Published in *Studies in History and Philosophy of Science* (2010) 41: 223–226.
doi:10.1016/j.shpsa.2010.07.011

mark.sprevak@ed.ac.uk

# Computation and cognitive science: Introduction to Special Issue

Mark Sprevak
*University of Edinburgh*

24 July 2010

Nowadays, it has become almost a matter of course to say that the human mind is like a computer. Folks in all walks of life talk of 'programming' themselves, 'multi-tasking', running different 'operating systems', and sometimes of 'crashing' and being 'rebooted'. Few who have used computers have not been touched by the appeal of the idea that our inner workings somehow resemble computing machines. The success of modern day computational psychology appears to bears witness to the explanatory and predictive pay-off in positing a connection between computers and minds. Among its other virtues, the computational framework has rendered theorising about inner processes respectable, it has provided a unified and naturalistic arena in which to conduct debates about psychological models, and it provides the tantalising possibility of accurately simulating and reproducing psychological processes.

There is almost universal agreement that the mind is *in some sense* like a computer. But consensus quickly ends once we ask *how*. Even after more than thirty years of model building, and a wealth of empirical work, surprisingly little consensus exists on the correct answer to this question. What is more, disagreement tends to lie at a relatively fundamental level. There is little agreement about the content of the notion of computation, what it means for a physical system, like the brain, to implement a computation, the broad-brush computational architecture of the mind, or how computational models fit with other models of the mind, such as control theoretic models, statistical models, or dynamical systems theory.

This volume targets these questions. The contributors analyse the role that computation plays in cognitive science, and the implications, based on our current evidence, this has for the architecture of human psychology. The intention is not only to secure firmer ground for contemporary cognitive science, but also to envision cognitive science's next steps.

Let us take up the questions above. First, what exactly do we mean by *computation*, and how does the notion of computation differ from related notions, such as information processing? Is there a single notion of computation, or should we be pluralists about computation? Second, what does it mean to say that a physical system, like the human brain, *implements* a computation? How is representational content involved in implementation, if at all? What are the necessary conditions for implementation to obtain? Under what conditions do two systems, e.g. an electronic PC and a human brain, implement the same computation? Third, granted that one can agree about implementation, what evidence can one bring to bear to determine the computational architecture of the human mind? How do we assess the merits of a computational model, and which computational architectures can we already, based on our current evidence, rule in or out? Fourth, how do other approaches to explaining cognition, such as statistical theory, control theory, and coupled-oscillator models, relate to computational models of the mind? Are they rivals to explaining the mind in terms of computation, or themselves types of computation?

The papers in this volume fall into four groups, corresponding to the four groups of questions above:

1. Distinguishing computation from related notions (e.g. information processing)
2. Theories of implementation of computation
3. Computation at work in cognitive science
4. Projected successors to the notion of computation in cognitive science

## 1 Delimiting the notion of computation

The first group of papers attempt to describe what we mean by *computation* in cognitive science and contrast it with related notions. Is computation the same as information processing? Does the concept of computation have the same content in all fields? What is the difference between genuine computations and computations that rely on an observer to do the computational work?

In the first paper, Ken Aizawa argues that the notion of computation has a more fragmented nature than is generally supposed. Orthodox histories of cognitive science tell of a special role played by the notion of computation developed by Alan Turing and others in mathematical logic. These accounts emphasise the exportation of a 'Turing-equivalent' notion of computation into neuroscience, cognitive science, AI, and cybernetics, and the subsequent computational revolutions caused in those fields. Aizawa argues that this simple exportation analysis is mistaken. What is meant by 'computation' in different fields depends largely on the local theoretical goals and interests of those fields. One should not expect, and will not find, a single fundamental notion of computation that is shared in each and every domain of enquiry. Rather than telling a history of how the Turing-equivalent notion was exported from mathematical logic, Aizawa counsels instead to analyse a target use of 'computation' that has been tailored to a particular end (say, neuroscience, cognitive science, etc.). The question a historian and philosopher should ask is not how a general notion of computation is deployed across all fields, but what is the distinctive theoretical contribution that a particular use of the term 'computation' plays in each discipline. Aizawa considers various ways in which these specific notions of computation can diverge from the Turing-equivalent notion. For example, they might relax Turing's finiteness conditions, collapse computation into the notion of information processing, or emphasise a distinction between computing and merely reproducing a signal.

