University of Wollongong

Research Online

Faculty of Engineering and Information
Sciences - Papers: Part A

Faculty of Engineering and Information
Sciences

1-1-2012

The explanatory role of computation in cognitive science

Nir Fresco

University of New South Wales, nfresco@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/eispapers>

Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

The explanatory role of computation in cognitive science

Abstract

Which notion of computation (if any) is essential for explaining cognition? Five answers to this question are discussed in the paper. (1) The classicist answer: symbolic (digital) computation is required for explaining cognition; (2) The broad digital computationalist answer: digital computation broadly construed is required for explaining cognition; (3) The connectionist answer: subsymbolic computation is required for explaining cognition; (4) The computational neuroscientist answer: neural computation (that, strictly, is neither digital nor analogue) is required for explaining cognition; (5) The extreme dynamicist answer: computation is not required for explaining cognition. The first four answers are only accurate to a first approximation. But the "devil" is in the details. The last answer cashes in on the parenthetical "if any" in the question above. The classicist argues that cognition is symbolic computation. But digital computationalism need not be equated with classicism. Indeed, computationalism can, in principle, range from digital (and analogue) computationalism through (the weaker thesis of) generic computationalism to (the even weaker thesis of) digital (or analogue) pancomputationalism. Connectionism, which has traditionally been criticised by classicists for being non-computational, can be plausibly construed as being either analogue or digital computationalism (depending on the type of connectionist networks used). Computational neuroscience invokes the notion of neural computation that may (possibly) be interpreted as a sui generis type of computation. The extreme dynamicist argues that the time has come for a postcomputational cognitive science. This paper is an attempt to shed some light on this debate by examining various conceptions and misconceptions of (particularly digital) computation.

Keywords

cognitive, explanatory, science, role, computation

Disciplines

Engineering | Science and Technology Studies

Publication Details

Fresco, N. (2012). The explanatory role of computation in cognitive science. *Minds and Machines: journal for artificial intelligence, philosophy and cognitive sciences*, 22 (4), 353-380.

The Explanatory Role of Computation in Cognitive Science

Nir Fresco

Abstract. This paper deals with the question: which notion of computation (if any) is essential for explaining cognition? Five answers are discussed in the paper.

1. The classicist answer: symbolic (digital) computation is required for explaining cognition.
2. The broad digital computationalist answer: digital computation broadly construed is required for explaining cognition.
3. The connectionist answer: sub-symbolic computation is required for explaining cognition.
4. The computational neuroscientist answer: neural computation (that, strictly, is neither digital nor analogue) is required for explaining cognition.
5. The extreme dynamicist answer: computation is not required for explaining cognition.

The first four answers are only accurate to a first approximation. But the “devil” is in the details. The last answer cashes in on the parenthetical “if any” in the question above. The classicist argues that cognition is symbolic computation. But digital computationalism need not be equated with classicism. Indeed, computationalism can, in principle, range from digital (and analogue) computationalism through (the weaker thesis of) generic computationalism to (the even weaker thesis of) digital (or analogue) pancomputationalism. Connectionism, which has traditionally been criticised by classicists for being non-computational, can be plausibly construed as being either analogue or digital computationalism (depending on the type of connectionist networks used). Computational neuroscience invokes the notion of neural computation that may (possibly) be interpreted as a *sui generis* type of computation. The extreme dynamicist argues that the time has come for a post-computational cognitive science. This paper is an attempt to shed some light on this debate by examining various conceptions and misconceptions of (particularly digital) computation.

1 Introduction

There is currently considerable confusion and disarray about just how we should view computationalism, connectionism and dynamicism as explanatory frameworks in cognitive science. In this paper, I endeavour to shed some light on the degree to which they are in

· This is a preprint of the article appearing in *Minds & Machines*. It is reproduced with the permission of Springer-Verlag. The final publication is available at <http://link.springer.com/article/10.1007%2Fs11023-012-9286-y>. This article has been superseded by Chapter 8 of “Physical Computation and Cognitive Science” (http://link.springer.com/chapter/10.1007/978-3-642-41375-9_8).

conflict versus their degree of overlap, on whether they are explanatory or merely descriptive, on which levels of analysis they belong to and on their explanatory posits. Since, by and large, this task is conceptually laden, it is taken primarily from a philosophical point of view. An important distinction that should be drawn in this context is between the conceptual issue of how computation is best characterised and the empirical issue of how cognition is best explained.

