Synthese (2006) 153:393–416
DOI 10.1007/s11229-006-9099-8

ORIGINAL PAPER

Why we view the brain as a computer

Oron Shagrir

Received: 6 July 2006 / Accepted: 8 August 2006 /
Published online: 20 October 2006

Abstract The view that the brain is a sort of computer has functioned as a theoretical guideline both in cognitive science and, more recently, in neuroscience. But since we can view every physical system as a computer, it has been less than clear what this view amounts to. By considering in some detail a seminal study in computational neuroscience, I first suggest that neuroscientists invoke the computational outlook to explain regularities that are formulated in terms of the information content of electrical signals. I then indicate why computational theories have explanatory force with respect to these regularities: in a nutshell, they underscore correspondence relations between formal/mathematical properties of the electrical signals and formal/mathematical properties of the represented objects. I finally link my proposal to the philosophical thesis that content plays an essential role in computational taxonomy.

Keywords Computation · Content · Information · Explanation

A central working hypothesis in cognitive and brain sciences is that the brain is a sort of a computer. But what, exactly, does it mean to say that an organ or a system such as the brain is a computer? And why do scientists take a computational approach to brain and cognitive function? In addressing these questions, I will put forward a revisionary account of computation that makes two radical claims. First, that everything can be conceived as a computer, and that to be a computer is not a matter of fact or discovery, but a matter of perspective. And second, that representational content plays an essential role in the individuation of states and processes into computational types.

As a friend of brain and cognitive science, and a proponent of the computational approach, one of my objectives is to show that there is no conflict between the view

O. Shagrir (✉)
Departments of Philosophy and Cognitive Science,
The Hebrew University of Jerusalem,
Jerusalem 91905, Israel
e-mail: shagrir@cc.huji.ac.il

I advance here and the foundations of computational cognitive and brain science. If anything, it helps explain why the computational approach has been fruitful.

The paper is organized as follows. In Sect. 1 I review the notorious problem of physical computation, that is, the problem of distinguishing computing physical systems, such as desktops and brains, from physical systems, such as planetary systems, digestive systems, and washing machines, that do not compute. After enumerating the conditions for being a computer that have been adduced in the literature, I conclude that some of the distinguishing features have to do with the way we conceive the systems in question. In this sense, being a computer is, at least in part, a matter of perspective.

Why do we assume the computational outlook when we study the brain, but not when we study other systems? In seeking an answer to this question I examine (in Sect. 2) a study in computational neuroscience (Shadmehr & Wise, 2005). I observe that we apply computational theories when we want to explain how the brain performs a *semantic* task, i.e., a task specified in terms of representational content, and the (computational) explanation consists in postulating an information-processing *mechanism*. From this I conclude that we adopt the computational approach because we seek to explain how a semantic task can be carried out, and computational explanations are able to do this. But what is the source of this explanatory force? Why are computing mechanisms able to explain semantic tasks? I suggest (in Sect. 3) that the explanatory force of a computing mechanism derives from its correspondence to mathematical relations between the represented objects and states. In the last section (Sect. 4), I discuss several objections to my account, and argue that the individuation of a computing mechanism makes an essential reference to these mathematical relations, and, hence, to certain aspects of representational content.

1 The problem of physical computation: what does distinguish computers from other physical systems?

Computer models are often used to study, simulate, and predict the behavior of dynamical systems. In most cases, we do not view the modeled system, e.g., the solar system, as a computer. When studying the brain, however, our approach is different. In this case, in addition to using a computer model to simulate the system under investigation, i.e., the brain, we also take the modeled system itself to compute, viewing its dynamical processes as computing processes. Why is this so? What is it about the brain that makes us consider it, along with desktops and pocket calculators, a species of computer, when it doesn't even occur to us to accord that status to solar systems, stomachs and washing machines?

When we talk about computation in the context of mind and brain, we have to distinguish between two categories of computing. One is associated with certain everyday activities, e.g., multiplying 345 by 872. This sort of calculation is done "in-the-head," often with the aid of a pencil and paper. I call it *mental calculation* to signify that the computing agent is largely aware of the process and "controls" it. In characterizing mental calculation, we take the contrast class to be other mental processes, such as dreaming, mental imagery, deliberating, falling in love, and so forth, and demarcating calculation here does not raise any particular problem. The other category is that of *cerebral computation*. When we say that the brain is a computer, we are assuming that the processes underlying many mental phenomena—mental calculation, dreaming,

and so on—are themselves computations. In characterizing cerebral computation, the contrast class is different. Since we take brain processes to be electrical, chemical and biological, the question we should be asking is what makes them computational. What distinguishes these processes from all the other physical, chemical and biological processes, e.g., planetary movements, digestive processes and wash cycles, which are not conceived as computations? And, more generally, what is it that makes some *physical* processes, but not others, computations?

It should be emphasized that I do not seek a precise and definitive answer to the question of what it is that makes us deem a physical process computation. There are, undeniably, fuzzy cases such as look-up tables and infinite-time machines. The puzzle about physical computation does not arise merely because we do not have a precise definition of physical computation, but because we have a hard time coming up with a definition that distinguishes even the most obvious cases of computation from non-computation. There are conditions deemed—correctly, in my opinion—necessary for something to be considered a computer. But astonishingly, these conditions, individually and jointly, fail to differentiate even a single clear-cut case of a computer from a clear-cut case of a non-computer.

Two conditions are often invoked in characterizing computation. The first is that computation is a species of information-processing, or as the saying goes, "no computation without representation" (Fodor, 1981, p. 122; Pylyshyn, 1984, p. 62). What is meant by 'information' or 'representation' here? Our first reaction is that the notion of representation assumed by computation must be restricted. After all, every physical system can be interpreted as representing something. We can, for instance, take planetary systems, stomachs and washing machines to compute the solutions of the mathematical equations that describe their operations; that is, we can construe their states as representing certain numerical values. The question, then, is whether there is a kind of representation that unequivocally distinguishes computing systems from at least some non-computing systems.

One proposal is that suitable representations are those whose content is observer-independent. On this criterion, representations whose content is "derived" are excluded from the computational domain.[1] Our cognitive states are usually classified as representations of the former sort: whether I believe that Bush is a good president is said to be independent of what others think or what they take me to believe.[2] By contrast, the aforementioned construal of states of washing machines as representing numbers does not seem to satisfy the criterion, since this interpretation is observer-dependent: it is we who ascribe to them this representational force.

But this proposal draws the line in the wrong place. On the one hand, it is by no means implausible that planetary systems, stomachs and washing machines have observer-independent representational powers. Planetary systems might carry information about the Big Bang, stomachs about what one has eaten recently, and washing machines about what one has worn recently.[3] On the other hand, digital electronic systems, e.g., desktops, the paradigm cases of computing systems, operate on symbols whose content is, indisputably, observer-dependent. That the states of Deep Junior

[1] The distinction is suggested, e.g., by Dretske (1988), who uses the terms 'natural' and 'conventional,' and Searle (1992), who speaks of the 'intrinsic' and 'non-intrinsic.'

[2] But see Davidson (1990) and Dennett (1971) for a different view.

[3] See Dretske (1988, Chapter 3). Dretske further distinguishes information from (natural) representation, which might improve things somewhat, but cannot account for computing systems that operate on conventional signs.

represent possible states of chessboards is an interpretation we have ascribed to them; we could just as well have ascribed to them very different content.

Another proposal starts from the premise that a computation operates solely on a symbol system, i.e., a system of representations with combinatorial syntax and semantics.[4] Hence, representations that can differentiate computations from non-computations are those that constitute such systems. This criterion seems more plausible than the preceding one, since it appears applicable to all digital electronics, and, arguably, to mental representations as well. But it is far too strict. First, there are all sorts of *analog computers* that range over representations that are not symbolic. Second, the current trend in brain and cognitive science is to view the brain as a computer whose computational processes operate on representations that have no combinatorial structure. This approach is now widespread in the fields of connectionism, neural computation, and computational neuroscience.[5] Whether it is empirically adequate is not at issue here: we are *not* inquiring into whether the computational architecture of the brain is "classical" or "connectionist." We are, rather, inquiring into whether we can describe the brain as a computer even if it turns out that it does not operate on a combinatorial system of representations.

Thus the no-computation-without-representation condition does not advance our investigation at this point. Every system can be seen as representing something. Attempting to distinguish between different sorts of representations turns out to be irrelevant in the computational context. Some computations range over representations whose content is observer-independent, some over representations whose content is observer-dependent. Some computations range over representations whose structure is combinatorial, some over other sorts of representations.