Gualtiero Piccinini and Andrea Scarantino argue that the notion of computation can be distinguished from the notion of information processing. Piccinini and Scarantino distinguish between a number of different varieties of computation and information processing. Among computations, they distinguish between *digital* and *generic* computation. *Digital computation* involves rule-governed processing of (i) discrete symbols, which (ii) yield input-output behaviour equivalent to that of some Turing machine. *Generic computation* relaxes constraints (i) and (ii) and so allows for analog computers and hypercomputation. Piccinini and Scarantino divide information processing into two kinds: processing of *Shannon information* and processing of *semantic information*. Shannon information is a thin use of 'information': it is no more than a measure of the probability of a certain event occurring out of a defined domain of possible outcomes. Shannon information does not require the event have meaning or semantic content (a meaningless message like '@X;D' can carry just as much Shannon information as 'The battle is won'). *Semantic information* accords more with the commonsense meaning of the term. Messages must, in addition to having a bare probability of occurrence, be individually meaningful. Piccinini and Scarantino argue that while computations in cognitive science *often* involve information processing of either the Shannon or semantic kind, they need not necessarily do so. The assumption that computation is always information processing has closed off potentially productive lines of research.

Jack Copeland and Diane Proudfoot ask about the limits that should be placed on the notation systems in a computation. All computations involve some notation system, but not any notation system is permissible. Some notations, such as binary ('0' denotes 0, '1' denotes 1, '10' denotes 2, etc.) are kosher. But others allow Turing machines to compute 'incomputable' functions. Imagine a Turing machine that outputs the sequence: '1 2 3 …'. Consider the following notation: the *n*th digit of the output of a Turing machine means 'The *n*th Turing machine halts' just in case the *n*th Turing machine halts; otherwise, the output means 'The *n*th Turing machine does not halt'. Relative to this notation, the Turing machine above solves the halting problem, a supposedly incomputable task. In order not to trivialise the concept of a computable function, some limit must be placed that blocks deviant notations. A natural thought is to allow only those notation systems that do not themselves require solving an incomputable task to use. This has the virtue of ruling out the halting-problem notation above, but the cost of being circular—it presupposes that a deviant notation system is not being used in order to describe which tasks are and are not computable. Copeland and Proudfoot argue that an alternative reply is available that avoids this circularity worry. Turing's original discussion of Turing machines offers two natural and simple restrictions that block deviant notation systems.

## 2 Implementing a computation

This group of papers concern how mathematical computation relates to the nuts and bolts of physical systems. Computation is a notion with two faces: one side concerns computers as abstract mathematical entities, the other concerns computers as physical machines. What relates these two types of entity is the implementation relation. But how does implementation work? What are the necessary and sufficient conditions for a physical system to implement a computation? How does implementation relate to representational content? The papers in this section put forward theories of implementation. The disagreement between contributors tends to lie in the particular ways that representational content should or should not enter into computational identity.

