

Marcin Miłkowski, *Explaining the Computational Mind*, MIT Press, 2013, 248 pp., \$35.00, ISBN: 9780262018869

It is often said that at least some explanations in cognitive science are computational; they assume that cognition is some sort of computation. In *Explaining the Computational Mind*, Miłkowski aims to provide a philosophical analysis of computational explanation in cognitive science.

As Miłkowski immediately notes, a major obstacle in providing such an analysis is that it is far from clear what is meant by computation in the current context of cognitive science. Most interestingly, there is a gulf between the notion of computation we find in philosophy of mind and the notion of computation we find in the current theoretical practices of cognitive and brain science (and it is important to note that cognitive science today is closely linked to, or perhaps even embedded in, cognitive, computational and system neuroscience). In philosophy of mind, computation has often been identified with formal proofs, Turing machines and digital computation. As it turns out, however, these notions are hardly relevant to many of the varieties of computation found in current computational approaches in cognitive science. What is needed, then, is a more general notion of a physical computing system.

In characterizing physical computation, philosophers and theoreticians turn in two (not mutually exclusive) directions. One is the semantic view, which associates computation with

one or another semantic feature; Miłkowski resists this direction. The other associates computation with some aspect of the causal structure of the system. Within this direction it has recently been proposed to link computation with mechanisms, and computational explanation with mechanistic explanation (Piccinini 2007). Miłkowski adopts this mechanistic line too. He suggests that the notion of mechanism underlies both the ontology of physical computation and the explanatory role of computational approaches. Mechanisms here are understood in terms of the "new mechanistic philosophy of science", which characterizes a mechanism as a set of entities (components) and their associated activities whose organization and interaction is responsible for producing a given phenomenon; accordingly, a mechanistic explanation for a phenomenon or a behavior is one that specifies responsible mechanisms (see, e.g., Craver 2007).

Explaining the Computational Mind consists of five chapters. The first chapter introduces with great clarity four computational models of cognitive processes. Miłkowski convincingly demonstrates that there are varieties of computational approaches in current cognitive science, and that most of them are far removed from classical, digital and symbolic computation. The first example is the work of Newell and Simon on high-level cognitive heuristics. The second is the renowned Rumelhart and McClelland's connectionist model for the acquisition of past-tense English verbs (now almost 30 years old); the third is the Neural Engineering framework advanced by Chris Eliasmith that was applied to several neuro-cognitive functions such as navigation. The last one is the work of Barbara Webb on embodied robotics. The chapter gives rise to the impression that the computational approaches dominate the theory of cognition,

that the more current computational approaches are far more statistically-based, and that they increasingly focus on low-level cognitive functions that are closely tied to neural activity.

The second chapter addresses the question of what it is for a physical process to implement or realize a computation. Miłkowski's thesis consists of two claims. One is that computation is information processing, and the other is that a mechanistic approach best accounts for the physical implementation of information processing. Miłkowski does not provide an argument for the first claim, i.e., that computation is information processing. He also does not provide a precise characterization of information processing. What he does say regarding this notion is that it is non-semantic (p. 48), in that the inputs, outputs and perhaps internal variables need not refer to anything, and that it is very broad: "for my purposes the only thing that is important here is that the philosophical account of implementation should be compatible with any notion of mathematical computation used in computer science, mathematics, or logic" (p. 28).

Most of chapter 2 is dedicated to a careful and detailed mechanistic account of the notion of implementation. According to Miłkowski, a mechanism is a multi-level system. A *computational* mechanism consists of at least three levels of organization: contextual, isolated, and constitutive. In the cognitive setting, the contextual includes features in the environment, the isolated level "features the computational processes" (p. 55), and the constitutive level features structures that realize the computational processes; in this respect Miłkowski's account might remind us of other level accounts of cognition. The novelty of Miłkowski's picture is that the bottom, constitutive, level is essential to the notion of computational implementation. A salient

constraint on implementation is that structures of the (realizing) mechanism are identified in terms of non-computational ("constitutive") features.

The third chapter reviews three theories of explanation that might explicate computational explanations in cognitive science; these are the covering-law theory, the functionalist account (including Marr's tri-level account) and the mechanistic framework. Not surprisingly, the conclusion is that the mechanistic framework best elucidates computational models in cognitive science; the models introduced in the first chapter are taken as case studies.

Interestingly, Miłkowski does not discard the other two accounts as completely inadequate or useless. He suggests, rather, that each successive account of explanation might be seen as an improved version of the preceding one. For example, the mechanistic account is more robust than the functionalist account in that it imposes a series of causal constraints on computational processes.

In chapter 5 Miłkowski clarifies that computational explanations are not full-fledged mechanistic explanations of cognition (and, hence, that cognition is not identical to computation). They do not fully explain how computational mechanisms are embedded in the environment; this is sometimes explained by appeal to representations; the aim of chapter 4 is developing an account of representation for computing systems. In addition, computational explanations do not fully account for how the computation is physically implemented, as the implementation is partly identified in non-computational terms (as argued in chapter 2).

To what extent does Miłkowski's mechanistic account succeed in explicating the explanatory role of computation in cognitive science? I am not convinced that the proposed account is

superior to other accounts of computation, especially the semantic accounts which I favor that are, to some extent, a target of the book. I will mention here two concerns about Miłkowski's account, one pertains to the characterization of physical computation and the other to computational explanation.