An oft-invoked constraint is what Fodor (1980) calls the "formality condition": a process is computational only if it is formal. But what does formality entail? It has been associated with a number of features, particularly mechanicalness, abstractness, and algorithmicity. Mechanicalness is generally understood in one of two ways. The sense more common in the philosophy of science is that of a mechanism, namely, a causal process underlying a certain phenomenon or behavior: "Mechanisms are entities and activities organized such that they realize of regular changes from start or setup conditions to finish or termination conditions" (Craver, 2002, p. 68). Physical computation is surely mechanical in this sense: it is a causal process that involves changes in entities from initial conditions (inputs) to termination conditions (outputs). But so are the other physical processes, such as planetary movements, digestive processes and wash cycles, which are non-computing.

Another sense of mechanicalness is the logical sense. A process is mechanical in this sense if it is blind to the specific content of the symbols over which it operates. The way the rules of inference function in axiomatic systems is, perhaps, the paradigm case, "the outstanding feature of the rules of inference being that they are purely formal, i.e., refer only to the outward structure of the formulas, not to their meaning, so that they could be applied by someone who knew nothing about mathematics, or by

[4] See, e.g., Newell and Simon (1976), Chomsky (1980), and Fodor and Pylyshyn (1988).

[5] See, e.g., Rumelhart, McLelland, and PDP Research Group (1986), Smolensky (1988), Churchland and Sejnowski (1992), and Churchland and Grush (1999). A specific example is considered in the next section, a propos discussion of the work of Shadmehr and Wise.

a machine" (Gödel, 1933, p. 45).[6] Mechanicalness in this sense is often associated with mental calculation, and is indeed central to the definition of a formal system.[7] But in the context of physical computation, the condition, though satisfied, is not helpful: it does not distinguish computing from non-computing systems. Almost every physical process is mechanical in this sense. Planetary movements, digestive processes and wash cycles proceed regardless of the content one might ascribe to their intermediate states.

A second feature of formality is *abstractness*. This feature refers to a description of the system, that is formulated in terms of an abstract—i.e., mathematical, logical, or "syntactical"—language, and often called a "program."[8] It is often further required that the physical system implement this abstract description, which means, roughly, that its physical states and operations "mirror" the states and operations of the abstract description.[9] Now I agree that the computational description of a system is formulated in terms of some logical or mathematical language, and I have no quarrel with the fact that we can view the system as implementing this program. The problem is that this condition does not advance our goal of distinguishing computing from non-computing systems. After all, planetary movements, digestive processes, and wash cycles are also described by a set of mathematical equations, known as the laws of nature. And, much like digital computers, they can be seen as "implementing" these equations, in the sense that their physical states mirror or otherwise correspond to the states of the describing equations.

A third feature of formality is *algorithmicity*. This feature can be understood as amounting to no more than the combination of the features of mechanicalness and abstractness.[10] But it can also be understood in a stronger sense, as a structural constraint on computing. So understood, a physical computing system is formal in that it does not implement just any abstract structure, for instance, a set of differential equations, but rather, the implemented structure must be of a special sort: a Turing machine, a finite-state automaton, or perhaps some other kind of "digital" or "discrete" process.[11]

---

[6] This sense of being mechanical is stressed by Fodor: "Formal operations are the ones that are specified without reference to such semantic properties of representations as, for example, truth, reference and meaning" (1980, p. 309).

[7] Cf. Gödel: "Turing's work gives an analysis of the concept of 'mechanical procedure' (alias 'algorithm' or 'computation procedure' or 'finite combinatorial procedure'). This concept is shown to be equivalent with that of a 'Turing machine'. A formal system can simply be defined to be any mechanical procedure for producing formulas, called provable formulas" (in his Postscript to Gödel, 1934, *Collected Works*, vol. 1, pp. 369–370).

[8] The appeal to abstraction is hinted at in Gödel's reference to the "outward structure of the formulas." It is made more explicit by Fodor, who defines computation as "mappings from symbols under syntactic description to symbols under syntactic description" (1994, p. 8).

[9] The notion of implementation is controversial. Roughly, a physical system implements an abstract structure if its physical states "mirror" the operations of the abstract structure in the sense that there is some correspondence between the physical states and the abstract states. For example, if a physical state P corresponds to an abstract state A, and a physical state Q corresponds to an abstract state B, and if P always brings about Q, i.e., the conditional is counterfactual supportive, then A always brings about B. For refinements of this idea see Chalmers (1996) and Scheutz (2001).

[10] In my opinion, computational neuroscientists, e.g., Shadmehr and Wise (see next section), tend to take algorithmicity in this weaker sense.

[11] This feature of formality is stressed in Haugeland (1981). The claim that algorithms are captured by Turing machines is also made in Gödel's comment (note 7 above), and is generally referred to as the Church–Turing thesis. Note, however, that what I am calling "algorithmicity" is close to Gödel's notion

But the algorithmicity condition faces two major difficulties. For one thing, appearances to the contrary, it is satisfied by any physical system. At some level of description, everything is algorithmic: every physical process can be seen as a discrete state-transition process. This point is argued for compellingly by Putnam (1988) and Searle (1992), who further maintain that any physical system implements any "algorithm" whatsoever.[12] It may well be that Putnam and Searle rely on an excessively liberal notion of implementation.[13] But even if they do, their point is well taken. As I have suggested elsewhere (Shagrir, 2001), even under the stricter understandings of implementation, very simple physical devices simultaneously implement very different automata. This indicates, at the very least, that every physical system simultaneously implements several algorithms, even if not every algorithm.

A more serious difficulty is that the algorithmicity condition seems to be inadequate: it draws the line between computing and non-computing descriptions in the wrong place. As we noted, there are analog computers whose processes are considered to be non-algorithmic.[14] It is also possible to conceive of ideal physical systems that are in some sense "digital," yet compute non-recursive functions, i.e., functions that cannot be computed by means of an algorithm.[15] And thirdly, there is the important class of neural networks, artificial and biological, that can be viewed as computing, even though their dynamics are not "digital" in any obvious sense. These dynamics are described by "energy" equations, of the sort that describe many other dynamical systems, including spin glass systems whose particles align in the same direction.[16] Of course, we can count these neural computations as algorithmic, but then, the other such systems must also be deemed algorithmic.

We find ourselves in a conundrum: any physical system computes something, or even many things, and every physical process can be seen as computation. The familiar constraints fail to differentiate computing processes from non-computing processes. While computing is information-processing, and is mechanical and abstract, these features can just as well be said to characterize any physical process. And while some computing processes are algorithmic, in the structural sense outlined above, some computing processes are non-algorithmic.

What are we to make of this? Churchland, Koch, and Sejnowski (1990) state, correctly I think, that "whether something is a computer has an interest-relative

Footnote 11 continued
of a finite procedure. On the relationship between the notions of "finite procedure" and "mechanical procedure," see my "Gödel on Turing on Computability" (2006).

12 Putnam provides a proof for the claim that every physical system that satisfies certain minimal conditions implements every finite state automaton. Searle claims that every physical process can be seen as executing any computer program.

13 See Chalmers (1996) and Scheutz (2001).

14 These machines are algorithmic in the sense that their processes can be approximated by a Turing machine. The problem with this sense of algorithmicity is that it encompasses every physical system, including planetary systems, stomachs and washing machines. But see also the well-known exception presented in Pour-El and Richards (1981).

15 See Hogarth (1992, 1994). Shagrir and Pitowsky (2003) argue that while the postulated computing machine is digital, in that it consists of two communicating Turing machines, it is not "finite," since it can carry out infinitely many steps in a finite time span. Hence, it does not refute the Church–Turing thesis.

16 For an extensive discussion of this point, see my "Two Dogmas of Computationalism" (Shagrir, 1997). An artificial network whose dynamics are given by "energy equations" is described in Shagrir (1992). On network models of the brain, see the next section. For a detailed discussion of the relations between the theory of neural networks and statistical mechanics, see Amit (1989).

component, in the sense that it depends on whether someone has an interest in the device's abstract properties and in interpreting its states as representing states of something else" (p. 48). But I wouldn't go as far as Searle (1992) who argues that "computation is not discovered in the physics" (p. 225), that "syntax is not intrinsic to physics" (p. 208), and that "there is no way that computational cognitive science could ever be a natural science, because computation is not an intrinsic feature of the world. It is assigned relative to observers" (p. 212). True, we do not discover that the brain is a computer, but decide to so describe it.[17] But it does not follow, without additional assumptions, that there is nothing here to discover. First, the claim that computation is observer-relative does not imply that what the mind/brain represents is observer-relative. All we have said is that some computations, e.g., desktops, are defined over representations whose content is observer-relative—not that all are. Second, the claim that computing is observer-relative does not entail that the truth-value of the syntactic descriptions is observer-relative. Syntactic descriptions are true (or false) abstract descriptions of what the system does; in this sense, the physical system really implements these abstract structures. The "interest-relative component" is our decision to describe a system in these terms.