On the one hand, as regards the empirical issue, *classicism*, connectionism and dynamicism are not in competition. A single system might be correctly modelled within each one of these paradigms. It is also even possible that all of them include some form of computation simpliciter (insofar as dynamicists are willing to accept it as being explanatorily relevant). So, in this sense, the three paradigms may explanatorily coexist, if within each paradigm the *same* system is modelled in *different* ways. Still, this implies that within dynamicism *models* of the cognitive phenomena in question are also made available.

On the other hand, viewed from a different perspective, they are in conflict. Either the bulk of cognitive phenomena are best explained symbolically or they are not. And if they are best explained symbolically, then a particular form of (digital) computation will indeed be central. This was the crux of the classicist-connectionist debate in the late 80's and throughout the 90's. Further, either the bulk of cognitive phenomena are best explained in a disembodied/non-embedded manner or they are not. Here enters the extreme dynamicist to the debate denouncing the computationalist and connectionist explanatory efforts.

Moreover, the concept of computation is ill-understood and it is the source of an ongoing conflict among the central paradigms in cognitive science. This conflict stems from an equivocation on the notion of computation simpliciter. *Computation* is invoked differently by

broad digital computationalism, connectionism and computational neuroscience with varying degrees of success. It is not just the dichotomy between analogue and digital computation that is the basis for this equivocation, but also the diversity of extant interpretations of *digital* computation. Analogue computation has received even less attention in the literature, and much like its digital counterpart, it remains equivocal. Some observations are also made in the paper regarding how the precise characterisation of ‘analogue computation’ varies among different authors. Still, my focus here is on concrete digital computation (i.e., as it is actualised in *physical* computing systems).

Two main arguments are presented throughout this paper. First, a blanket dismissal of the key role computation plays in cognitive science is unwarranted. For ‘computation’ is an ambiguous concept and is invoked differently across a range of research programs in cognitive science. And whilst some accounts of concrete *digital* computation proper are untenable, others remain plausible and have important implications for the explanatory paradigms that are underpinned by them.

Second, the idea that computationalism, connectionism and dynamicism are mutually exclusive is wrong. For computationalism can be narrowly construed as classicism, but also more broadly as digital computationalism, generic computationalism or even pancomputationalism (and also as analogue computationalism). Further, connectionism is compatible with generic computationalism, since it may be classified as either digital or analogue computationalism, depending on the type of neural nets used. Digital computationalism and connectionism make available mechanistic models of the cognitive phenomena in question. But dynamicism proper is not on a par with either connectionism or digital computationalism, as it does not (necessarily) offer a mechanistic explanation.

The paper proceeds as follows in reply to the question of which notion of computation simpliciter is essential for explaining cognition. In Section 2, I make some observations concerning representations in cognitive science and the type of representations that plays a role in computing systems proper. Subsequently, in Section 3, the classicist stance is reviewed and then compared with broader versions of computationalism: digital computationalism, generic computationalism and pancomputationalism based on the chosen construal of computation. In Section 4, I show how (pace classicists) connectionism is an important variant of computationalism. Section 5 examines the role that computation plays in computational neuroscience. In Section 6, I explore the non-computational shift promoted by extreme dynamicists, who dismiss the key role computation should play in cognitive science. Section 7 addresses the mechanistic versus non-mechanistic debate and how computationalism, connectionism and dynamicism figure in that debate.

2 Representations in Cognitive Science

Since the following discussion revolves around representations as they figure in the philosophy of mind, but supposedly also in computation proper, a brief digression is required to briefly examine them. When the mind is viewed as being involved in coordinating the behavior of a cognitive agent in its environment, one plausible strategy is to view some of its internal states and processes as carrying information about or *standing in* for those relevant aspects of its body and external states of affairs in negotiating its environment (Bechtel 1998a: p. 297). Mental representations function as Stand-Ins for objects or events outside the cognitive agent and once the agent obtains those representations, it can operate on them rather than needing the actual objects or events (Fodor 1980).

