Friday Aug 27 2010 09:36 PM/PHOS10651/2010/77/5
/sji/lfw/ /ms editing started/use-graphics/narrow/default/

Computation, San Diego Style

Oron Shagrir†‡

What does it mean to say that a physical system computes or, specifically, to say that the nervous system computes? One answer, endorsed here, is that computing is a sort of modeling. I trace this line of answer in the conceptual and philosophical work conducted over the last 3 decades by researchers associated with the University of California, San Diego (UCSD). The linkage between their work and the modeling notion is no coincidence: the modeling notion aims to account for the computational approach in neuroscience, and UCSD has been home to central studies in neurophilosophy, connectionism, and computational neuroscience.

1. Introduction. The view that the brain computes is a working hypothesis in cognitive and brain sciences.[1] But what exactly is meant by the statement that the brain computes? What distinguishes computing systems, such as brains, from noncomputing systems, such as stomachs and tornadoes? Elsewhere (Shagrir 2010), I offer a tentative account according to which computing is a sort of modeling. The gist of the account is that a system computes just in case its input-output mapping relations preserve (in a sense to be specified) mathematical relations in the represented domain. My aim here is to describe how this modeling notion evolved, at least partly, from projects conducted at the University of California, San Diego (UCSD). The first phase was a critique, during the 1980s, of the "structural notion" on which the distinction between computing and noncomputing has to do, at least in part, with the structure of the mechanism (sec. 2). It is shown that this notion is inadequate in the context of neuroscience (sec. 3). The second phase was an attempt to advance an alternative account according to which computing has nothing to do with structure but only with information processing (sec. 4). The third phase was the

†To contact the author, please write to: Department of Philosophy, Hebrew University of Jerusalem, Jerusalem 91905, Israel; e-mail: shagrir@cc.huji.ac.il.

‡Thanks to Yonatan Loewenstein, Gualtiero Piccinini, and a journal referee. This research was supported by the Israel Science Foundation grant 725/08.

1. See, e.g., Marr (1982, 5), Koch (1999, 1), and Shadmehr and Wise (2005, 143).

Philosophy of Science, 77 (December 2010) pp. 000–000. 0031-8248/2010/7705-0004$10.00
Copyright 2010 by the Philosophy of Science Association. All rights reserved.

recognition that computing is not just a mapping from representation to representation but a special sort of mapping. It is special in that it preserves, through its input-output mapping relations, mathematical relations in the represented domain (sec. 5).

2. Computing as Structure. The accepted wisdom is that the distinction between computing and noncomputing has to do, at least in part, with the structure of the mechanism. Obviously, every mechanism has some structure. But the claim is that a mechanism is a computing one only if it has unique structural properties. By structural properties is meant certain architectural properties; in effect, those holding this view associate computing mechanisms with digital architectures.

For example, Fodor characterizes computing processes as those that "apply to representations in virtue of (roughly) the *syntax* of representations" (1980, 309), which for him are processes that can be characterized as "occurring in a languagelike medium" (1994, 9), namely, in classical architectures (Fodor and Pylyshyn 1988). An example of noncomputing processes are those that occur in most connectionist machines. Newell and Simon famously associate a computer with a "physical symbol system," which has "a specific architectural" structure (1976, 42). Newell specifies this structure in terms of five subsystems: memory, operators, control, input, and output (1980, 142; see also Harnish 2002, 396–97).

Cummins characterizes computing processes in terms of program execution, which, for him, means that they have the unique structure of *step* satisfaction: "To compute a function g is to execute a program that gives o as its output i just in case $g(i) = o$. Computing reduces to program execution. . . . Program execution reduces to step-satisfaction" (1989, 91–92). Noncomputing processes also satisfy functions, but in contrast to the computing ones, they are not doing so in a stepwise manner. Piccinini characterizes computation as "the generation of output strings of digits from input strings of digits in accordance with a general rule that depends on the properties of the strings and (possibly) on the internal state of the system" (2008b, 34; see also Piccinini 2007, 521). Computing processes thus manipulate inputs in accordance with a rule defined over the input types and their place within strings, whereas the noncomputing mechanisms are often described as nondiscrete processes governed by dynamical equations.

