Metaphysics and Computational Cognitive Science: Let's Not Let the Tail Wag the Dog

Frances Egan

Rutgers University
fegan@rci.rutgers.edu

David Chalmers characterizes the central commitments of computational cognitive science in terms of two theses: *computational sufficiency*, the idea that the right kind of computational structure suffices for the possession of a mind, and *computational explanation*, the idea that computation provides a general framework for the explanation of cognitive processes and behavior. The computational program has been challenged by Hilary Putnam (1988) and John Searle (1991), who argue that every physical system implements every computation, with the consequence that any computational 'explanation' of cognition is utterly trivial. What is needed, according to Chalmers, is an account of implementation, which would both answer the Searle/Putnam challenge and provide a foundation for computational cognitive theorizing.

In this paper I argue that computational cognitive models typically do not satisfy Chalmers' notion of implementation, and so his account does not provide a conceptual foundation for computational theorizing as it is actually practiced. I argue further that the 'in-principle' possibility of deviant implementations of the Putnam/Searle sort does not undermine that practice – it does not make computational explanation trivial – though seeing why it doesn't requires that we take account of the use to which a computation is put in the exercise of a cognitive capacity.

Key words: cognition, computation, explanation, implementation, use

Journal of Cognitive Science 13: 39-49, 2012

1. Implementation

Chalmers spells out the key idea of implementation as follows:

> A physical system implements a computation when there exists a grouping of physical states of the system into state-types and a one-one mapping from formal states of the computation to physical state-types, such that formal states related by an abstract state-transition relation are mapped onto physical state-types related by a corresponding causal state-transition relation. (p.326)

The account of implementation is more fully specified in terms of the class of *combinatorial-state automata* (CSAs), whose internal states have a combinatorial structure, and are specified by a vector $[S^1, S^2, S^3, \ldots]$. The elements of the vector can be thought of as components ("substates") of the state, each corresponding to an independent element of the physical system. Inputs and outputs are characterized in an analogous way. State transition rules specify a function characterizing how each combination of new internal state-vector and output-vector depends on the old internal state-vector and input-vector.

As in the general case, the crucial requirement for implementing a CSA is that the causal structure of the physical system mirrors the formal structure of the computation. This is spelled out as follows:

> A physical system P implements a CSA M if there exists a vectorization of internal states of P into components $[s^1, s^2, \ldots]$, and a mapping f from the substates s^j into corresponding substates S^j of M, along with similar vectorizations and mappings from inputs and outputs, such that for every state-transition rule $([I^1, \ldots, I^k],[S^1, S^2, \ldots] \rightarrow ([S'^1, S'^2, \ldots], [O^1, \ldots O^l])$ of M: if P is in internal state $[s^1, s^2, \ldots]$ and receiving input $[i^1, \ldots i^n]$ which map to formal state and input $[S^1, S^2, \ldots]$ and $[I^1, \ldots, I^k]$ respectively, this reliably causes it to enter an internal state and produce an output that maps to $[S'^1, S'^2, \ldots]$ and $[O^1, \ldots, O^l]$ respectively. (p.329)

How does the account of implementation provide a framework for computational cognitive theorizing? It does so, according to Chalmers (p.323), by supporting two theses that characterize the foundational role of computation: (1) a computational description is an abstract characterization of the causal organization of the system; and (2) mental properties are causal invariants.[1] In other words, cognition, in fact mentality in general, depends upon causal invariants, and a computational description *just is* the appropriate description of these causal invariants. These theses constitute what Chalmers calls *minimal computationalism*, which he claims is all the foundation that computational cognitive science needs.

Let us focus first on Chalmers' account of implementation. In the next section we will turn to the foundational claims it is alleged to support.

A preliminary point is perhaps obvious, though worth mentioning explicitly. For an application of the account to have any bite, the two levels of analysis – the computational and the physical – have to be *independently specified*. In particular, the relevant causal structure, on the physical side, needs to be characterized independently of the formal, computational structure it is supposed to implement. For complex physical systems such as ourselves, isolating the relevant causal organization – relevant for *cognition* – from the mess of physical detail is likely to be very difficult. Computational cognitive science is committed to the idea that it can be so isolated, at least in principle (although in actual practice it has often been unable to make good on the commitment). I shall return to this point below, where I argue that computational cognitive science is committed to the project of characterizing that causal structure *function-theoretically*.