Frances Egan targets what she calls the 'standard view'. The standard view says that the representations involved in cognitive computations have essential distal representational content. For example, the computational states involved in David Marr's 'edge-detection' computation in the human visual system involve representations of distal features of the subject's environment: *edges*, *light intensities*, etc. Egan claims that it is wrong to think that this distal content is relevant to the computation that the system performs. Computations are individuated by the *mathematical function* they compute, not by the distal content they possess. The correct description of Marr's edge-detection computation is that it computes the mathematical function $\nabla^2 G * I$. It is irrelevant to the computation that its inputs represent light intensities, and its outputs represent changes in light intensities. Devices in different environments, or in different parts of the organism, can, and often do, perform the same computation (e.g. a fast Fourier transform), even if they manipulate different distal content (e.g. one mechanism may process auditory signals, the other visual signals). Egan suggests that distal content plays a non-computational role in psychology. On her view, distal content ascriptions provide an 'explanatory gloss' on the purely mathematical computation. The gloss is a bridge between the abstract mathematical computation and the environmental task that the computation was invoked to explain: e.g. How do we detect *edges* in the world? How do we perceive *depth* from two-dimensional input? A gloss links the mathematical description to the relevant environmental content. Egan claims that there is no science of how to develop these explanatory glosses. She argues that explanatory glosses are not part of computational psychology proper, although they are part of the informal motivation and explanatory deployment of that theory. One consequence of her view is that cognitive science is freed from a commitment to a naturalistic account of distal representation—ascription of distal content lies outside the realm of natural science.

Mark Sprevak argues for a more catholic view about the representational content involved in computation. Sprevak claims that there are no serious constraints on the type of representational content that can determine computational identity (representational content may be distal, proximal, mathematical, narrow, broad, etc.). To this end, he focuses initially on Egan's ban on distal content. Sprevak argues that, with a combination of two strategies, distal content can be accommodated just as well as Egan's mathematical content in determining computational identity. Indeed, a mixed position better explains the pattern of our judgements about computational identity than a restriction to mathematical content. Sprevak then turns to Fodor's 'formality condition' that has been taken to justify a restriction to narrow representational content in computations. Sprevak claims that Fodor's argument rests on a mistaken privileging of computational states over processes. Once computational processes take centre-stage, broad content can be seen to be just as capable of determining computational identity as narrow content. In the final section, Sprevak sketches three arguments for why computation has to involve representational content at all. The first argument is based on paradigm cases of computation. The second argument is based on the diverse physical and functional character of physical systems that perform the same computational task. The third argument is based on the need for any adequate notion of computation to draw certain distinctions, such as the distinction between AND and OR, which are only visible after one appeals to representational content.

Oron Shagrir puts forward a different account of implementation based on the notion of *analog computing*. Shagrir argues that the notion of analog computing does not involve any restriction to continuous variables. Rather, analog computing signifies that the states of a computer undergo the same transformation as the states in the environment that they are taken to represent. An analog computer simulates a target system in a specific sense: the computer's physical states stand in the same mathematical relations as the target states that the computer's states represent. For example, cells in the V1 area of the brain *analog compute* just in case the mathematical relationship between the electrical properties of V1 cells (e.g. their firing rates), is the same as the mathematical relationship between what firing of those cells represents (e.g. light intensities). Shagrir considers three objections to his account. First, the proposal is not well-formed since the represented entities are typically different in kind from the representing entities, and so cannot stand in the same mathematical relation. Second, the proposal cannot accommodate certain types of computational processes (e.g. planning) because they do not involve transformation of existing states. Third, the proposal is too liberal because it permits any system to simulate any other. Shagrir's replies draw heavily on the representational nature of computation, and like Sprevak, Shagrir argues that the content involved may be of almost any kind (mathematical, distal, existing, non-existing, etc.).

## 3 Computation at work in cognitive science

Computational psychology faces a steep underdetermination challenge. Often the only way we test models in computational psychology is by employing behavioural evidence, and such behavioural evidence confounds many psychological aspects of the human subject. Teasing out how the behavioural evidence bears on specific proposals about internal computations requires a great deal of care. The contributors in this section consider how empirical evidence in cognitive science supports or undermines competing computational models.

