under matter to be changed	≂
Change to lower case	Encircle matter to be changed	
Change italic to upright type	(As above)	⊥
Change bold to non-bold type	(As above)	⊥
Insert 'superior' character	/ through character or ⋏ where required	Y or ⋏ under character e.g. Y or ⋏
Insert 'inferior' character	(As above)	⋏ over character e.g. ⋏
Insert full stop	(As above)	⊙
Insert comma	(As above)	,
Insert single quotation marks	(As above)	Y or ⋏ and/or Y or ⋏
Insert double quotation marks	(As above)	Y or ⋏ and/or Y or ⋏
Insert hyphen	(As above)	⊢⊣
Start new paragraph	⌐	⌐
No new paragraph	↩	↩
Transpose	⊔⊓	⊔⊓
Close up	linking ⁀ characters	⁀
Insert or substitute space between characters or words	/ through character or ⋏ where required	Y
Reduce space between characters or words	| between characters or words affected	⊤