Mental representations have two important features, namely being *physically realisable*

and being *intentional*. The first feature implies that they have causal powers. And being intentional, mental representations convey meaning or content. This characterisation presupposes a distinction between the *vehicle* (a physical state or structure, such as a string of symbols) and its *content*. The issue concerning the admissible *vehicles* of representation remains highly controversial (Egan 2011), yet it is common in computational cognitive science to assume that these vehicles are computational structures or states in the brain (von Eckardt 1993: pp. 168-169).

Moreover, there are two main approaches in computational cognitive science to the interpretation of representational vehicles. According to classicism, complex data structures (formally construed) constitute the representational vehicles of our mental representations. According to connectionism, the representational vehicles are either local, in which case they are attributed to individual activated units, or distributed, in which case they are attributed to sets of activated units (von Eckardt 1993: pp. 169, 176). Whilst the main motivation for taking the classicist's data structures to be the bearers of representational content is their compositionality-enabling structure, connectionist networks do not straightforwardly exhibit such structure.

Also, an important distinction to be drawn regarding mental representations in this context is between processes operating *on* representations and representations *figuring in* processes (Bechtel 1998a: pp. 299-300). The 'operating on representations' alternative gives rise to the interpretation of representations as static data structures awaiting some operation to be performed on them. On the other hand, the 'representation figuring in processes' alternative allows for representations to change dynamically. The former alternative is the basis for the classicist thesis, where representations, which have a propositional format, are

operated on by explicit language-like rules. Still, arguably in connectionist networks distributed representations figure in activation-spreading processes and change dynamically.

However, in the context of accounts of (digital) computation proper, a further distinction should be drawn between *intrinsic* and *extrinsic representations*. We should distinguish what computer scientists call *formal* semantics from *real-world* semantics invoked by philosophers (White 2011: p. 194). An *intrinsic* representation in a digital computing system is “confined” to the physical boundaries of that system (and has some formal semantics), whilst an *extrinsic* one (which has real-world semantics) is not. Internal symbols, for example, are intrinsic representations, whose referents are also internal to the computing system. So, both the representer (e.g., a symbol or a string) and the representee (e.g., a memory register or an instruction) reside within the physical boundaries of the computing system. Internal reference to symbols and expressions in conventional digital computers is assigned and it is a primitive in the computer’s architecture. Further, symbols in programming languages have formal semantics that is given by the semantics of those languages (ibid: p. 191).

Any *semantics* of intrinsic representations is confined to the physical boundaries of the computer¹. This semantics interprets the primitives of the computer language as actions on the (somewhat abstracted) internal state of the computing system (ibid: p. 194). An example of an intrinsic representation is the primitive ADD operation in digital computers. It is

¹ This claim raises some ontological quandaries about semantics being confined to some physical boundaries. To avoid a metaphysical debate, let me clarify. In conventional digital computers, computer programs are translated into machine language, which drives the operation of the computer at the hardware level. Take the following code example in assembly (a low-level programming language that works very close to the hardware level).

```
__asm__ ( “movl $2, %eax;”  
         “movl $25, %ebx;”  
         “imull %ebx, %eax;”)
```

This instruction tells the computer to multiply 2 and 25 and store the result into register %eax. The end result might represent, say, a total of 50 apples for a field trip of 25 children. But that makes no difference to the execution of the instruction above. The semantics of that instruction (i.e., moving data between registers, multiplying values, etc.) is contained within the boundaries of the computer.

described as a numerical opcode in the machine language, which is interpreted by the system as *standing in for* or *representing* the ADD operation itself. This may invite the challenge that an interpretation by the computing system implies that it has knowledge of that instruction. But this is hardly the case in human-engineered computing systems. A reply to this challenge requires a further distinction between *know-how* and *know-that*. Crudely put, the former is implicit knowledge, which is typically based on heuristics, whereas the latter is explicit and consists of propositional knowledge. Some have argued that know-how, such as how to ride a bike, how to play a piano, etc., “cannot be analyzed in terms of abilities, dispositions and so on; rather, there appears to be an *irreducible cognitive element*” (Chomsky 1992: p. 104, *italics added*). Or in other words, know-how requires know-that.

Yet, others have argued that not all know-how consists of propositional knowledge. “[I]f, for any operation to be intelligently executed, a prior theoretical operation [of considering appropriate propositions] had first to be performed [...], it would be a logical impossibility for anyone ever to break into the circle” (Ryle 1949: p. 31). Instead, know-how is construed as a skilled performance of an operation that is measured by its success, efficiency, etc. (*ibid*: p. 29). Still, some researchers have insisted that know-how always consists of propositional knowledge (Stanley and Williamson 2001).