Richard Samuels defends a classical computational account of human reasoning and belief revision. Cognitive science to date has made disappointingly slow progress in modelling central thought processes like inductive inference and belief revision in a way that scales beyond a highly restricted domain of beliefs. Rapid progress, in contrast, has been made on computational models of peripheral cognitive processes, such as visual perception and motor control. This pattern of explanatory failure has been explained by Fodor as arising from a fundamental limitation of computational architectures. Samuels analyses a cluster of reasons why classical architectures have been perceived to be inadequate to central reasoning: (i) the *frame problem*; (ii) the perceived *holistic* nature of rational inference (*viz*. any belief may be relevant to the epistemic evaluation of any other belief); (iii) the perceived *global* nature of rational inference (*viz*. evaluating certain properties of beliefs, like simplicity and conservativeness, requires examining all one's other beliefs). Samuels argues that none of these features in fact decisively tell against a classical computational account of reasoning. Rival non-computational, or non-classical computational, accounts fare equally badly, if not worse, at accommodating these properties of reasoning. Samuels puts forward an alternative explanation of the observed pattern of success and failure in cognitive science: that pattern stems from the epistemic inaccessibility of central processes in contrast to the relative accessibility of peripheral systems like perception and motor control to testing. One should expect belief management to be a hard problem on any architectural hypothesis—our lack of progress in explaining it does not specially disconfirm a classical computational architecture.

Daniel Weiskopf examines two rival computational hypotheses about linguistic understanding. The first hypothesis, the *embodied* hypothesis, claims that linguistic understanding *reuses sensory and motor representation*. According to this view, linguistic comprehension consists in experiential simulation (recreation) of the described situation using existing sensorimotor representations and capacities. The second hypothesis, the *traditional* hypothesis, claims that the representations in linguistic understanding are unique, amodal and *distinct from sensory and motor codes*. According to the second hypothesis, linguistic understanding is a distinct computational module, separate from sensory and motor systems, whose job it is to create an amodal semantic representation of the truth conditions of the linguistic input. The two hypotheses offer rival computational architectures for linguistic understanding. Weiskopf argues that the empirical evidence cited in favour of the embodied hypothesis fails to support it, and that the traditional hypothesis better accounts for the data. Empirical data for the embodied hypothesis consists largely in *compatibility effects*: the observation that sensory and motor processing prime, or inhibit, linguistic understanding (and vice versa). Weiskopf argues that compatibility effects can be readily explained on the traditional hypothesis by saying linguistic comprehension is *causally related* to sensorimotor processing, without assuming, as the embodied hypothesis does, that the latter *constitutes* the former. Weiskopf accuses advocates of embodied cognition of a vehicle/content fallacy when reasoning about how we bring information about our sensorimotor capacities to bear. A cognitive system may use representations that have a sensorimotor *content* in its linguistic processing, but that says nothing about the nature of the *format* of those representations: the representations may be those employed by the sensorimotor system, or they may be unique amodal representations that happen to have the same sensorimotor content.

Raymond Gibbs and Marcus Perlman reply to Weiskopf by raising three points in favour of the embodied view. First, Weiskopf takes too thin a view of experiential simulation. Once simulation is understood to involve, not just visual and motor representations, but also emotional responses, then the embodied hypothesis can deal with certain counterexamples that Weiskopf raises. Second, evidence from cognitive neuroscience demonstrates activation of motor cortex and pre-motor cortex during linguistic comprehension. Gibbs and Perlman ask why these brain areas would be automatically activated unless motor simulation was part of linguistic understanding. Third, Gibbs and Perlman claim that the traditional view is committed to an implausible two-stage model of language processing: first, literal *truth-conditional meaning* is computed, then the *speaker's meaning*—which may be figurative or metaphorical—is computed by combining truth-conditional meaning with knowledge of the wider context and pragmatics. Gibbs and Perlman argue that if linguistic understanding occurs in this way, then we would be slower to understand figurative meanings than literal ones. But we often comprehend figurative meaning faster than literal meaning. Gibbs and Perlman claim that grasping figurative meaning requires sensorimotor representations, and computing literal content cannot be separated from this process. Therefore, no part of language comprehension escapes sensorimotor involvement.