Searle's conclusion is, therefore, premature. That being a computer is a matter of perspective does not entail that computational cognitive science (neuroscience) has no empirical content. In particular, it is consistent with their *discovering* (a) the representational contents of the brain—which entities it represents, and (b) the operations that are performed over these representations. It might well be, therefore, that cognitive (and brain) science seeks to discover "the computational structure of the brain": the implemented abstract structure that is defined over the mental (or cerebral) representations.[18]

Searle arrives at his daring conclusion because he thinks that if being a computer is a matter of perspective, then the claim that the brain is a computer "does not get up to the level of falsehood. It does not have a clear sense" (p. 225). Indeed, we have described the brain as a computer without a clear sense of what we are talking about. But it does not follow that our talk is meaningless, just that we still have to unpack the notion of cerebral computation. Given that applying the computational outlook involves some interest-relative component, our task is to clarify what motivates us to apply the computational approach when studying brain and cognitive functions. We have to figure out why we apply it to some systems, e.g., the brain, and not others, e.g., planets or washing machines. The next step in our investigation is to consider this question in the context of a seminal study of computation.

2 Why to apply the computational approach: a case study

The Computational Neurobiology of Reaching and Pointing, by Shadmehr and Wise, offers a comprehensive treatment of the motor-control problem.[19] I will focus on part II (Chapters 9–14), where Shadmehr and Wise theorize about how the brain might

[17] See also Smith (Smith, 1996, 75 ff.).

[18] Whether computational cognitive science/neuroscience is a natural science in the "naturalistic" sense that computational types make no essential reference to "mental" or "semantic" items is discussed in length in Sect. 4.

[19] I learned about Shadmehr's work at the 2004 Washington University Computational Modeling and Explanation in Neuroscience workshop, both from Frances Egan, who discussed it from a

compute the vector difference between the location of the end-effector (i.e., the hand and objects it controls) and the target location. The discussion is divided into three parts: computing the end-effector location (Chapters 9–10), computing the target location (Chapter 11), and computing the difference vector between the end-effector and target locations (Chapters 12–14). I will not go into the empirical content of the theory in any detail; my aim is to see what we can learn about the philosophical issues at hand from this scientific study.

Like many scientists, Shadmehr and Wise do not provide a comprehensive account of what they mean by "computation." In a section of the introductory chapter entitled "Why a Computational Theory?," they rely mainly on Marr's framework:[20]

> In his often-quoted work on vision, David Marr described three levels of understanding CNS [central nervous system] functions: the level of a *computational theory*, which clarifies the problem to be solved as well as the constraints that physics imposes on the solution; the level of an *algorithm*, which describes a systematic procedure that solves the problem; and the level of *implementation*, which involves the physical realization of the algorithm by a neural network. A computational-level theory thus explains some of what a complex system does and how it *might* work. (pp. 3–4)

There are, however, striking differences between the approach advocated by Marr and the approach taken by Shadmehr and Wise, both with respect to the methodology of investigation and to the characterization of the three levels. Marr's methodology, famously, is top-down, moving from the computational level to the algorithmic, and ultimately, implementation. Computational theory "clarifies the problem" by identifying "the constraints that physics imposes on the solution", where "physics" denotes the physical environment. Shadmehr and Wise, on the other hand, rarely appeal to constraints from the physical environment.[21] In imposing constraints on the problem and its solution, they appeal, more often, to data from evolutionary biology, experimental psychology, robotics, and most of all, neuroscience.[22] Let me bring two examples to illustrate the role played by neurobiological constraints in computational theories.

Shadmehr and Wise's theory about the computation of the target location relies on the three-layer neural net model presented in Zipser and Andersen (1988). The inputs in the model are two sets of cells, one encoding information about eye position (orientation) relative to the head, another encoding the location of the stimulus (target) on the retina, i.e., its retinotopic location. The output is the location of the target in

Footnote 19 continued
philosophical viewpoint, and from Shadmehr's former student, Kurt Thoroughman, who discussed it from an empirical viewpoint.

[20] See Marr (1982), Chapter 1.

[21] It should be noted, however, that Marr's focus on the physical environment is not exclusive, and he also invokes neurobiological data as a constraint on solutions, and that Shadmehr and Wise do, to some extent, take the agent's environment into account. But there is a significant difference in the weight they give these factors.

[22] Evolutionary constraints are addressed in the first part of the book; see also the discussion of the evolutionary advantages of fixation-centered coordinates (p. 185). For behavioral data, see, e.g., Sect. 10.1.2 (pp. 160–162), where the results from behavioral experiments are taken to suggest that the CNS aligns visual and proprioceptive cues to produce an estimate of hand location. For cues from virtual robotics, e.g., in computing the forward kinematics problem, see Sect. 9.6 (pp. 148–151).

a head-centered set of coordinates.[23] The model rests on earlier electrophysiological results by Andersen, Essick, and Siegel (1985), who found three classes of cells in the PPC (in area 7a) of the monkey: (1) cells that respond to eye position only (15% of the sampled cells); (2) cells that are not sensitive to eye orientation (21%), but have an activity field in retinotopic coordinates; and (3) cells that combine information from retinotopic coordinates with information about eye orientation (57%).[24] Based on these constraints, Zipser and Andersen constructed the aforementioned three-layer model so as to have the two sets of inputs in the model correspond to the first two classes of PPC cells, and the hidden layer of the (trained) network correspond to the third class of PPC cells.

The second example pertains to the coordinate system relative to which the computation is carried out: it can be either body-centered, e.g., relative to the head or shoulder, or fixation-centered. These different systems become relevant with respect to the coordinate system in which the CNS represents the end-effector location.[25] The assumption underlying the Zipser and Andersen model is that the CNS uses a head-centered coordinate system. However, more recent data collected by Buneo, Jarvis, Batista, and Andersen (2002) suggests that in area 5d of the PPC, neuronal activity encodes target and hand locations in fixation-centered coordinates.[26] Based on this data, Shadmehr and Wise adjusted their computational theory, adopting visual coordinates, with the fovea as the point of origin.

It is also notable that Shadmehr and Wise differ from Marr in the way they characterize and distinguish between the three levels. Consider the distinction between the computational and algorithmic levels. For Marr, the computational theory "clarifies the problem to be solved," whereas the algorithmic level "describes a systematic procedure that solves the problem." For Shadmehr and Wise, the objective of the computational theory is to explain how the system *might* solve the problem, which, they say, amounts to ascertaining the "systematic procedure that solves the problem," namely, the "the process of computing" (p. 147). This is, of course, precisely what Marr sees as the objective of the algorithmic level. It would thus seem that Shadmehr and Wise do not recognize any significant difference between the levels. As far as they are concerned, to theorize at the computational level is to conjecture, on the basis of all available data, as to how the problem *might* be solved, whereas the 'algorithmic level' refers to way it *is* solved.[27]

Or consider the distinction between the algorithmic and implementation levels. Marr reserves the terms 'computation' and 'algorithm' for a cognitive, e.g., visual, system, and the term 'implementation' for their realization in the brain. Shadmehr and Wise draw no such distinction. On the one hand, it is clear that Shadmehr and Wise take the brain itself to be a computer. Indeed, the phrase "the CNS computes" occurs dozens, if not hundreds, of times throughout the book, for example, in the general thesis that "according to the model presented in this book, in order to control

[23] Shadmehr and Wise, pp. 193–197. Shadmehr and Wise diverge from Zipser and Andersen mainly in locating the outputs in a fixation-centered coordinate system, i.e., a visual frame with the fovea as its point of origin (see below).

[24] Shadmehr and Wise, pp. 188–192.

[25] See Shadmehr and Wise, pp. 209–212.

[26] See Shadmehr and Wise, pp. 212–216.

[27] Shadmehr and Wise thus conclude the section "Why a Computational Theory?" with the statement that the book "presents one plausible, if incomplete, framework for understanding reaching and pointing movements" (p. 4).

a reaching movement, the CNS computes the difference between the location of a target and the current location of the end effector" (p. 143). On the other hand, by "algorithm," Shadmehr and Wise do not refer to a sequential, digital, discrete, or controlled process, but to something altogether different. They almost always refer to a multi-directional, parallel, spontaneous, and often analog process, and always describe it by means of a neural network model.[28]

Keeping these differences in mind, let us focus on the objectives of computational theories. According to Shadmehr and Wise, the aim of a computational theory is to *explain* "some of what a complex system does and how it *might* work." This statement raises two questions: (a) What exactly is the explanandum of a computational theory: what are the computational problems that the brain solves? And (b) What exactly is a computational explanation: how does a computational theory explain how these problems are—or might be—solved? I address them in turn.