The plot thickens, but we need not go that far. To execute the primitive ADD operation, the CPU follows the opcode direction to the physical address of ADD. And the ADD operation itself is simply a hardwired mechanism² that converts input bits to output bits using some combination of logic gates. Put another way, the ADD operation is coded by a unique

² If, for some technical reasons, this mechanism is replaced with a soft-wired mechanism (i.e., either through explicit how-to rules or a soft-constraint learning mechanism), the overall principle will still hold. Even in the case of the soft-constraint learning mechanism, it will eventually learn (say, by heuristics) *how* to perform effectively without knowing *what* it is doing.

binary pattern and whenever this particular sequence lands in the CPU's instruction register, it is akin to a dialled telephone number that mechanically opens up the lines to the right special-purpose circuit (Dennett 1991: p. 214). The CPU's "know-how" requires no "know-that" of the ADD operation.

On the other hand, *extrinsic* representations refer to symbols, data or objects that exist *outside* the physical boundaries of the computing system. Unlike their intrinsic counterparts, extrinsic representations are external-knower-dependent: a knower assigns external (or real-world) semantics to data structures, strings or symbols. The contents of some data structures or computer programs may have external semantics relating to some states of affairs when the computing system directly interacts with the environment in which it is embedded. Still, the computer program will perform, say, the same database search operation (if prompted) just as well, even if the strings of symbols searched for were the names of planets (rather than, say, names of employees) and the corresponding numerals were their coordinates in the galaxy (rather than, say, salaries of employees)³.

3 Computationalism

3.1 A narrow construal – classicism, and symbolic computation

The classicist thesis is that cognition is symbolic computation. Zenon Pylyshyn claims that the idea that "certain behavioural regularities can be attributed to different representations (some of which are called beliefs [...]) and to symbol-manipulating processes operating over these representations" is fundamental to cognitive science (1999: p. 10). Similarly, John

³ At the program level, any factual information entered by a user is converted into something recognisable by the computing system by using an implicit *semantics dictionary*. This dictionary is used to translate any factual information into some data structure that is recognisable by the program. The ace of hearts card, for instance, is translated into a data structure with properties such as a shape, a number, etc. This data structure can be processed by the program and when appropriate, the processed data can be translated back into some form of human readable information as output.

Haugeland claimed that “thinking and computing are radically the same” (1985: p. 2) and that “an intelligent system *must contain* some computational subsystems [...] to carry out [...] internal manipulations” (1985: p. 113, italics added).

Let us pause briefly to consider two similar accounts of concrete computation that underpin the classicist view, according to which digital computation is interpreted as *program-controlled symbol manipulation*. The first one is the *formal symbol manipulation* (FSM, for short) account, which is nicely summarised by Jerry Fodor. “[Digital] computation is a causal chain of computer states and the links in the chain are operations on semantically interpreted formulas in a machine code” (Fodor 1981: p. 122). Fodor, Pylyshyn and Haugeland subscribe to the FSM account (though they diverge on some of the particulars). According to this account, a physical system performs digital computation when it processes semantically interpreted (not just interpretable) symbols (Pylyshyn 1984: pp. 62, 72). Digital computing systems manipulate symbol tokens, which are representations of the subject matter the computation is about, in accordance with some purely formal principles.

The second relevant account of concrete computation in this context is the physical symbol systems (PSS, for short) account. Its main champions were Allen Newell and Herbert Simon. According to this account, digital computing systems just are (universal) physical symbol systems containing sets of interpretable and combinable entities (i.e., symbols) and a set of processes that operate on these entities by generating, copying, modifying, combining and destroying them according to instructions. These symbols are physical patterns (i.e., tokens) that can occur as components of symbol structures (Newell and Simon 1976: p. 116). The resemblance to the FSM account is clear. Newell and Simon argued that “[a] physical symbol system has the necessary and sufficient means for general intelligent action” (ibid).