According to Chalmers' account, CSAs provide a suitable formalism "for our purposes", viz., answering the Putnam/Searle challenge and providing a foundation for computational cognitive explanation. One reason to specify the account in terms of CSAs is that the implementation conditions on

[1] (1) and (2) are reformulations of the claims that Chalmers dubs '(a)' and '(b)' on p.323. *Computational sufficiency* – the idea that the right kind of computational structure suffices for the possession of a mind – and *computational explanation* – the idea that computation provides a general framework for the explanation of cognitive processes – are claimed to be consequences of (a) and (b).

CSAs are highly constrained:

> An implementation of a CSA is required to consist in a complex causal interaction among a number of separate parts; a CSA description can therefore capture the causal organization of a system to a much finer grain [than a *finite-state automata* description]. (p.329)

Another reason for employing the CSA framework is its *generality* – it provides a unified account of implementation conditions for both finite and infinite machines, and, with minor tweaking, for nondeterministic and probabilistic automata.

Chalmers presumes that the appropriate formal characterization of the causal organization of the human mind will be a very complex CSA description with lots of input and internal-state parameters. As he points out, the requirement that states of the physical system satisfy reliable state-transition rules is what does the work in ruling out Putnam's and Searle's trivial implementations. The chance that an arbitrary physical system would satisfy these constraints is vanishingly small.

I shall assume for the sake of argument that Chalmers' account succeeds at the *metaphysical* project of providing sufficient conditions for implementing a computation – conditions not open to Putnam/Searle trivialization. The constraints on implementing a complex CSA with significant combinatorial structure appear to be sufficiently stringent that they can't be trivially satisfied. This would establish the theoretical possibility of a computational account of human cognition and behavior. But it provides a foundation for computational theorizing only if computational models of cognition typically satisfy the sufficient condition. I argue that they typically do not.

It is worth emphasizing that whether or not the CSA formalism is the appropriate computational characterization for explaining cognition is an *empirical* question. Chalmers would not deny this. He presumes that as a matter of contingent, empirical fact the human mind is a very complex CSA with significant compositional structure.[2] I'm sure he is right about

[2] What is not contingent is that *given this computational structure* we have the

that. However, if our concern is with computational *explanation* as it functions in actual computational practice, there is little reason to think that the appropriate formal characterization will be in terms of CSA.

A look at some representative examples of computational models of cognitive capacities supports the point. Marr's (1982) theory of early vision explains edge detection by positing the computation of the Laplacian of the Gaussian of the retinal array. Ullman (1979) describes the visual system recovering the 3D structure of moving objects by computing a function from 3 distinct views of 4 non-coplanar points to the unique rigid configuration consistent with the points. Shadmehr and Wise's (2005) computational account of motor control explains how a subject is able to grasp an object in view by computing the displacement of the hand from its current location to the target location, i.e. by computing vector subtraction. In a well-known example from animal cognition, Gallistel (1993) explains the Tunisian desert ant's impressive navigational abilities by appeal to the computation of the displacement vector to its nest from any point along its foraging trajectory.

In none of these cognitive models is the computational characterization a CSA description. The posited computation has little or no combinatorial structure. Rather, the explanatory strategy might be described as 'function-theoretic,' in the sense that the model explains the cognitive capacity in question by appeal not to some arcane, highly complex formal structure, but rather to an independently well-understood mathematical function under which the physical system is subsumed. In other words, what gets computed, according to these computational models, is the value of a mathematical function (e.g. addition, vector subtraction, the Laplacean of the Gaussian, a fast Fourier transform) for certain arguments for which the function is defined. For present purposes we can take functions to be mappings from sets (the arguments of the function) to sets (its values). The theory may go on to propose an *algorithm* whereby the computation of the value of the function(s) is effected.[3]

---

cognitive capacities that we do have.

[3] Chalmers claims (p.330) that combinatorial-state automata provide a basis for a general account of implementation, arguing that we can re-describe other

The upshot is that CSA formalism is not generally appropriate for characterizing computational cognitive science as it is actually practiced. But how does that practice stand with respect to the Putnam/Searle challenge? Are the computations posited in the above models implemented by arbitrary physical systems, and if so, does it follow that the explanations of cognition afforded by these models are trivial? I doubt that 'deviant' implementations can be conclusively ruled out, but this does not make computational explanation trivial. The argument will require some setting up.

I have argued that computational models do not typically posit computations with complex compositional structure; however, the physical system must still satisfy reliable state transition rules that require that when the system is in the physical state(s) that (under the interpretation imposed by the computational description) realizes the arguments of the function, it goes into the physical state that (under interpretation) realizes the value of the function, for all the arguments for which the function is defined.[4] These conditionals have modal force. For the sorts of functions discussed above, satisfying this condition will require significant *physical* structure. The causal organization of the neural mechanism that computes the structure-from-motion function, for example, is likely to be quite complex.[5]

---

computational formalisms in CSA terms. He explicitly mentions Turing machines, cellular automata, and non-deterministic and probabilistic automata. However, there is no reason to think that arbitrary machines and CSAs will have the same complexity profiles. A re-description of Ullman's structure-from-motion device in CSA terms will impose structure that is not really there.