Two observations about the nature of computational problems come to mind immediately. First, the problem is formulated in terms of regularities: what is being computed is an input–output *function*. Thus in Chapter 9 Shadmehr and Wise consider "the problem of computing end-effector location from sensors that measure muscle lengths or joint angles, a computation called **forward kinematics**" (p. 143). In Chapter 11 they advance a theory, based on the Zipser–Andersen model, that seeks to explain how the CNS computes the target location from information about eye orientation and the target's retinotopic location. And in Chapter 12 they consider the problem of computing the vector difference from information about the locations of the end-effector and the target.

The second observation I want to make is that these regularities are specified in *semantic* terms. By this I mean more than that the inputs and outputs are representational states. I mean, in addition, that the function being computed is specified in terms of the content of these representations: *information* about joint angles, hand and target location, eye orientation and so forth. For example, information about the "end-effector location" is computed from information about "muscle lengths or joint angles": "the CNS computes the difference between the location of a target and the current location of the end effector" (p. 143). However, in calling these terms semantic I do not imply that they are intentional. In computational neuroscience, it is far from clear that the semantic concepts that *scientists* invoke, e.g., representation, information and encoding, refer to much more than stimuli–response causal relations, i.e., a way to interpret neural activity in cells as a response to certain environmental stimuli. The point is simply that regularities in the brain are formulated in these semantic terms.

I also do not suggest that the semantic terms are introduced once we apply the computational approach. To the contrary, assigning content and information to brain/mental states is often *prior* to taking the computational outlook. This is perhaps obvious in the case of computational models of "higher-level" cognitive phenomena. But it is also the case in computational neuroscience. To see this, let us take a closer look at the electrophysiological studies of Andersen et al. (1985). In these experiments

[28] Shadmehr and Wise also advocate a "non-classical," neural network approach in robotics. They write, e.g., that "rather than a symbolic computer program to align "proprioception" with "vision," the imaginary engineer might use one neural network based on feedforwarded connections to map proprioception to vision (forward kinematics) and another network to map vision to proprioception (inverse kinematics). Both of these networks would perform what is called a *function-approximation* computation" (p. 150).

Andersen and his colleagues record PPC neuronal activity from awake monkeys trained in various visuospatial tasks, e.g., fixating on a small point at different eye positions. They then label one group of neurons as "eye-position cells" interpreting their activity as "coding" a range of horizontal or vertical eye positions. They state, more generally, that the "brain receives visual information" and that "at least nine visual cortical fields ... contain orderly representations of the contralateral visual field" (p. 456). Undertaking their computational model, Zipser and Andersen (1988) do not introduce new semantic concepts but build on the interpretation and terminology of the prior electrophysiological studies.[29]

Taken together, we can say that the problems the brain computes are certain regularities or functions that are specified in terms of the representational content of the arguments (inputs) and values (outputs). I call these problems *semantic tasks*. The assertion that it is semantic tasks that are being computed is consistent with other computational studies of the brain.[30] It is also consistent with the computing technology tradition that calculators compute mathematical functions such as addition and multiplication, chess machines compute the next move on the chess-board, and robots compute motor-commands.[31] And it is consistent with much of the philosophical tradition.[32]

Let us now turn to the explanation itself. Perusal of Shadmehr and Wise suggests two features that are associated with computational explanations. First, explanation of a semantic task consists in revealing the (computing) process that mediates between the "initial conditions" (e.g., states that represent the joint angles) and the "termination conditions" (e.g., states that represent the hand location). In this sense, a computational explanation is a sort of *mechanistic explanation*: it explains how the computational problem is solved by revealing the mechanism, or at least a potential mechanism, that solves the problem. The second feature is that the computing process invoked by the explanation satisfies the conditions on computation mentioned in the first section: it is both information-processing and formal, whether the latter is taken to connote mechanicalness or abstractness.

The information-processing character of the process is evident throughout the book. On a single page, we find several statements to this effect: "the *process of computing* motor commands depends crucially on *information* provided by sensory systems"; "the coordinate frame used by the CNS for *representing* hand location reflects this dominance" ; "the idea that your CNS *encodes* hand location in a vision-based coordinates intuitive enough" (p. 147, emphasis added). The mechanical character of computing processes is also abundantly evident in the book, particularly in descriptions of computing processes in engineered robots. And the abstractness of computing

[29] I am grateful to an anonymous referee who encouraged me to add these comments.

[30] See, e.g., Marr (1982), and Lehky and Sejnowski (1988), whose computational theories seek to explain, e.g., how the brain extracts information about an object's shape from information about shading.

[31] E.g., Shadmehr and Wise's discussion of how to approach the problem of forward kinematics in robotics engineering (pp. 148–151): "the engineer's computer program can be said to *align* the mappings of gripper location in *visual and proprioceptive coordinates*" (pp. 148–149).

[32] E.g., Fodor (1994, Chapter 1) states that computational mechanisms are invoked to explain intentional regularities.

processes is manifest in the fact that mathematical language and neural net models are used to describe them.[33]

To gain insight into computational explanation, let us examine a real computational theory, the computational theory at the center of part II of Shadmehr and Wise, which seeks to explain how the brain computes the difference between the target and the current location of the hand. The first step is to divide the problem into sub-problems, each of which is also a *semantic task* that calls for explanation.[34] Shadmehr and Wise divide the problem into three parts: computing hand location from information about joint-angles; computing target location from information about eye-orientation and the retinotopic location of the target, and computing the difference vector from the locations of the hand and target.

The second step is to characterize the sub-tasks in abstract terms, that is, to describe input–output relations by means of, say, mathematical equations. I will focus on the computational problem of the second phase—computing target location from information about eye-orientation and the retinotopic location of the target. This problem is described by the vectorial transformation [R] + [xR] → [Cr], where [R] is a vector of the numerical activation values of cells that encode the stimulus (target) location in terms of a retinotopic coordinate system, [xR] stands for the values of electrical discharges that encode eye-location in head-centered coordinates, and [Cr] stands for those that encode the target location in head-centered coordinates.[35]

The third step is to clarify the relations between the abstract inputs, in our case, [R] and [xR], and the output [Cr]. To do this, Shadmehr and Wise appeal to the Zipser–Andersen model, which is a three-layer neural network trained to accomplish the task. The network consists of 64 input units that stand for the visual input [R] and another 32 input units that stand for the eye-position [xR]. Two output representations were used, both of which are meant to represent the stimulus location in head-centered position [Cr]. The number of units in the hidden layer is not specified. The discharge p_j of each cell j in the second and third layers, is a logistic function $1/(1+\mathrm{e}^{-\mathrm{net}})$.[36] After

[33] In the "Why a Computational Theory?" section, Shadmehr and Wise repeatedly associate computational theories with mathematical terms and entities, declaring, e.g., that numbers "enable computations and, therefore, computational theories" (p. 3).

[34] Obviously, the "steps" here serve to highlight the logical structure of the explanation; I am not claiming that the explanation actually proceeds by means of these "steps."

[35] Shadmehr and Wise, p. 194. Two points should be noted. First, this vectorial transformation fits the Zipser–Andersen model, and does not reflect Shadmehr and Wise's preference for fixation-centered coordinates. Second, [R] + [xR] signifies a vectorial combination that is, in fact, non-linear.

As to the computational problems tackled in the other phases, the forward kinematic problem is described in terms of a vector that represents the trigonometric relations between the joint angles and the end-effector location in a shoulder-based coordinate system; see pp. 151–157, and the supplementary information on Shadmehr's website: http://www.bme.jhu.edu/~reza/book/kinematics.pdf. The computational problem tackled in the last phase, computing the difference vector, is described by the equation $X_{\mathrm{dv}} = X_{\mathrm{t}} - X_{\mathrm{ee}}$, where X_{dv} is the difference vector (target location with respect to end-effector), X_{t} is the vector representing the target location in fixation-centered coordinates, and X_{ee} is the vector representing the end-effector location in fixation-centered coordinates.

[36] Net $= \Sigma w_{ij} \cdot p_i$, where p_i is the discharge of cell i, and w_{ij} is the weight from i to j. Since the network is feed-forward, for each cell, j, w_{ij} is defined only for (all) cells in the preceding layer. The initial weights are arbitrary: Zipser and Andersen train the net to find the exact mathematical relations between the input [R] + [xR] and the output [Cr] "by itself."

a training period, in which the weights are modified by means of back-propagation, the network exhibits the behavior [R] + [xR] → [Cr].[37]

The fourth and crucial step is to make use of the model to explain the semantic task. The Zipser–Andersen model is meant to explain two semantic features. It explains, first, the pattern of behavior in area 7a, i.e., how cells of the "third group" combine information from retinotopic coordinates with information about eye orientation. These cells are represented in the model by the units in the hidden layer. Analyzing the behavior of the hidden units, Zipser and Andersen found that these cells combine information about the inputs, much like the cells in area 7a. In particular, they behave in accordance with the equation $p_i = (k_i^T e + b_i) \exp(-(r - r_i)^T (r - r_i)/2\sigma^2)$, where e stands for eye-orientation with respect to two angular components, k_i and b_i represent the cell's gain and bias parameters, r_i is the center of the activity field in retinotopic coordinates, and σ describes the width of the Gaussian.[38] A second explanatory goal of the model is to show that there can be, at least in principle, cells coding eye-position-independent location of the stimulus. This goal is achieved by demonstrating that the output units extract from the information in the hidden units a coding of the target location in head-centered coordinates. The existence of such "output" cells in the brain, however, is debatable and has not been unequivocally demonstrated.