Undoubtedly, classicists are committed to the idea that cognitive capacities are underpinned by mental representations. They insist that the combinatorial structure and compositionality of mental representations is critical for our cognitive capacities (Fodor and Pylyshyn 1988: pp. 17-18). Fodor claims that mental processes are operations defined on syntactically structured mental representations, much like sentences in natural language (1981). Pylyshyn adds that “[w]hat makes it possible for [...] intelligent organisms to behave in a way that is correctly characterised in terms of what they represent (say, beliefs and goals) is that representations are *encoded* in a system of physically instantiated symbolic codes” (1999: p. 5, italics original).

In short, classicism is a narrow conception of digital computationalism. It is committed to a symbolic model of cognition consisting of at least two levels. Physical symbol systems are describable at two levels: the symbol level and the physical level. Newell asserted that symbol structures and operators⁴ on these structures (at the symbol or program level) are realisable in physical mechanisms (1980: p. 156). Pylyshyn proposes a tripartite decomposition of cognitive systems (akin to David Marr’s tripartite analysis⁵). At the top/semantic level, knowledge and goals as well as certain behaviours of the cognitive system are attributed to different representations and the processes operating on them respectively. At the middle/symbol level, symbolic expressions encode the semantic content of the system’s knowledge and goals. At the bottom/physical level, representation-governed behaviour of the entire system is implemented by some biological substrate (Pylyshyn 1993, 1999: pp. 7-11).

⁴ Operators (such as ‘+’, ‘-’, or ‘copy’) are symbols or symbolic expressions that have an external semantics built into them (Newell 1980: p. 159).

⁵ The semantic level, for example, is sometimes equated with Marr’s top/computational level, but it should not be. Marr’s top level characterises the function computed by the cognitive system. This computation may (but need not) involve the assignment of semantic contents.

It is easy to see then why classicists invoke computation-theoretic language to explain cognition. In the light of the above, the great flexibility of program-controlled *digital* computers makes them ideal models of cognitive agents performing complex tasks in virtue of language-processing-like operations. By endorsing either the FSM or PSS accounts of computation, classicists can easily appeal to existing computational architectures and related tools to explain cognitive phenomena. Yet, they also insist on a too narrow class of digital computing systems and impose an extrinsic representational constraint on computation proper⁶.

3.2 *Broad construals of computationalism, and digital computation*

How broadly can we construe computationalism? The short answer is: *it depends*. The classical dichotomy of computation simpliciter is between digital and analogue computation. Even if we took computation to be just *digital* computation, we would still be left with many versions of digital computationalism depending on the particular account of computation that we endorsed. Broad *digital computationalism* is certainly more encompassing than classicism, which posits a narrow class of digital computing systems. A classicist, who subscribes to the FSM account, takes physical computing systems to be program-controlled digital computers. Her fellow classicist, who subscribes to the PSS account, takes physical computing systems to be programmable stored-program digital computers.

Importantly, different accounts of digital computation entail different versions of broad digital computationalism. For example, according to the view endorsed by John Searle (1990) and Hilary Putnam (1988), every sufficiently complex physical system (trivially) performs digital computation. This view inevitably leads to *strong digital pancomputationalism*, for

⁶ This imposed representational constraint is unsurprising, as the motivation of the classicists, who promote the FSM and PSS accounts, was advancing a substantive empirical hypothesis about how human cognition works.

rocks, chairs, paper clips, oranges, humans and the physical universe – all digitally compute. It is not only that every sufficiently complex physical system computes *some* Turing-computable function (i.e., *weak* digital pancomputationalism), but rather that the system computes *every* Turing-computable function. This version of digital pancomputationalism is hardly illuminating. More precisely, it is an anti-realist version of pancomputationalism. It does not tell us that the universe has a particular structure, but it is rather invoked to argue against cognition being computational in any substantial sense (Dodig-Crnkovic and Müller 2011: p. 154). It stems from the anti-realist view that it is merely our subjective description that makes a physical system computational.⁷

However, we need not go so far as to promote digital pancomputationalism, to be able to endorse a digital computationalist thesis that is broader than classicism. Subscribing to some of the other extant accounts of concrete digital computation, which do not appeal to extrinsic representations, leads to a broader digital computationalist thesis. For example, according to the mechanistic account of computation, a physical system performs digital computation, if it manipulates input strings of digits⁸, depending on the digits' type and their location in the string, in accordance with a rule defined over the strings (and possibly the system's internal states) (Piccinini and Scarantino 2011: p. 8).