[4] Not just, as in Putnam's example, for the arguments that happen to be realized during the specified time period.

[5] More generally, a theory characterizing the neural implementation of a computational model is likely to posit systems of neurons that compute the function. Recall the earlier point that the computational and physical levels must be independently specified. Individual neural structures will often receive no computational interpretation, thus there will be neural complexity without corresponding computational complexity. It follows that Chalmers' gloss (the "short answer") on his account of implementation – "A physical system implements a given computation when the causal structure of the physical system mirrors the formal structure of the computation" (p.326) – is not generally correct. A complex causal structure may be needed to implement a function with little or no formal

I shall have more to say below about constraints on implementing computational models, but for now let me emphasize that there is nothing approaching a *formal demonstration* that the structure-from-motion computation or vector subtraction (employed by Shadmehr and Wise's motor control system) do not have 'unintended' implementations. However, there is no reason to think that the possibility of such implementations renders the explanations that computational theories do provide trivial or otiose. To see why not we must again focus on actual practice.

The epistemological context in which computational models are developed is instructive. We don't start with a computational description and wonder whether a given physical system implements it. Rather, we start with a physical system such as ourselves. More precisely, we start with the observation that a given physical system has some cognitive competence – the system can add, or it can understand and produce speech, or it can see the three-dimensional layout of its immediate environment. Thus, the *explanandum* of the theory is a manifest cognitive capacity. The computational theorist's job is to explain how the physical system accomplishes the cognitive task. There is, of course, no evidence that Putnam's rock or Searle's wall has *any* cognitive capacities.

With the target capacity in view, the theorist hypothesizes that the system computes some well-defined function (in the mathematical sense), and spells out how computing this function would explain the system's observed success at the cognitive task. Justifying the computational description requires explaining how computing the value of the function contributes to the exercise of the cognitive capacity. For example, computing the Laplacean of the Gaussian of the retinal array produces a smoothed output that facilitates the detection of sharp discontinuities in intensity gradients across the retina, and hence the detection of significant boundaries in the scene. In other words, the computational description is justified by reference to the use to which the computation is put in the exercise of a manifest cognitive capacity.

Computational theorizing is constrained from above, as it were, by data about the performance of the system, and from below, by knowledge of

---

complexity.

available neural hardware. The computational hypothesis may predict a pattern of error that the system is prone to make in its normal environment. Observation of the system's successes and failures at the cognitive task, or discoveries about available neural hardware, may lead the theorist to revise her initial computational hypothesis. Perhaps the device computes the function only for a more restricted domain than initially thought.[6] Algorithms may be suggested for how the function is computed, but ultimately these need empirical motivation.

I appealed to the notion of use in considering how the computational description is justified. Appeal to use also helps constrain implementation. As Chalmers points out, a computational description is an abstract specification of the causal organization of the system. But as I noted above, isolating the causal organization responsible for *cognition* from the mess of physical detail (including the organization responsible for maintaining other bodily functions) is likely to be very difficult. Appeal to use can help here. Let me elaborate.

Take a simple adder. The physical states that (under interpretation) realize the addends and the physical states that (under interpretation) realize the sum stand in a causal-transition relation. In other words, the former states cause the system to go into the latter state (possibly with a significant number of intermediary physical states). In order for the system to be used to compute the addition function these causal relations have to hold *at a certain level of grain*, a level that is determined by the discriminative abilities of the user. That is why, whatever the status of Putnam's argument, no money is to be made trying to sell a rock as a calculator. Even if (*per mirabile*) there happens to be a set of state-types at the quantum-mechanical level whose causal relations do mirror the formal structure of the addition function, microphysical changes at the quantum level are *not* discriminable by human users, hence human users could not use such a system to add. (God, in a playful mood, could use the rock to add.)

The same point holds for natural computational mechanisms and the neural processes that employ them. Neural processes, it is reasonable to assume,

[6] Just as the size of the display requires restricting the arithmetical functions that a hand calculator can be said to compute.

are not sensitive to (merely) quantum-level changes, such that their behavior could be conditioned on such changes. Relatively gross (or macro-level) changes are required for the central nervous system to use a computational mechanism (in particular, to use its *outputs*) to accomplish a cognitive task. In other words, the appropriate level of grain at which the causal organization of a computational device must be specified is relative to the discriminatory capacities of the systems using it. The theorist attempting to specify the neural structures that implement a computation can (and must) look to the processes that use them.[7]

In summary, appeal to use can both help isolate the relevant causal organization responsible for a cognitive capacity and distinguish substantive computational hypotheses from deviant cases.