The last stage in the explanation is to combine the three "local" computational theories into an "integrated" computational theory that explains how the more general problem is solved. This is done via a neural net model that combines the three networks that describe how the sub-tasks are carried out.[39]

We are now in a better position to say why we apply the computational approach to some systems, e.g., brains, but not others, e.g., planetary systems, stomachs and washing machines. One reason is that computational theories seek to explain semantic tasks. We apply the computational approach when our goal is to explain a semantic pattern manifested by the system in question: to explain how the CNS produces certain motor commands, how the visual system extracts information about shape from information about shading, and how Deep Junior generates the command "move the queen to D-5." We do not view planetary systems, stomachs and washing machines as computers because the tasks they perform—moving in orbits, digesting and cleaning—are not defined in terms of representational content. Again, we could view them as computing semantic tasks, in which case we might have applied the computational approach. But we don't: we aim to explain the semantic tasks that desktops, robots and brains perform, but not the semantic tasks that planetary systems, stomachs and washing machines might have performed.[40]

[37] For Zipser and Andersen's presentation of the model, see Zipser and Andersen (1988), Fig. 4, p. 681. For a sketch of a model for forward kinematics, see see Shadmehr and Wise (2005) Fig. 9.3 on p. 149. For a sketch of a model for the third phase (computing the difference vector), see Fig. 12.5 on p. 212. For a detailed mathematical description, see Sect. 12.4 (p. 216 ff.) and Shadmehr's website: http://www.bme.jhu.edu/∼reza/book/recurrent_networks.pdf

[38] For further details, see Shadmehr and Wise, pp. 193–197.

[39] A sketch of the network is provided by Shadmehr and Wise in Fig. 12.11 on p. 223.

[40] There is the question of why we view the mind/brain as performing semantic tasks. Answering this question is beyond the analysis of computation: assigning semantic tasks is not part of the computational explanation, but is made prior to taking the computational approach (though the computational study might reveal that the semantic task being solved is not the one we initially attributed to the mind/brain).

A second reason is that computational theories do explain the pertinent semantic tasks that these systems perform. When we look at the Zipser–Andersen model, we *understand*, at least in principle, how the semantic task is carried out. We understand how the CNS might solve the semantic task of converting information about eye-orientation and the target's retinotopic location into information about the target's location in body-centered coordinates. This is not a trivial achievement. It is far from clear that other sorts of explanation, e.g., physical, chemical or biological, have this or any explanatory force with respect to semantic tasks.

So we have made some progress, but our analysis is not yet complete. To better understand why we apply the computational approach we should clarify what is the source of its explanatory force with respect to semantic tasks: we have to better understand the contribution of computational theories in explaining how semantic tasks are carried out.

3 On the explanatory force of computational theories

Where do we stand? We noted in the first section that to view something as a computer is to describe its processes as operating on representations, and as being formal, understood as both mechanistic and abstract. But since everything can be described this way, we concluded that computation is a matter of perspective, at least in part. We then asked why we adopt the computational attitude to some systems and not others. Why do we take brains and desktops, but not planetary systems, stomachs and washing machines, to be computers? Surveying the work of Shadmehr and Wise has provided at least a partial answer. The computational approach is an explanatory strategy that seeks to explain a system's execution of semantic tasks. Computational theories in neuroscience explain how the CNS accomplishes tasks such as solving the forward kinematics problem of extracting the end-effector location from information about muscle lengths and joint angles.

We now need to account for the fact that the computational approach gives us what we want, that is, they can explain how semantic tasks are carried out. Let me sharpen the point in need of clarification a bit more. Consider a semantic task of the type $F \rightarrow G$, e.g., converting information about eye-orientation and the retinotopic location of the target (F) into information about the target location in body-centered coordinates (G).

Let $[\mathrm{R}] + [\mathrm{xR}] \rightarrow [\mathrm{Cr}]$ signify the computing mechanism described in the Zipser–Andersen model, where $[\mathrm{R}] + [\mathrm{xR}]$ is the computational description of the state F, and [Cr] that of G. Let us also assume that $N_0 \rightarrow N_n$ is a "low-level" neural mechanism underlying the semantic task, where N_0 is, e.g., the electric discharge of certain cells in area 7a correlated with F, and N_n that which is correlated with G.

Now, when we look at the electrophysiological studies of Andersen et al. (1985) we see that they not only point to the pertinent neurological properties, N_i. They already formulate, at least partly, the semantic task $F \rightarrow G$ that is being performed: "many of the neurons can be largely described by the product of a gain factor that is a function of the eye position and the response profile of the visual receptive field. This operation produces an eye position-dependent tuning for locations in head-centered coordinate

space" (p. 456).[41] Nevertheless, Andersen et al. (1985) do not provide an explanation for how this semantic operation is produced. A possible explanation is provided only later on by Zipser and Andersen (1988), and, following them, by Shadmehr and Wise (2005), in terms of the computational model.

So what is it about computational models that makes them explanatory with respect to semantic tasks? And what is the explanatory force of the computational mechanism above and beyond its neurobiological counterpart? Before I present my own answers, let me address, very briefly, some answers that have been suggested by others.

One answer cites multiple realization. The claim here is that a semantic task can be implemented in different physical mechanisms, some of which are non-biological.[42] But this answer is not satisfying. First, it is far from obvious that with respect to multiple realization there is any difference between the computational and the neurological. On the one hand, it might well be that different species apply different computational mechanisms in performing a given semantic task. On the other, there is no evidence that the semantic tasks we seek to explain are multiply-realizable in creatures of the same species, e.g., humans.[43] Second, multiple realization does not seem to account for the explanatory force of computational models, e.g., the Zipser–Andersen model, in neuroscience. The explanatory force of the model does not stem from the possibility of being applied to other remote creatures. Its force is in explaining even particular cases of semantic regularities, e.g., how *my* CNS comes to represent the location of a target—that keyboard, say.

Fodor has argued that computational processes, which he views as "mappings from symbols under syntactic description to symbols under syntactic description" (1994, p. 8), are truth-preserving: "It is characteristic of mental processes they [psychological laws] govern that they tend to preserve semantic properties like truth. Roughly, if you start out with a true thought, and you proceed to do some thinking, it is very often the case that the thoughts that the thinking leads you to will also be true" (1994, p. 9). I agree with Fodor that computational processes "preserve semantic properties," but I do not think that he correctly accounts for the explanatory import of this feature. First, if the syntactic processes are truth-preserving, their neural implementations must be truth-preserving too. But this leaves open the question of why these processes, under neural descriptions, do not suffice for explanatory purposes, even though they *are* truth-preserving. Second, Fodor's account is confined to "classical" processes, whereas we are interested in the explanatory power of computational models in general. For the question we seek to answer is *not* that of the empirical adequacy of the classical model of mind.[44] Rather, we are trying to understand why computational models, classical or non-classical, have explanatory power with respect to semantic tasks.

[41] I note again that the task described by Andersen et al. (1985) parallels the transformation between the input and hidden units in Zipser and Andersen (1988).

[42] See, e.g., Fodor (1974) and Putnam (1975). For a version of this argument that highlights complexity considerations see Putnam (1973); in particular, the peg-hole example. For a response, see Sober (1999).

[43] For a detailed discussion of these points with reference to Zipser and Andersen (1988), see Shagrir (1998).

[44] This is, however, the question Fodor addresses. He thus writes: "This emphasis upon the syntactical character of thought suggests a view of cognitive processes in general—including, for example, perception, memory and learning—as occurring in a languagelike medium, a sort of 'language of thought' " (p. 9).

According to Sejnowski, Koch, and Churchland, "mechanical and causal explanations of chemical and electrical signals in the brain are different from computational explanations. The chief difference is that a computational explanation refers to the information content of the physical signals and how they are used to accomplish a task" (1988, p. 1300). Indeed, Zipser and Andersen explain the semantic task $F \rightarrow G$ by referring to F, that is, they explain how cells in state N_n encode information, G, about the target location in head-centered coordinates, by referring to the information content, F, of cells in state N_0 in area 7a, i.e., that the electrical signals of some of these cells encode information about stimulus retinotopic-location and that the electrical signals of other cells encode information about eye-orientation. Thus referring to the information content of electrical signals is a central ingredient of computational explanations. The question, still, is why invoke, *in addition*, the computing mechanism to explain the semantic transformation. Why describing the mediating mechanism between N_0 and N_n in terms of the computational model, and not in terms of chemical and electrical signals in the brain?