I use the phrase "structural notion" to refer to all those accounts that associate computing with unique architecture. These accounts can differ in detail and origin. Some accounts are anchored in logical proof-theoretic systems (Fodor 1980; see also Haugeland 1981a), others in computability theory (Cummins 1989; Searle 1992), automata theory (Putnam 1988; Chalmers 1996), artificial intelligence (Newell and Simon 1976), or mechanistic explanations (Piccinini 2007; see also Edelman 2008). Nevertheless, q1 they all reduce computing to some type of a digital architecture; they all view the digital electronic devices as paradigm examples of computing systems.

3. Computational Neuroscience: The Case against Structure. The structural notion has its virtues but also one vice: it is at odds with much of the work in connectionist cognitive science and, even more so, in computational neuroscience. This gap is highlighted, with respect to connectionist computation, by Rumelhart, McClelland, and the PDP Research Group (1986, 121–27; the group was centered at UCSD) and then, also with respect to computational neuroscience, by some philosophers, for example, Paul Churchland, who argues that "it has become increasingly clear that the brain is organized along computational lines radically different from those employed in conventional digital computers" (1989, 156).[2]

A concrete example can clarify the conceptual gap between computation and structure, at least in the context of computational neuroscience. It is known that the brain moves the eyes with quick saccadic movements and that, between saccades, it keeps the eyes still (Leigh and Zee 2006). What is not obvious is that the brain holds the eyes still because it stores a memory of eye positions; when the memory is damaged, there is a constant drift of the eyes to a null point. The brain's memory of horizontal eye position appears to be implemented by a persistent neural activity, which is spread over several areas in the brain stem and the cerebellum. Various experimental studies conducted on humans, monkeys, cats, and goldfish show that this system converts transient eye-velocity-encoding inputs into persistent eye-position-encoding outputs. It was concluded that the system locates the new eye position by computing integration on the pulse saccadic inputs with respect to time, hence, the name oculomotor integrator (Robinson 1989). The motor neurons, reading out the memory, keep the eyes still in the new location.

Using the language of state space analysis, the oculomotor integrator moves from one state, S_i, which represents one eye position, E_i, to another state, S_j, which represents another eye position, E_j, by performing integration on eye-velocity-encoding inputs *I* (fig. 1). How does this system compute integration in arriving at the new position? The experimental findings show that when the eyes are still, the pattern of neural activity in the memory is constant in time and that, for every eye position, the pattern of activity is different and persistent. This persistence is thus

2. See also Churchland and Sejnowski, who emphasize that "identifying computers with serial digital computers is neither justified nor edifying" (1992, 61–62).

Friday Aug 27 2010 09:36 PM/PHOS10651/2010/77/5
/sji/lfw/ /ms editing started/use-graphics/narrow/default/

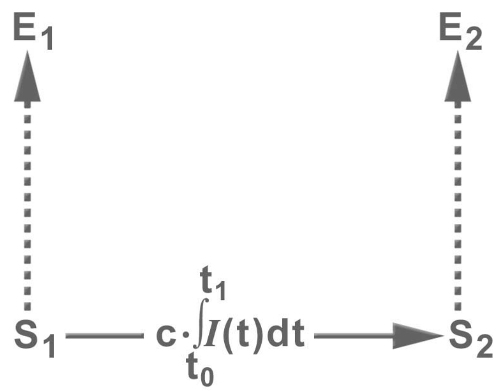

Figure 1. Dynamics of the oculomotor memory. The system consists of states S_i, each representing (*dashed arrow*) a different eye position E_i, and moves from one state, say S_1, to another state, say S_2, by integrating on pulse saccadic inputs, *I*, which encode eye velocity.

explained in terms of a multistable recurrent network with a *continuous*
q2 *line attractor* dynamics (Cannon, Robinson, and Shamma 1983; Seung
1996, 1998); the continuity is required for consistency with the analog,
graded encoding of the eye position in the neural activity. Each point
along the line encodes a different eye position. A new stimulus destabilizes
the network, which gradually relaxes on a new point along the attractor
q3 line; this point encodes the current eye position (fig. 2).