The resulting digital computationalist thesis, which is based on the mechanistic account, is broader than the classicist thesis. On this account, digital computing systems are individuated by their functional properties that are specified mechanistically without invoking any

⁷ Gordana Dodig-Crnkovic asserts that to make pancomputationalism a substantial thesis that plays a key role in a scientific theory about the universe, we should adopt a realist weak version of pancomputationalism (Dodig-Crnkovic and Müller 2011: pp. 154-155). All processes can be described as computational processes, since such a description happens to be useful in a scientific theory. It is 'weak' in the sense that it focuses on ways of description, rather than on realist ontology.

⁸ A *digit*, on this account, is a stable state of a component that is processed by the computing system. In ordinary electronic computers digits are states of physical components of the machine (e.g., memory cells).

extrinsic representations (Piccinini 2008a). Further, unlike the FSM and PSS accounts of computation, it is not restricted only to symbolic computation. Relevant digital computing systems, on the mechanistic account, range from special-purpose TMs and special-purpose computers through universal TMs, programmable stored-program systems, Gandy machines⁹ and discrete neural networks to (the contentious) hypercomputers. Any one of these systems has its own pros and cons as an adequate model of cognition. Still, the point is that the resulting digital computationalist thesis is more encompassing than the classicist thesis.

There are other accounts of computation that neither presuppose any extrinsic representational vehicles nor restrict the class of digital computation to the class of symbolic computation. Such accounts include the algorithm execution account and Robin Gandy's account (of *parallel* computation). How 'algorithm' is interpreted affects the resulting algorithm execution account¹⁰. For the purposes of this paper, we adopt Jack Copeland's account, according to which a physical system performs digital computation when it acts in accordance with an algorithm (1996). He defines an algorithm, *Al*, as a finite set of instructions such that, for some computing system, *CS*, each instruction of *Al* calls for one or more primitive operations of *CS* to be performed, either unconditionally or if specific conditions, recognisable by *CS*, are met (Copeland 1997: p. 696).

Moreover, a key feature of his account alluding to representations is the "labelling scheme" requirement (ibid: p. 338). The labelling scheme of *CS* consists of two parts, the

⁹ A Gandy machine is a deterministic discrete machine that can perform operations in parallel. It can be conceptualised as multiple TMs working in parallel, sharing the same tape and possibly writing on overlapping regions of it.

¹⁰ It is worth noting that Robert Cummins, for instance, also holds the view that digital computation is the execution of algorithms (or programs), but his view does presuppose extrinsic representations. "[B]eing able to track computations under their semantic interpretations allows us to see how a physical engine – a computer – can satisfy epistemic constraints" (Cummins 1996: p. 66). But his account of computation proper is ultimately inadequate for other reasons as well. On his account, Searle's wall also computes (Copeland 1996: p. 353).

designation of certain parts of *CS* as label-bearers and the method for specifying the label borne by each label-bearing part at any given time. Yet, this designation is limited to *intrinsic* representations of numbers, functions and computing instructions. On this account, digital computation is not limited to symbolic computation, and it includes any system that acts *in accordance* with an algorithm, such as special- and general-purpose digital computers, TMs and finite state automata. Accordingly, the resulting digital computationalist thesis in this case is that cognition is algorithmic computation that need not be symbolic.

Gandy's account also gives rise to a broad digital computationalist thesis. According to this account, a physical system performs digital computation when it goes through a sequence of state transitions whose input is encoded as the system's initial state, and each one of its states is its output at a given time (1980: p. 127). On this account, labels designate "the various parts of the machine – e.g., [...] a transistor and its electrodes [... but also] positions in space (e.g., for squares of the tape of a Turing machine) and [...] physical attributes (e.g., [...] the symbol on a square)" (ibid). Yet, this designation need not involve any extrinsic representation. Further, Gandy's account encompasses parallel digital computation by violating Turing's boundedness condition. The resulting computationalist digital thesis is broader than the classicist thesis, and it is *prima facie* more biologically plausible, given what neuroscience tells us about the parallel neural activity in the brain.