Shadmehr and Wise suggest that a computational theory is an incomplete framework for how the brain "*might* solve these and other problems". A computational theory simply "helps to understand–in some detail—at least one way that a system *could* solve the same problem" (p. 4). In doing this, they serve to bridge between neuroscientists, who study brain function, and engineers, who design and construct mechanical devices,[45] and they promote further research, e.g., additional electrophysiological experiments, by pointing to a set of alternative hypotheses as to how the brain might accomplish the task.[46] On this picture, the explanatory advantage of computational theories is similar to that of mathematical models in other sciences: being more abstract, they underscore regularities that neurological, not to mention molecular, descriptions obscure.[47] But this does not mean that computational descriptions have extra explanatory qualities with respect to semantic tasks that strict neurobiological descriptions lack.

I do not underestimate the role of mathematical models in science in general, and in neuroscience in particular. But I want to insist that computational models have additional and unique explanatory role with respect to semantic tasks. Let me sketch what I take this explanatory role to be. A computational theory in neuroscience aims to explain how the brain extracts information content G from another, F. We *know* from electrophysiological experiments that G is encoded in the electrical signals of a neural state N_n, and F in those of N_0. Let us also assume that we can track chemical and electrical processes that mediate between N_0 and N_n. What we still want to know is *why* the information content of the electrical signals of N_n is G, e.g., target location in head-centered coordinates, and not, say, the target's color. The question arises since the encoded information G are not "directly" related to the target, but is mediated through a representational state that encodes F. The question, in other words, is why a chemical and electrical process that starts in a brain state N_0 that encodes F, and terminates in another brain state, N_n, yields the information content G. After all, this causal process $N_0 \rightarrow N_n$ is not sensitive to what is being represented, but to chemical and electrical properties in the brain.

45 Shadmehr and Wise, pp. 2–3.

46 This was suggested by a referee of this article.

47 A more sophisticated version of this picture, in terms of levels of organization in the nervous system, is presented in Churchland and Sejnowski (1992).

A computational theory explains why it is G that is extracted from F by pointing to correspondence relations between mathematical or formal properties of the representing states—what we call computational structure—and mathematical or formal properties of the represented states. The idea is that by showing that N_0 encodes F, and that the mathematical relations between N_0 and N_n correspond to those of the "worldly" represented states that are encoded as F and G, we have shown that N_n encodes G.

Let me explicate this point by means of three examples. The first is fictional. The brown–cow cell transforms information about brown things and about cow things into information about brown–cow things. The cell receives electrical inputs from two channels: it receives 50–100 mV from the "brown-channel" just in case there is a brown thing in the visual field and 0–50 mV otherwise, and it receives 50–100 mV from the "cow-channel" just in case there is a cow thing out there and 0–50 mV otherwise (I assume that the information is about the same object). The cell emits 50–100 mV whenever there is a brown cow in the visual field, and 0–50 mV otherwise. How does this cell encode information about brown cows: how does it extract information about brown cows from information about brown things and information about cow things?

The computational explanation is obvious enough. First, we describe the cell as an AND-gate that receives and emits 1's and 0's; these abstracts from emission/reception of 50–100 mV and 0–50 mV. This is the computational structure of the cell. Second, we refer to the information content of the inputs, which are brown things and cow things. And, third, we note that the AND-gate corresponds to a mathematical relation between the represented objects, that is, it corresponds to the *intersection* of the set of brown things and the set of cow things. Taken together, we understand why the information content of emission 50–100 mV is brown-cows: emitting 50–100 mV is an AND result of receiving 50–100 mV, and the AND-operation corresponds to the intersection of the sets of represented objects. Thus given that receiving 50–100 mV from each output channel represents brown things and cow things, emitting 50–100 mV must represent brown–cow things.

Marr's computational theory of early visual processes explains how cells extract information about object-boundaries from retinal photoreceptors that measure light intensities.[48] Marr and Hildreth suggest that the activity of the former cells, known as edge-detectors, can be described in terms of zero crossings in the mathematical formula $(\nabla^2 G) * I(x, y)$, where $I(x, y)$ is the array of light intensities (retinal image), $*$ is a convolution operator, G is a Gaussian that blurs the image, and ∇^2 the Laplacian operator $(\partial^2/\partial x^2 + \partial^2/\partial y^2)$ that is sensitive to sudden intensity changes in the image. Thus at one level, this formula describes a certain mathematical relation between the electrical signals of the edge-detectors and those of the photoreceptors that constitute the retinal image, $I(x, y)$. But this alone does not explain why the activity of the edge-detectors corresponds to object boundaries and other discontinuities in surface properties.[49] The explanation is completed when we note that these mathematical properties correspond to certain mathematical properties of the represented states, for example, of sudden changes in light reflection along boundaries of objects. Had the reflection laws been very different, the mathematical formula would still have

[48] Marr (1982, Chapter 2).

[49] Thus Marr (1982, p. 68) states that, at this point, it is better not to use the word 'edge', since it "has a partly physical meaning—it makes us think of a real physical boundary, for example—and all we have discussed so far are the zero values of a set of roughly band-pass second-derivative filters" (p. 68).

described the same cellular activity, but the cells would no longer carry information about object boundaries.[50]

Let us return to the Zipser–Andersen model. Analyzing the activity of the units in the intermediate layer, Zipser and Andersen found that these units behave like cells in area 7a that combine information about the target's retinotopic location with information about eye-orientation. It was then found that the mathematical description of the activity is $(k_i^T e + b_i) \exp(-(r - r_i)^T (r - r_i)/2\sigma^2)$, which might explain how these cells encode the combined information. The explanation refers to the content of the first two groups of cells in area 7a, and, in particular, that the parameter e stands for eye-orientation with respect to two angular components, and r_i is the center of the activity field in retinotopic coordinates. But it also establishes the findings of the earlier electrophysiological studies: that the described mathematical relations between the electrical signals correspond to certain mathematical relations between the represented states: for example, that the described mathematical relation $(r - r_i)$ of cellular activity (in the second group) corresponds (roughly) to the distance ("in numbers") of the retinotopic location of the stimulus from the receptive field.[51] Without this and the other correspondence relations between mathematical properties, we could have not explained why the cellular activity described by the mathematical formula conveys combined information about eye-orientation and the retinotopic location of the target.

The gist of my account, then, is as follows. When we describe something as a computer we apply a certain explanatory strategy. We apply this strategy when we seek to explain how a complex system performs a semantic task. The explanatory force of this strategy arises from identifying correspondence relation between the computational structure of the system and certain mathematical properties of the states and objects that are being represented.

4 Discussion

The following discussion, in the form of objections and replies, will further elaborate on my position, and situate it in the context of the philosophical discussion about computation and content.

Objection #1: It seems that you uphold some version of a "semantic" view of computation, but it is not clear what it amounts to.

Reply: Piccinini (2004) has recently characterized the semantic view of computation in terms of the no computation without representation condition. My view is certainly "semantic" in this sense, and as I said, this condition has been widely adopted.[52] But I think that there is another, more significant distinction to be made.

[50] As Marr (p. 68) puts it, if we want to use the word 'edges', we have to say why we have the right to do so. The computational theory justifies this use since it underscores the relations between the structure of the image and the structure of the real world outside.

[51] See also Grush (2001, pp. 160–162) who highlights, in his analysis of Zipser and Andersen (1988) the relations between what he calls the algorithm-semantic interpretation (which I take to be the mathematical properties of the cells) and the environmental-semantic interpretation (which I take to be the objects/states/properties that are being represented).

[52] Thus Piccinini rightly says that "the received view" is that there is no computation without representation. A notable exception is Stich (1983). I discuss Piccinini's objection to the no-computation-without representation condition in objection 6.

Because some—many, in fact—of those who accept that there is no computation without representation also hold computation to be "non-semantic" in the sense that the specification of states and processes as computational types makes no essential reference to semantic properties, including representational content. Indeed, this is, I would say, a more accurate statement of the received view about computation. That is, on the received view, computation is formal, in virtue of which it is—notwithstanding the no computation without representation constraint—non-semantic.[53] There are, however, those who maintain that computation is "semantic" in the sense that computational individuation does make essential reference to representational content. This view has been advanced by Burge (1986) and others with respect to computational theories in cognitive science.[54] I belong to this latter "semantic" camp, though my view is distinctive in crucial respects (see below).

Objection #2: You seem to be contradicting yourself. On the one hand, you say that computation is "semantic" in the stronger, individuative, sense. On the other, you uphold the view that computational structure is formal. You thus have to clarify how computational individuation that makes essential reference to content can be compatible with the formality condition.