This example highlights the difficulties of applying the structural notion of computing in neuroscience. First, the integrating mechanism is at odds with the idea that computing has to do with a digital architecture. It is true that the network consists of separable cells, but the activation of cells is described in terms of a continuous function governed by differential equations. Moreover, the space of states is not discrete, in the sense that it consists of separable attractors; the space, rather, is dense, in that it is a continuous line of attractors, whereas each point along the line is, ideally, a different stable state. In addition, the dynamics of the network is not, in any obvious sense, a rule-governed or step-satisfaction process. The dynamics is described as minimization of the energy landscape. In short, the dynamics is no more digital than the dynamics of stomachs, tornadoes, or any other noncomputing system.

Second, the integrating mechanism is at odds with the more general claim, namely, that the distinction between computing and noncomputing has to do with structure. The oculomotor integrator has no obvious struc-

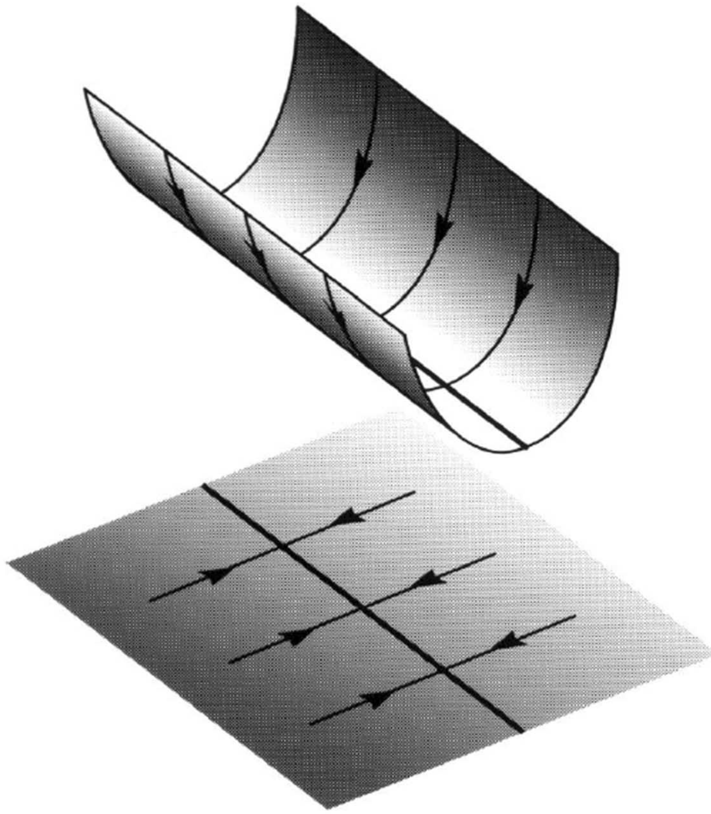

Figure 2. State space portrait of the memory network. All trajectories flow toward a line attractor. Each point along the line is a different attractor state S_i, and each state S_i represents a different eye position E_i. A new eye-velocity-encoding stimulus, *I*, forces the network to move to a new attractor state (representing the current eye position), by integrating over the stimulus. From Seung 1996, 13340; © 1996 by National Academy of Sciences, U.S.A.

tural properties that distinguish it from all the noncomputing dynamics. Even the language used to describe the dynamics is not the language of automata or computability theory. The integrating process is described in the language of control theory and dynamical theory, which is the same language used to describe many noncomputing dynamical systems. So, all in all, it is very difficult to see what could be a relevant structural difference between this computing system and what we count as noncomputing. Structure of mechanism, it seems, contributes nothing to this distinction.

4. Computing as Information Processing: The CKS Version. In a series of papers, Patricia Churchland, Christoff Koch, and Terrence Sejnowski (henceforth, CKS) advance an alternative conception of computing, with the aim of making sense of the computational view in neuroscience. The gist of their account is that computing has to do with information or representation. In their words, "A physical system is considered a computer when its states can be taken as representing states of some other system" (Churchland, Koch, and Sejnowski 1990, 48). They further argue that this semantic feature differentiates between computational explanations and other causal explanations: "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" (Sejnowski, Koch, and Churchland 1988, 1300).