Let us start with the characterization of physical computation. There are two related yet *different* problems that motivate the characterization of physical computation. One is the Putnam/Searle triviality results about universal implementation. Miłkowski's characterization mainly responds to this objection. He copes with the Putnam/Searle results by invoking constraints at the constitutive level. The other problem is this: There are physical systems that are being described by the same mathematical terms and yet one system computes whereas the other does not. Thus, to take Miłkowski's example, we can describe a (non-computing) mousetrap in the same mathematical terms we describe a simple computing system. This problem becomes acute if one holds a broad conception of computation that encompasses many dynamical systems as computing ones. Miłkowski (correctly!) considers many systems that are being described by dynamical equations as computing ones (pp. 2, 80-81, 194-195). He describes the rat navigation system as computing path integration even though its evolution is being described (in Eliasmith's model) by a set of dynamical equations. The problem is that many other systems that are described this way are not considered (under this description) as computing. The energy levels of a magnetic system (say its temperature) are very often described in terms of dynamical equations (replacing neural cells with particles). Yet, we do not say that the magnetic system computes its temperature; or if it does, then virtually every

physical system (when described this way) is computing. So why say that the rat's neural system computes but other systems do not?

One way to tackle this problem is to invoke a semantic feature. It can be said, for example, that the rat's brain carries semantic information about the rat's environment, but the particles do not carry information at all, or at least not about energy levels. Miłkowski also alludes to information (p. 2; p. 196), but given that his notion of information is non-semantic and very minimal (consistent with every mathematical model of computation), it is hardly clear whether it precludes the non-computing systems. At some point, Miłkowski writes that "to decide whether the mousetrap is a computer or not, we need to see how it is embedded in the environment and what role it plays" (p. 54). If so, it seems that the solution to the problem lies at his top, contextual, level, at which "the function the mechanism performs is seen in a broader context" (p. 55). Thus, the mousetrap and the magnetic system might be computing; whether they are or not depends on the relations of the mechanism implementing the mathematical structure with the environment. I agree with this move, but it should be supplemented with further conditions. Almost every system is embedded within an environment, yet most of them are not computing. The challenge, then, is to identify the system-environment relations that are pertinent to computing, without recourse to semantic features.

Another concern is about computational explanations. Miłkowski proposes that the mechanistic framework best accounts for computational processes. I think that he is right to observe that the isolated level, as well as the algorithmic and functional levels, can, and perhaps should, be

integrated into the mechanistic framework (see also Piccinini and Craver 2011). But one can challenge the assumption that the isolated level is the sole target of computational explanations. One might suggest, for example, that computational explanations essentially refer to contextual features (as to those features that might differentiate mousetraps from real computing systems). Indeed, David Marr (1982) famously proposes a tri-level framework for studying cognition that has some affinities to the one Miłkowski advances. For Marr, however, the computational level is the top one and pretty much parallels Miłkowski's contextual level. According to Marr, at least the way I understand him, computational-level theories are explanatory (Shagrir 2010). They display the function that the mechanism computes and they explain why this function is appropriate for a given cognitive task. The explanation locates the computed function in the broader, environmental context. Thus the rat's navigation system computes integration because the integration function captures certain velocity-position relations in the rat's movements in the environment.

Miłkowski discusses Marr's framework at some length (pp. 114 ff.). He says that Marr's computational level is spelled out as a purely functional analysis of the task (p. 118). But even if Miłkowski is right about this, the deficiency can be remedied by putting more constraints on the inter-level relation between the computational level and the lower levels. Another criticism is that environmental features at the computational level do not supervene on the "internal" features at the lower levels (p. 120). This, I think, is correct, but only makes Marr's proposal more interesting. Marr's computational level is not another level of organization nested in the algorithmic and neural levels, and his computational explanations are not sketches of algorithmic and neural explanations. Rather, computational descriptions explain why the

mechanisms described at the lower levels are appropriate to perform a cognitive task in the particular environment that the creature happens to live in. Miłkowski takes it (correctly, I think) that the task of the lower levels is to explain how the function displayed at the contextual level is being computed. But the task of computational-level theories is not just to display the function; its task is also to explain why computing this function has anything to do with the cognitive task that depends on the mind-world relation. Indeed, the advantage of Marr's framework, in my view, is in differentiating computational explanations from other, non-computational, explanations that also invoke mathematical models.

Despite these critical comments, I think that *Explaining the Computational Mind* is a substantial and excellent contribution to the growing literature on the foundations of computational cognitive neuroscience. On the one hand, it presents thorough arguments against extant accounts of computation, while, on the other, it advances a detailed and formidable positive account. *Explaining the Computational Mind* is thus a must-read piece that will have to be dealt with by anyone writing on computation in cognitive science.

Oron Shagrir
The Hebrew University of Jerusalem

References:

- Craver, C. F. 2007. *Explaining the Brain: Mechanisms and the Mosaic unity of Neuroscience*. New York: Oxford University Press.
- Marr, D. C. 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. San Francisco: W.H. Freeman.
- Piccinini, G. 2007. Computing Mechanisms. *Philosophy of Science*, 74: 501-526.
- Piccinini, G. and Craver, C. 2011. Integrating Psychology and Neuroscience: Functional Analyses as Mechanism Sketches. *Synthese* 183: 283-311.
- Shagrir, O. 2010. Marr on Computational-Level Theories. *Philosophy of Science*, 77: 477-500.