Reply: I accept the idea that computation is formal, in the senses of mechanicalness and abstractness. A computational description is formulated in abstract terms, and what it describes is a causal process that is insensitive to representational content. However, it does not follow from these features that computation is non-semantic. It is consistent with mechanicalness and abstractness that semantic elements play an essential role in the individuation of the system's processes and states into computational types. They play an essential role in picking out "the" computational structure from the set of the abstract structures that a mechanism implements.

To better understand how content constrains computational identity, consider again the brown–cow cell. Assume that receiving/emitting 0–50 mV can be further analyzed: the cell emits 50–100 mV when it receives over 50 mV from each input channel, but it turns out that it emits 0–25 mV when it receives under 25 mV from each input channel, and 25–50 mV otherwise. Now assign "1" to receiving/emitting 25–100 mV and "0" to receiving/emitting 0–25 mV. Under this assignment the cell is an OR-gate. This means that the brown–cow cell simultaneously implements, at the very same time, and by means of the very same electrical activity, two different formal structures. One structure is given by the AND-gate and another by the OR-gate.[55]

Now, each of these abstract structures is, potentially, computational. But only the AND-gate is the computational structure of the system with respect to its being a brown-cow cell, namely, performing the semantic task of converting information about brown things and cow things into information about brown cow things. What determines this, that is, picks out this AND-structure as the system's computational structure, given that OR-structure abstract from the very same discharge? I have

[53] Fodor is the most explicit advocate of this view: "I take it that computational processes are both *symbolic* and *formal*. They are symbolic because they are defined over representations, and they are formal because they apply to representations, in virtue of (roughly) the *syntax* of the representations.... Formal operations are the ones that are specified without reference to such semantic properties of representations as, for example, truth, reference and meaning" (1980, p. 64).

[54] See also Kitcher (1988), Segal (1989, 1991), Davies (1991), Morton (1993), Peacocke (1994, 1999), Shagrir (2001). I have also argued for this "semantic" view in the context of computer science (Shagrir, 1999).

[55] A more complex example is presented in detail in Shagrir (2001).

suggested that it is content that makes the difference: discharges that are greater than 50mV correspond to certain types of content (of cows, browns, and brown cows) and the discharges that are less than 50 mV corresponds to another (their absence). Thus the identity conditions of the process, *when conceived as computational*, are determined, at least partly, by the content of the states over which it is defined.

Objection #3: This semantic account of computation cannot be correct. We know that we can assign many different interpretations to the same computational (abstract) structure, e.g., a computer program. This demonstrates that differences in content do *not* determine computational identity, for the computational identity can be the same even where there is a difference in content.

Reply: I am not saying that computational taxonomy takes every aspect of content into account; in fact, I do not think it takes specific content into account at all. Rather, it takes into account only mathematical properties of the represented objects; these features of content that have been called 'mathematical content' by Egan (1995) and 'formal content' by Sher (1996).[56] Take the brown–cow cell whose computational structure is given by the AND-gate. It is apparent that the computational identity of the cell would have been the same had the '1' and '0' has been given a different interpretation, e.g., that it is a black-dog cell. Still, these interpretations do have something in common. Their mathematical content is the same: the AND-gate corresponds to the same set-theoretic property in the visual field, i.e., that of the intersection of two sets.

Objection #4: But why should we care about how the computational structure is picked out? What matters is that we can *identify* this formal structure without appealing to semantic properties at all, i.e., identify it as an abstraction from the physical properties of the system.

Reply: I agree that the computational structure of a physical system is an abstraction from its physical or neurological properties, and as such, is "non-semantic." It is evident that the AND-operation describing the behavior of the brown–cow cell is an abstraction from the cell's electrical activity, viz., that its discharge is 50–100 mV if the stimulus in each input channel exceeds 50 mV, and 0–50 mV otherwise. I also agree that we can arrive at this very same formal description by abstracting from physical properties. I have emphasized, moreover, that Shadmehr and Wise often arrive at computational structure via constraints from the neurobiological level. My point about computation being semantic, therefore, is *not* methodological or epistemic; it is not about the way we arrive at the computational structure, about top-down versus bottom-up.

I am claiming, rather, that (a) we take an abstract (formal) structure to be computational when it plays a role in explaining how a system carries out a pertinent semantic task, and that (b) the explanatory role of this abstract structure consists in its being identified by reference to certain semantic features, i.e., mathematical content. We may find out, one way or another, that a certain mathematical description of a cell's behavior is an AND-operation, which is an abstraction from the cell's electrical activity. However, this mathematical description, on its own, is not computational. Physicists often describe systems in mathematical terms, but no one takes

[56] Sher presents the notion of formal content in her analysis of logical consequence. She argues that it is the formal, e.g., set-theoretic, features of the objects they denote, that make certain relations "logical" ; for a full account, see Sher (1991). Egan introduces the notion of mathematical content in the context of computational theories of vision. I follow her in this regard, but emphasize, like Sher, that formal properties are higher-order mathematical structures.

such descriptions to be computational. We take the description to be computational only in the context of explaining a semantic task, in this case, explaining the behavior of the cell as a brown–cow cell. The explanatory power of the AND-operation is due to its satisfying certain semantic constraints, viz., that it correlates with the intersection of two sets. And having satisfied these constraints is essential for its being identified as the computational structure of this cell. The OR-operation, which is simultaneously implemented by the cell, is not identified as the computational structure of the brown–cow cell; for, unlike the AND-operation, the OR-operation does not satisfy the relevant semantic constraints, and thus cannot serve to explain the behavior of the cell as a brown–cows detector.

Objection #5: Your semantic account cannot be correct simply because there can be computation without representation. In his "Computation without Representation" (forthcoming), Piccinini argues that "although for practical purposes the internal states of computers are usually ascribed content by an external semantics, this need not be the case and is unnecessary to individuate their computational states and explain their behavior."

Reply: In this paper I have focused on the computational approach in neuroscience and cognitive science. In these disciplines, even a glimpse at ongoing research suffices to make apparent the intimate connection between computation and representation. In other contexts, the connection may be less obvious, but I believe that careful scrutiny of any process described as computation will reveal the centrality of representations. I agree with Piccinini's assertion that the "mathematical theory of computation can be formulated without assigning any interpretation to the strings of symbols being computed." But while this means that the mathematical properties of computations can be studied without reference to semantic values, it does not entail that the processes themselves, qua computations, are not identified by their semantic values.

Objection #6: A semantic view of computation undermines the whole objective of computational theories of cognition, which is to explain content in non-semantic terms. As Piccinini puts it, an important motivation for many supporters of the computational theory of mind [CTM] is that "it offers (a step towards) a naturalistic explanation of mental content" (2004: 376). This objective is incompatible with a semantic view of computation, for "one problem with naturalistic theories of content that appeal to computational properties of mechanisms is that, when conjoined with the semantic view of computational individuation, they become circular" (Piccinini, forthcoming). By adopting the semantic view of computation, in other words, "we are back to explaining contents, which is what we were hoping to explain in the first place" (2004, p. 377).

Reply: I agree that CTM is entrenched in our philosophical landscape: many philosophers maintain that computational theories explain in non-semantic terms phenomena that are formulated in semantic terms. But in my opinion this philosophical outlook is flawed. Computational explanations are "semantic" in two ways: they refer to the specific content of the initial states, and they invoke a computing mechanism (structure) which is individuated by reference to (mathematical) content. I thus also agree that "naturalistic theories of content that appeal to computational properties of mechanisms . . . become circular." But I also note that a semantic view of computation is consistent with a naturalistic approach to content. A theory of content that does not appeal to computation, namely, a causal or teleological theory, might well

"naturalize" computation-in-the-brain, e.g., by accounting for the pertinent contents in non-semantic terms.[57]

Objection #7: Your semantic notion is only one way of accounting for computation. But there could be others that do not appeal to semantic features. You considered only non-semantic accounts, according to which computations are abstractions from internal physical properties. But there could be other, broader accounts, which take into consideration not only internal physical properties, but distal stimuli and responses as well.[58]

Reply: I have provided an analysis of the concept of computation as it is applied to physical systems, and as it functions in neuroscience. The analysis, if correct, reveals that when we apply the computational strategy, we individuate states and processes by taking certain aspects of content into account. This insight helps to explain why we apply the computational strategy to brains but not to washing machines, and why we apply the computational strategy, rather than other strategies, in studying how the brain functions. Other accounts of computation did not suggest themselves in the course of the analysis, certainly not broader accounts: Shadmehr and Wise construe computations as processes that take place within the brain, and explicate the relations between the brain and the external environment via the notion of representation. As I just mentioned, I do not rule out the possibility that the pertinent semantic relations can be reduced, e.g., to certain causal relations involving distal stimuli and responses. And, admittedly, I cannot rule out the possibility that this reduction might give rise to a broader, non-semantic, account of computation. I doubt such an account will be forthcoming, but my reasons for this pessimism will have to wait for another occasion.