Characterizing computation as information processing is not unusual. In fact, most structural accounts assume that computing is a species of information processing. They assume that yet another constraint on computing, apart from digital architecture, is that there is "no computation without representation" (Fodor 1981, 122; Pylyshyn 1984, 62; notable exceptions are Stich [1983] and Piccinini [2008a]). The unusual feature in CKS's characterization is that it does not include, in addition to the semantic constraint, a structural constraint. Computing mechanisms, according to CKS, do not have unique architectural features that differentiate them from other causal and mechanical processes.

Giving up a structural constraint raises the concern that the characterization is too liberal. One worry is that being a computer is not just a matter of fact; it is, at least in part, a matter of perspective. This is because, strictly speaking, we can interpret the states of every system as representing something else. Interestingly, CKS themselves concede just that: "Whether something is a computer has an interest-relative 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" (Churchland et al. 1990, 48). "Delimiting the class of computers is not a sheerly empirical matter, and hence that 'computer' is not a natural kind"; whether something is a computer has to do with "the subjective and practical matter of whether we care what the function is" (Churchland and Sejnowski 1992, 65).

I agree with CKS that, in some trivial sense, everything can be seen as computing. I argue elsewhere (Shagrir 2010), however, that this "universality feature" has no negative implications for the status of the computational approach in neuroscience. Here, I will make just three brief comments. First, we might view something as a computing system, although actually we do not. CKS tell us what it is that we do when we actually view something as a computer. So their account, if right, is valuable in that it supports the distinction between systems that we actually view as computing and those that we do not. Second, it is doubtful that adding a structural constraint makes the universality feature go away; in some trivial sense, everything can be seen as implementing an automaton (Putnam 1988) or executing a program (Searle 1992). Third, the universality feature does not entail that "computation is not discovered in the physics" (Searle 1992, 225). We could simply distinguish between systems, arguably desktops, whose representational powers are "derivative" or "observer dependent" and systems, arguably brains, whose representational powers are "natural" or "observer independent" (see, e.g., Dretske 1988). If brains belong to the latter sort, there is no reason to think that their representational, hence computational, powers are not "intrinsic to physics." Quite the contrary: scientists describe brains as computing because they think that brains have natural representational (hence, computational) powers and that discovering and referring to these powers is essential in explaining behavior.

The real concern with CKS's account lies elsewhere. It is that there are many processes that we view as mappings from representation to representation, yet we do not view them as computing; examples are old-fashioned slide projectors and mental processes such as dreaming and imagining (viewed at a phenomenological level). Unlike these processes, computational processes are, in a sense to be specified, *formal* processes. CKS, in effect, repeatedly emphasize that when we view something as a computer we also refer to the "device's abstract properties." But then the question is, "Which abstract properties?" It does not seem the case that merely referring to the device's abstract properties (or describing it in abstract terms) will make the slide projector compute. The "formality condition," it seems, states more than that.

5. Computing as Modeling. The modeling notion characterizes computing as a special sort of information processing. Computing is information processing in the sense that it maps representation to representation. But computing is more than that. In addition to representing entities in some domain, the system's input-output functional relations preserve mathematical relations between the entities that the inputs and outputs represent.

The characterization of the modeling notion originates, at least partly, in several philosophical works conducted at UCSD. One source is the notion of computing advanced by CKS that puts semantics rather than structure in the forefront. Another is the pioneering work of Gila Sher (1991, 1996) on the foundations of logic. Sher advances a semantic, model-theoretic account of logical consequence. In this account, the formality constraint on logical consequence is not identified with the recursive-syntactic structure of the formulas or the inference rules. A consequence is formal (hence, logical) due to certain mathematical properties and relations denoted by the sentences/formulas. Thus, the consequence "everything is round and blue; therefore everything is blue" is logical due to certain set-theoretic relations, that is, inclusion, between the set of round and blue things and the set of blue things.[3]

Another important source is Paul Churchland's work in neurophilosophy. Looking at neural network models, Churchland (see his collected papers: 2007) noticed that there is similarity or isomorphism between high-dimensional relations (e.g., geometrical congruence) in the state space of the representing network and high-dimensional relations (e.g., geometrical congruence) in the represented domain "in the world." Thus, when exemplifying his point with road maps, Churchland writes that "it is these interpoint distance relations that are collectively isomorphic to a family of real world similarity relations among a set of features within the domain being represented" (2007, 107). Churchland, however, does not associate this isomorphism directly with computing; his main objective is to show how these similarities underlie central notions in epistemology and in semantics.