Weiskopf has a quick rejoinder. First, he claims that even on a richer notion of simulation, counterexamples still arise to the embodied hypothesis's treatment of compatibility effects. Second, the neural imaging evidence about the activation of motor representations during language comprehension is equivocal. Third, the traditional hypothesis is not committed to a two-stage literal/figurative model of language processing. The traditional hypothesis is only committed to the existence of a capacity to understand sentences abstracted from their possible contexts of use. This capacity underlies semantics: theorising about properties like entailment, compatibility, contradiction, synonymy, and so on. Furthermore, Weiskopf argues that even if sensorimotor representations do constrain figurative understanding in the way that Gibbs and Perlman suggest, that is compatible with sensorimotor representations having a merely causal rather than a constitutive role in comprehension.

## 4 Successors to the notion of computation in cognitive science

Computational models cast a long shadow over cognitive science but they are not the only theories on the market. Other approaches—dynamical systems theory, statistical models, control theory, coupled-oscillator models—also enjoy predictive and explanatory success. Are these models genuine alternatives to a computational approach, or are they different types of computation? What advantages do these models have over traditional computational models? Why do some psychological processes appear more apt to be described by certain models than others? Is there a single theoretical framework that can unite all approaches to cognition? The contributors in this section explore the properties of alternatives to traditional computational models in cognitive science.

Chris Eliasmith examines four approaches to cognitive and brain function: traditional computational models, dynamical systems theory, statistical models, and control theory. He argues that control theory provides the best framework for understanding mind and brain function. Eliasmith argues that other approaches are ultimately heuristics along the way to the unified framework offered by control theory. Control theory provides a quantitative approach for explaining brain function that stretches all the way from models of single neurons, to neural populations, to high-level cognitive processes like language comprehension and inference. Eliasmith compares the merits of the different approaches along a single metric for the *best quantitative description* of cognition. According to this metric, a description should: (1) provide a simple mapping from experimental data to descriptive states; (2) decompose the system into its component mechanisms; (3) posit mechanisms that support interventions; (4) explain the effects of noise and variability on the system. Eliasmith argues that control theory fares better than rival approaches on these requirements, and thereby promises to best describe brain function. Control theory may also be glossed as positing *computations* in a wider sense: systematic manipulation of representations.

William Bechtel and Adele Abrahamsen highlight *coupled-oscillator models*. They analyse recent research on circadian rhythms and consider how models developed in that context could inform, and ultimately provide a new paradigm for, research in cognitive science. Typically, explanations in cognitive science focus on finding the *component parts* and *operations* of a cognitive mechanism. Bechtel and Abrahamsen argue that it is less common to find emphasis placed on the *dynamics* of how those parts and operations relate, and how distinct cognitive mechanisms couple to produce complex effects. Bechtel and Abrahamsen call this latter kind of explanation *dynamic mechanistic explanation*. In circadian rhythm research, typically the component parts and the operations are known by laboratory work on the composition of the relevant cells. The challenge is to show how those parts and operations are dynamically orchestrated to produce the observed phenomena. In cognitive science, the component parts and operations of the mechanisms, as well as their dynamics, are largely unknown. This places computational modelers in a more precarious epistemic position, and often dramatically underdetermines their proposals, leaving competition between rival proposals unwinnable (cf. classical vs. connectionist debates over mechanisms for past tense formation). The solution, according to Bechtel and Abrahamsen, is to demand more detailed specifications of the mechanisms that make predictions about the neurophysiology responsible for a given cognitive process. That would allow empirical evidence from neuroscience to be brought to bear to validate the models. Once we have a better grip on the parts and operations that realise a cognitive mechanism, then the road is open for more sophisticated models that use the dynamics of those parts and operations to explain how cognition is produced by, for example, coupling relations.

## Acknowledgements

A number of papers in this volume developed out of papers presented at a conference on Computation and Cognitive Science held in King's College, Cambridge, 7–8th July 2008. I would like to thank all the participants and speakers for making the conference such a productive and enjoyable occasion. I would also like to thank the sponsors of the conference: the British Academy, the British Society for the Philosophy of Science, King's College, Cambridge, the Mind Association, and Microsoft Research.