Acknowledgements Thanks to Arnon Levy, Gualtiero Piccinini, Haim Sompolinsky, Jonathan Yaari and two anonymous referees for many insightful comments. This research was supported by The Israel Science Foundation (Grant No. 857/03).

References

Amit, D. J. (1989). *Modelling brain function*. Cambridge: Cambridge University Press.

Andersen, R. A., Essick, G. K., & Siegel, R. M. (1985). Encoding of spatial location by posterior parietal neurons. *Science, 230*(4724), 456–458.

Buneo, C. A., Jarvis, M. R., Batista, A. P., & Andersen, R. A. (2002). Direct visuomotor transformations for reaching. *Nature, 416*, 632–636.

Burge, T. (1986). Individualism and psychology. *Philosophical Review 95*, 3–45.

Chalmers, D. J. (1996). Does a rock implement every finite-state automaton? *Synthese, 108*, 309–333.

Chomsky, N. (1980). *Rules and representations*. New York: Columbia University Press.

Churchland, P. S., & Grush, R. (1999). Computation and the brain. In R. A. Wilson, & F. C. Keil (Eds.), *The MIT Encyclopedia of the cognitive sciences* (pp. 155–158). Cambridge, MA: MIT Press.

Churchland, P. S., Koch, C., & Sejnowski T. J. (1990). What is computational neuroscience? In E. L. Schwartz (Ed.), *Computational neuroscience* (pp. 46–55.) Cambridge, MA: MIT Press.

Churchland, P. S., & Sejnowski, T. J. (1992). *The computational brain*. Cambridge, MA: MIT Press.

Craver, C. F. (2002). Structures of scientific theories. In P. Machamer, & M. Silberstein (Eds.), *The Blackwell Guide to the Philosophy of Science*. (pp. 55–79.) Oxford: Blackwell.

[57] See Piccinini (2004). This would not mean, however, that this alleged naturalistic account can be applied to all computers, for there are other computers that operate over "non-mental" representations.

[58] See, e.g., Wilson (1994) and Piccinini (forthcoming).

Davidson, D. (1990). Turing's Test. In K. A. Mohyeldin Said, W. H. Newton-Smith, R. Viale, & K. V. Wilkes (Eds.), *Modeling the mind* (pp. 1–11.) Oxford: Oxford University Press.
Davies, M. (1991). Individualism and perceptual content. *Mind, 100*, 461–484.
Dennett, D. C. (1971). Intentional systems. *Journal of Philosophy, 68*, 87–106.
Dretske, F. (1988). *Explaining behavior*. Cambridge, MA: MIT Press.
Egan, F. (1995). Computation and content. *Philosophical Review, 104*, 181–203.
Fodor, J. A. (1974). Special sciences, or the disunity of science as a working hypothesis. *Synthese, 28*, 97–115.
Fodor, J. A. (1980). Methodological solipsism considered as a research strategy in cognitive psychology. *Behavioral and Brain Sciences, 3*, 63–73.
Fodor, J. A. (1981). The mind-body problem. *Scientific American, 244*, 114–123.
Fodor, J. A. (1994). *The elm and the expert*. Cambridge, MA: MIT Press.
Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition, 28*, 3–71.
Gödel, K. (1933). The present situation in the foundations of mathematics. In S. Feferman, J. W. Dawson, W. Goldfarb, C. Parsons, & R. M. Solovay (Eds.), *Kurt Gödel collected works*, Vol. III (pp. 45–53). New York: Oxford University Press (1995).
Gödel, K. (1934). On undecidable propositions of formal mathematical systems. In S. Feferman, J. W. Dawson, S. C. Kleene, G. H. Moore, R. M. Solovay, & J. van Heijenoort (Eds.), *Kurt Gödel collected works*, Vol. I (pp. 346–371). New York: Oxford University Press (1986).
Grush, R. (2001). The semantic challenge to computational neuroscience. In P. Machamer, R. Grush, & P. McLaughlin (Eds.), *Theory and method in the neurosciences* (pp. 155–172). Pittsburgh, PA: University of Pittsburgh Press.
Haugeland, J. (1981). Semantic engines. In J. Haugeland (Ed.), *Mind design* (pp. 1–34). Cambridge, MA: MIT Press.
Hogarth, M. L. (1992). Does General Relativity allow an observer to view an eternity in a finite time? *Foundations of Physics Letters, 5*, 173–181.
Hogarth, M. (1994). Non-Turing computers and non-Turing computability. *Proceedings of the Philosophy of Science Association, 1*, 126–138.
Kitcher, P. (1988). Marr's computational theory of vision. *Philosophy of Science, 55*, 1–24.
Lehky, S. R., & Sejnowski, T. J. (1988). Network model of shape-from-shading: Neural function arises from both receptive and projective fields. *Nature, 333*, 452–454.
Marr, D. (1982). *Vision*. San Francisco: W. H. Freeman.
Morton, P. (1993). Supervenience and computational explanation in vision theory. *Philosophy of Science, 60*, 86–99.
Newell, A., & Simon, H. (1976). Computer science as empirical inquiry: Symbols and search. *Communications of the Association for Computing Machinery, 19*, 113–126.
Peacocke, C. (1994). Content, computation, and externalism. *Mind and Language, 9*, 303–335.
Peacocke, C. (1999). Computation as involving content: A response to Egan. *Mind and Language, 14*, 195–202.
Piccinini, G. (2004). Functionalism, computationalism, and mental contents. *Canadian Journal of Philosophy, 34*, 375–410.
Piccinini, G. (forthcoming). Computation without representation. *Philosophical Studies*.
Pour-El, M. B., & Richards, I. (1981). The wave equation with computable initial data such that its unique solution is not computable. *Advances in Mathematics, 39*, 215–239.
Putnam, H. (1973). Reductionism and the nature of psychology. *Cognition, 2*, 131–146.
Putnam, H. (1975). Philosophy and our mental life. In H. Putnam (Ed.), *Mind, language and reality*, philosophical papers, volume 2 (pp. 291–303). Cambridge: Cambridge University Press.
Putnam, H. (1988). *Representations and reality*. Cambridge, MA: MIT Press.
Pylyshyn, Z. W. (1984). *Computation and cognition.* Cambridge, MA: MIT Press.
Rumelhart, D. E., McLelland, J. L., & the PDP Research Group (1986). *Parallel distributed processing*, Vol. 1–2. Cambridge, MA: MIT Press.
Scheutz, M. (2001). Computational versus causal complexity. *Minds and Machines, 11*, 543–566.
Searle, J. R. (1992). *The rediscovery of the mind*. Cambridge, MA: MIT Press.
Segal, G. (1989). Seeing what is not there. *Philosophical Review, 98*, 189–214.
Segal, G. (1991). Defense of a reasonable individualism. *Mind, 100*, 485–494.
Sejnowski T. J., Koch, C., & Churchland, P. S. (1988). Computational neuroscience. *Science, 241*(4871), 1299–1306.
Shadmehr, R., & Wise, S. P. (2005). *The computational neurobiology of reaching and pointing: A foundation for motor learning*. Cambridge, MA: MIT Press.

Shagrir, O. (1992). A neural net with self-inhibiting units for the n-queens problem. *International Journal of Neural Systems, 3*, 249–252.
Shagrir, O. (1997). Two dogmas of computationalism. *Minds and Machines, 7*, 321–344.
Shagrir, O. (1998). Multiple realization, computation and the taxonomy of psychological states. *Synthese, 114*, 445–461.
Shagrir, O. (1999). What is computer science about? *Monist, 82*, 131–149.
Shagrir, O. (2001). Content, computation and externalism. *Mind, 110*, 369–400.
Shagrir, O. (2006). Gödel on turing on computability. In A. Olszewski, J. Wolenski, & R. Janusz (Eds.), *Church's thesis after 70 years* (pp. 393–419). Frankfurt: Ontos Verlag.
Shagrir, O., and Pitowsky, I. (2003). Physical hypercomputation and the Church–Turing thesis. *Minds and Machines, 13*, 87–101.
Sher, G. Y. (1991). *The bounds of logic: A generalized viewpoint*. Cambridge, MA: MIT Press.
Sher, G. Y. (1996). Did Tarski commit "Tarski's Fallacy"? *Journal of Symbolic Logic, 61*, 653–686.
Smith, B. C. (1996). *On the origin of objects*. Cambridge, MA: MIT Press.
Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioral and Brain Sciences, 11*, 1–23.
Sober, E. (1999). The multiple realizability argument against reductionism. *Philosophy of Science, 66*, 542–564.
Stich, S. P. (1983). *From folk psychology to cognitive science*. Cambridge, MA: MIT Press.
Wilson, R. A. (1994). Wide computationalism. *Mind, 103*, 351–372.
Zipser, D., & Andersen, R. A. (1988). A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature, 331*, 679–684.