An early characterization of computing in terms of modeling is presented by Rick Grush (2001).[4] Analyzing the Zipser-Andersen model (1988), Grush states that "we can suppose that the function computed by an idealized posterior parietal neuron is something like $f = (\mathrm{e} - \mathrm{e_P})\,\sigma(\mathrm{r} - \mathrm{r_i})$" (2001, 161). According to Grush, however, the term 'compute' does not just mean that the cells "are acting in accord with" function (159), namely, that there is some mechanism that supports this input-out
q4 behavior. When we say that the PPC cells "really are computing a function" (159), we mean that in addition to their behaving in accordance with *f*, we take *f* to be the "interpretation of stimulus distance from preferred direction relative to the head" (161).

Grush's point can be explicated more easily with the example of the oculomotor integrator. In this context, we can see that the term $\int I(\mathrm{t})\,\mathrm{d}(\mathrm{t})$ is a description of two different (mathematical) relations (fig.

3. The relevance of Sher's account to the characterization of computing is pointed out in my dissertation (Shagrir 1994; see also Shagrir 2001).

4. Grush discusses at length other aspects of cognitive modeling in his dissertation (1995).

Friday Aug 27 2010 09:36 PM/PHOS10651/2010/77/5
/sji/lfw/ /ms editing started/use-graphics/narrow/default/

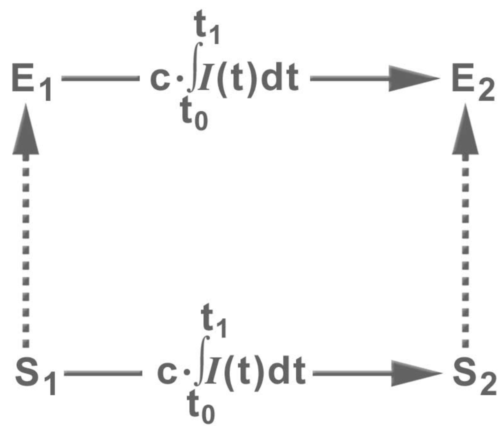

Figure 3. Modeling by the oculomotor integrator: $\int I(t)\,d(t)$ describes both the relations between two states of the neural memory, S_1 and S_2, and the relations between the two eye positions, E_1 and E_2, that S_1 and S_2 represent.

3). One is the mapping relation between the stable states of the neural network (oculomotor integrator), S_i and S_j. This relation is just the integration over the transient saccadic electrical inputs to the network, *I*, with respect to time.[5] Under this description ("a-interpretation" in Grush's vocabulary), the system is acting in accordance with integration. But we can note that $\int I(t)\,d(t)$ also describes the relation between the eye positions E_i and E_j being represented by these states. This relation is just the integration of the eye velocity, *I*, with respect to time. Under this description ("e-interpretation"), the system is really computing integration.

More generally, we would say that a system is acting in accordance with a mathematical function *f*, when *f* is a description of the system's functional (input-output) relations. We would say that a system computes a mathematical function *f* when (and only when), in addition, it satisfies the following two conditions: (*a*) the states of the system are (or can be interpreted as) representing states in another domain, and (*b*) the functional relations between the states being represented (by the inputs and outputs) are also *f*.[6]

What is the scope of the modeling notion of computing? Does it account

5. The relation is mathematical in the sense that it relates real-number values abstracted from the electric discharges of the cells.

6. The notion of modeling has affinities with the notion of S-representation (Cummins 1989; Swoyer 1991). The modeling notion takes S-representation to be a (in fact *the*) defining feature of computing.

Friday Aug 27 2010 09:36 PM/PHOS10651/2010/77/5
/sji/lfw/ /ms editing started/use-graphics/narrow/default/

for all the cases in which we describe something as computing? Does it cover, at the very least, the cases where we describe the nervous system as a computing system? I do not argue that the modeling notion underlies all the cases where the system is described as computing. Nor do I argue that it covers all the cases where the described system is a cognitive, or perhaps even a nervous, system. It might well be that structural accounts can be applied to cases to which the modeling account does not apply. What I do insist on, rather, is that the modeling notion can account for many cases, in neuroscience and elsewhere, some of which cannot be addressed by the structural notion.