2. Minimal Computationalism

Chalmers' account of implementation is said to support two theses that together serve as the foundation for computational cognitive science: (1) a computational description is an abstract characterization of the causal organization of the system; and (2) mental properties are causal invariants. Chalmers dubs the view that emerges from his elaboration and defense of these theses *minimal computationalism*, which he elaborates as follows:

> Minimal computationalism is compatible with such diverse programs as connectionism, logicism, and approaches focusing on dynamic systems, evolution, and artificial life…. All such theories are theories of causal organization, and computation is sufficiently flexible that it can capture almost any kind of organization, whether the causal relations hold between high-level representations or among low-level neural processes. Even such theories as the Gibsonian theory of perception are ultimately compatible with minimal computationalism. If perception

[7] For perceptual mechanisms, of course, there are additional sources of constraint on the input side. The theorist trying to characterize the neural implementation of edge detection must look for structures that are differentially sensitive to changes in light intensity.

> turns out to work as the Gibsonians imagine, it will still be mediated by causal mechanisms, and the mechanisms will be expressible in an appropriate computational form. (pp.354-355)

Minimal computationalism, so understood, appears to involve little more than a commitment to the idea that mental processes are causal processes.[8] Indeed, if Gibson's theory of perception counts as computational *simply* because it holds that perception is "mediated by causal mechanisms" then doorbells and mousetraps count as computational devices too; their operations are mediated by causal mechanisms which are expressible in some computational form.[9] It is another matter whether the computational description of a standard-issue doorbell or spring mousetrap is *explanatory* of the device, but the same question could be asked of the mechanisms posited in a Gibsonian theory of perception.

Chalmers takes it to be a virtue of his framework that minimal computationalism is unlikely to be falsified by empirical discoveries about the mind. But in casting his net so widely he has lost what is distinctive about computational explanation as it figures in cognitive science.[10]

I am certainly not denying that a computational description is an abstract

---

[8] This construal of Minimal Computationalism is given independent support by the discussion in section 3, where Chalmers introduces the notion of *causal topology* (the abstract causal organization of the system) and argues that mental properties depend only on causal topology, and that computational descriptions capture causal topology. Thanks to an anonymous referee for pointing this out.

[9] It is unclear how to understand "*appropriate* computational form" in the above quote. Perhaps Chalmers means 'expressible in CSA formalism', but there is no reason to assume that Gibsonian mechanisms can be so characterized, at least not without introducing unmotivated structure. (Gibson himself insists that they should be treated as 'black boxes', from the perspective of psychology, though from a physiological perspective they may be quite complex.) See fn. 3 above.

[10] It is hard to reconcile the very weak thesis Chalmers endorses as a foundation for the computational study of cognition in the last part of the paper (*minimal computationalism*) with the demanding requirements levied on implementation earlier (requiring complex combinatorial formal structure). The account of implementation is also claimed to play a key foundational role, but the relation between it and minimal computationalism is not transparent.

characterization of the causal organization underlying cognition. But I want to stress that a computational description is a *particular sort* of abstract characterization of that causal organization. Computational theorists share a commitment about the form that the specification of relevant causal structure should take. As I argued above, a computational characterization is a *function-theoretic* characterization of a mechanism. It characterizes the causal organization of the system by specifying the function (in the mathematical sense) that the system computes. This characterization is general enough to subsume connectionist devices, finite-state machines, CSAs, cellular automata, dynamic systems, and so on – precisely the systems that we tend to think of as *computers* – but it is not fruitfully applied to doorbells and mousetraps.[11]

References

Chalmers, D. 2011. "A Computational Foundation for the Study of Cognition." *The Journal of Cognitive Science* 12: 323-357.

Gallistel, C.R. 1993. *The Organization of Learning*. Cambridge, MA: MIT Press.

Marr, D. 1982. *Vision*. New York: Freeman Press.

Putnam, H. 1988. *Representation and Reality*. Cambridge, MA: MIT Press.

Searle, J.R. 1991. *The Rediscovery of the Mind*. Cambridge, MA: MIT Press.

Shadmehr, R. and Wise, S. 2005. *The Computational Neurobiology of Reaching and Pointing: A Foundation for Motor Control*. Cambridge, MA: MIT Press.

Ullman, S. 1979. *The Interpretation of Visual Motion*. Cambridge, MA: MIT Press.

[11] Thanks to an anonymous referee for helpful comments on an earlier version of this paper.