First, a computing system does not have to represent only actual physical states; it can also represent potential, counterfactual, and imaginary states. It is hypothesized, for example, that our integrator is also used to compute outputs that encode desired eye position from inputs that encode the velocity of the head movement (Robinson 1989; Goldman et al. 2002). A system can also represent, as I shall indicate in a moment, abstract entities such as numbers and automata.

Second, the modeling notion covers many instances (although perhaps not all) of digital computing systems. There are models in classical cognitive science that assume that our cognitive (representational) system q5 preserves high-dimensional relations (for discussion, see Ramsey 2007, 77–92). The modeling notion also fits with machines that realize abstract automata. We can think of the "realization relation" in terms of representation, where the mathematical relations between the representing states correspond to the relations between the states of the automaton.[7] Thus, the modeling notion encompasses McCulloch and Pitts's (1943) neural network, whose behavior can be seen as "realizing" logical expressions.

Third, the modeling notion is related to analog computers; this indicates that the modeling notion is not invoked just in neuroscience and has respectable historical roots. The structural notion cannot tell why we view these systems as computing, as their structure is typical of the noncomputing dynamics. But the modeling notion can; in fact, the aim of the analog machines was to simulate mathematical relations of other physical systems by implementing these very relations in the computing system itself. A paradigm case of such a computing system is an analyzer designed to solve differential equations by using wheel-and-disc mechanisms that perform integration. Another example is the tide-predicting machine that determines the height of the tides by the integration of 10 principal toothed

7. See Dresner (forthcoming) for the claim that the realization relation is essentially a sort of representation relation.

Friday Aug 27 2010 09:36 PM/PHOS10651/2010/77/5
/sji/lfw/ /ms editing started/use-graphics/narrow/default/

wheels that simulated the motion of the sun, moon, earth, and other factors that influence tides.

Finally, there are works in computational neuroscience that are a better fit with the modeling notion. One is the oculomotor integrator; others are sensorimotor systems such as the ones described by the Zipser-Andersen model (1988) and by Shadmehr and Wise (2005; see discussion in Shagrir 2006). Computational theories of perception also lean toward the modeling notion. Consider, for example, the theory of edge detection (Marr and Hildreth 1980; Marr 1982, chap. 2; see discussion in Shagrir, forthcoming). Marr describes the pertinent early visual processes as computing the zero crossings of the formula $\nabla^2 G * I(x, y)$; I refers to the array of photoreceptors (retinal image), "$*$" is a convolution operator, and $\nabla^2 G$ is a filtering operator; G is a Gaussian that blurs the image, and ∇^2 is the Laplacian operator ($\partial^2/\partial x^2 + \partial^2/\partial y^2$), which is sensitive to sudden intensity changes in the image. The structural notion does not fit easily with this analog-to-digital dynamics. But the modeling notion does. The formula describes mathematical relations at two different levels. It describes the "inner" relations, between the activity of the retinal photoreceptors (and retinal ganglion cells) and the activity of cells in the primary visual cortex (V1). But it also describes the "outer" relations between what is being represented by these cells. The formula states that the relations between light-intensity values, signified by I, and the light reflectance along physical edges are those of derivation (i.e., that sudden changes in light intensities occur along physical edges such as object boundaries).

6. Conclusion. This article has described how the modeling notion of computing evolved from various research projects conducted at UCSD. It was first noticed, during the 1980s, that the structural notion of computing is at odds with much of the work in connectionist cognitive science and computational neuroscience. This observation led Churchland, Koch, and Sejnowski to advance an alternative account that puts the emphasis on the idea that computing is information processing. Their account was gradually refined, however, in a way that respects the idea that computing mechanisms are formal processes. The outcome of this refinement is the modeling notion according to which a (computing) system preserves, through its input-output functional relations, mathematical relations in the represented domain.

REFERENCES

Cannon, Stephen C., David A. Robinson, and Shihab A. Shamma. 1983. "A Proposed Neural Network for the Integrator of the Oculomotor System." *Biological Cybernetics* 49:127–36.

Chalmers, David J. 1996. “Does a Rock Implement Every Finite-State Automaton?” *Synthese* 108:309–33.

Churchland, Patricia S., Christof Koch, and Terrence J. Sejnowski. 1990. “What Is Computational Neuroscience?” In *Computational Neuroscience*, ed. Eric L. Schwartz, 46–55. Cambridge, MA: MIT Press.

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

Churchland, Paul M. 1989. *A Neurocomputational Perspective: The Nature of Mind and the Structure of Science*. Cambridge, MA: MIT Press.

——— 2007. *Neurophilosophy at Work*. New York: Cambridge University Press.

Cummins, Robert. 1989. *Meaning and Mental Representation*. Cambridge, MA: MIT Press.

Dresner, Eli. Forthcoming. “Measurement-Theoretic Representation and Computation-Theoretic Realization.” *Journal of Philosophy*. q6

Dretske, Fred. 1988. *Explaining Behavior: Reasons in a World of Causes*. Cambridge, MA: MIT Press.

Edelman, Shimon. 2008. *Computing the Mind: How the Mind Really Works*. New York: Oxford University Press.

Fodor, Jerry A. 1980. “Methodological Solipsism Considered as a Research Strategy in Cognitive Psychology.” *Behavioral and Brain Sciences* 3:63–73. Repr. Haugeland 1981a, 307–38.

——— 1981. “The Mind-Body Problem.” *Scientific American* 244:114–23.

——— 1994. *The Elm and the Expert*. Cambridge, MA: MIT Press.

Fodor, Jerry A., and Zeno W. Pylyshyn. 1988. “Connectionism and Cognitive Architecture: A Critical Analysis.” *Cognition* 28:3–71.

Goldman, Mark S., Chris R. S. Kaneko, Guy Major, Emre Aksay, David W. Tank, and Sebastian H. Seung. 2002. “Linear Regression of Eye Velocity on Eye Position and Head Velocity Suggests a Common Oculomotor Neural Integrator.” *Journal of Neurophysiology* 88:659–65.

Grush, Rick. 1995. “Emulation and Cognition.” PhD diss., University of California, San Diego.

——— 2001. “The Semantic Challenge to Computational Neuroscience.” In *Theory and Method in the Neurosciences*, ed. Peter Machamer, Rick Grush, and Peter McLaughlin, 155–72. Pittsburgh: University of Pittsburgh Press.

Harnish, Robert M. 2002. *Minds, Brains, Computers: An Historical Introduction to the Foundations of Cognitive Science*. Malden, MA: Wiley-Blackwell.

Haugeland, John, ed. 1981a. *Mind Design: Philosophy, Psychology, Artificial Intelligence*. Cambridge, MA: MIT Press.

——— 1981b. “Semantic Engines: An Introduction to Mind Design.” In Haugeland 1981a, 1–34. q7

Koch, Christof. 1999. *The Biophysics of Computation: Information Processing in Single Neurons*. New York: Oxford University Press.

Leigh, John R., and David S. Zee. 2006. *The Neurology of Eye Movements*. 4th ed. New York: Oxford University Press.

Marr, David C. 1982. *Vision*. San Francisco: W. H. Freeman.

Marr, David C., and Ellen Hildreth. 1980: “Theory of Edge Detection.” *Proceedings of the Royal Society of London* B 207:187–217.

McCulloch, Warren S., and Walter Pitts. 1943. “A Logical Calculus of Ideas Immanent in Nervous Activity.” *Bulletin of Mathematical Biophysics* 5:115–33.

Newell, Allen. 1980. “Physical Symbol Systems.” *Cognitive Science* 4:135–83.

Newell, Allen, and Herbert Simon. 1976. “Computer Science as Empirical Inquiry: Symbols and Search.” *Communications of the Association for Computing Machinery* 19:113–26. Repr. Haugeland, 1981a, 35–66.

Piccinini, Gualtiero. 2007. “Computing Mechanisms.” *Philosophy of Science* 74:501–26.

——— 2008a. “Computation without Representation.” *Philosophical Studies* 137:205–41.

——— 2008b. “Computers.” *Pacific Philosophical Quarterly* 89:32–73.

Putnam, Hilary. 1988. *Representations and Reality*. Cambridge, MA: MIT Press.

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

Friday Aug 27 2010 09:36 PM/PHOS10651/2010/77/5
/sji/lfw/ /ms editing started/use-graphics/narrow/default/

Robinson, David A. 1989. “Integrating with Neurons.” *Annual Reviews of Neuroscience* 12: 33–45.

Rumelhart, David E., James L. McClelland, and the PDP Research Group. 1986. *Parallel Distributed Processing*. Vol. 1. Cambridge, MA: MIT Press.

Searle, John R. 1992. *The Rediscovery of the Mind*. Cambridge, MA: MIT Press.

Sejnowski, Terrence J., Christof Koch, and Patricia S. Churchland. 1988. “Computational Neuroscience.” *Science* 241:1299–1306.

Seung, Sebastian H. 1996. “How the Brain Keeps the Eyes Still.” *Proceedings of the National Academy of Sciences of the USA* 93:13339–44.

——— 1998. “Continuous Attractors and Oculomotor Control.” *Neural Networks* 11:1253–58.

Shadmehr, Reza, and Steven P. Wise. 2005. *The Computational Neurobiology of Reaching and Pointing: A Foundation for Motor Learning*. Cambridge, MA: MIT Press.

Shagrir, Oron. 1994. “Computation and Its Relevance to Cognition.” PhD diss., University of California, San Diego.

——— 2001. “Content, Computation and Externalism.” *Mind* 110:369–400.

——— 2006. “Why We View the Brain as a Computer.” *Synthese* 153:393–416.

——— 2010. “Brains as Analog-Model Computers.” *Studies in the History and Philosophy of Science*, forthcoming.

——— Forthcoming. “Marr on Computational-Level Theories.” *Philosophy of Science*.

Sher, Gila Y. 1991. *The Bounds of Logic: A Generalized Viewpoint*. Cambridge, MA: MIT Press.

——— 1996. “Did Tarski Commit ‘Tarski’s Fallacy’?” *Journal of Symbolic Logic* 61:653–86.

Stich, Stephen P. 1983. *From Folk Psychology to Cognitive Science: The Case against Belief*. Cambridge, MA: MIT Press.

Swoyer, Chris. 1991. “Structural Representation and Surrogative Reasoning.” *Synthese* 87: 449–508.

Von Neumann, John. 1958. *The Computer and the Brain*. New Haven, CT: Yale University
q8 Press.

Zipser, David, and Richard A. Andersen. 1988. “A Back-Propagation Programmed Network That Simulates Response Properties of a Subset of Posterior Parietal Neurons.” *Nature* 331:679–84.

QUERIES TO THE AUTHOR

q1. Au: Italics omitted here and throughout when used for emphasis yet meaning seems clear without them or when not at first use of key terms.

q2. Au: Replaced Cannon and Robinson 1985 with Cannon, Robinson, and Shamma 1983 here, OK? If not, please (1) provide full reference details for Cannon and Robinson 1985 (or eliminate its callout here) and (2) provide at least one gcallout for Cannon, Robinson, and Shamma 1983 somewhere in run of text (or eliminate it from References).

q3. Au: Figure 2: Have copyright permissions been obtained for this figure? I assume they're needed, given how you've cited it here (with the copyright). Also, is the copyright text prescribed by the academy as is, or should it be, e.g., National Academy of Sciences of the USA vs. National Academy of Sciences, U.S.A?

q4. Au: (1) Please define PPC here. Also, (2) added "their" ("their behaving in accordance"), assuming the PPC cells are "behaving," OK? If not, please revise to clarify what is "behaving."

q5. Au: Ramsey 2007 not in reference list; please add, or omit citation here.

q6. Au: Do you have updated publication information for this forthcoming article?

q7. Au: Haugeland 1981b not cited in text; please cite, or omit from reference list.

q8. Au: Von Neumann 1958 not cited in text; please cite, or omit from reference list.

Friday Aug 27 2010 09:36 PM/PHOS10651/2010/77/5
/sji/lfw/ /ms editing started/use-graphics/narrow/default/