Computationalism may be further extended beyond *digital* computationalism. If 'computation' is taken as *generic computation*, then we get the broadest version of computationalism (that is not digital pancomputationalism). Generic computation includes digital computation, analogue computation, quantum computation and neural computation. It is characterised as the processing of medium-independent vehicles according to rules

allowing for the processing of continuous variables, strings of digits, or neuronal spike trains (Piccinini and Scarantino 2011: p. 10-13). *Generic computationalism* is the thesis that cognition is computation in a generic sense. It does not amount to pancomputationalism though. It is a plausible (but weak) explanatory framework and still falsifiable (e.g., if it turned out that cognitive capacities depended inherently on some particular physical properties).

In sum, computationalism should not be identified with classicism. The latter is but one digital computationalist alternative positing a narrow class of digital computing systems as candidate models of cognition. How broadly computationalism should be construed depends on the particular account of computation invoked. Choosing the right account is no easy task. The discussion thus far still leaves out the analogue computationalism alternative, which is based on analogue computation. This alternative is less common in cognitive science and it is discussed in the next section.

4 Connectionism and Sub-symbolic Computation

The connectionist thesis is that cognition is sub-symbolic computation. Accordingly, cognition should be explained by neural network activity (in a more generic sense than the association between stimuli and responses). Modern connectionists argue that these neural networks¹¹ perform sub-symbolic computation (Rumelhart and McClelland 1986; Smolensky

¹¹ Neural (or connectionist) networks consist of multiple interconnected homogenous units called ‘neurons’. These nets can be classified into two general categories: feedforward nets and feedback (or recurrent) nets. In the former case, units are arranged across multiple layers such that the output of units in one layer depends only on those in previous layers. The outputs of units are updated layer by layer with the first one being the *input layer* and the last one being the *output layer*. In the latter case, feedback loops in the network allow signals between units to travel in both directions (rather than just in a unidirectional forward manner). A source of controversy arises in regard to representations in connectionist nets. On the localist interpretation, each individual unit, which is active in a particular distributed activation pattern, realises an individual representation contributing to the overall content of the activation pattern. On the distributive interpretation, a representation is realised by either by an activation pattern or an activation pattern and its

1988; Smolensky and Legendre 2006; Chalmers 1992; MacLennan 2001). Under this interpretation, it is implicit that neural nets are capable of computations that are not limited to discrete manipulations of symbolic representations. In a sharp contrast to classicism, most connectionists reject the claim that a language of thought is required for an adequate explanation of cognition. This makes the tension between connectionism so construed and classicism obvious. Some have explicitly advanced the thesis that neural nets are analogue computers (Diederich 1990; O'Brien 1999; O'Brien and Opie 2006). Others have restricted the classification of neural nets as analogue computers to a certain kind of networks, primarily those that process real-valued quantities (Siegelmann 1999; Kremer 2007).

Moreover, according to Gerard O'Brien and Jon Opie, connectionism is grounded in analogue computation, for neural nets "compute by exploiting relations of structural resemblance between their connection weights and their target domains" (2006: p. 41). On their view, "[a]nalog computers are systems whose behaviour is driven [...] by semantically 'active' analog representations that physically or structurally resemble what they represent" (ibid: p. 33). It follows, by their lights, that neural networks are analogue computers. The representational vehicle invoked in a connectionist analysis is based on a structural isomorphism between the network's activation patterns and the task domain. This isomorphism renders the shape of the activation landscape *semantically* significant (ibid: pp. 32-34; O'Brien 1999).

If they are right, then connectionism is a variant of *analogue computationalism*, that is, the thesis that cognition is analogue computation. But, as already mentioned above, the notion of analogue computation is also equivocal. According to O'Brien and Opie, analogue

weighted connections. For further discussion on neural networks see, for example, John Tienson's introduction (1988).

computation is defined over analog representations. When Hava Siegelmann invokes this notion, she refers to computation performed by a very specific type of recurrent neural networks, which perform operations on real variables and allow loops among some of the units (1999). Nevertheless, the most precise characterisation of analogue computation may be attributed to the Shannon-Pour-El Thesis. According to this thesis, the outputs of general-purpose analogue computers correspond exactly to differentially algebraic functions. Therefore, there exists a universal analogue computer that using just a handful of integrators can compute (to some arbitrary degree of approximation) any possible continuous function (Rubel 1985: pp. 75-76). The main point is that analogue computation is a continuous change of real variables over time.

Yet, there remain the questions whether analogue computation has to be defined over representations and whether connectionist networks are rightly classified as analogue computers. Analogue computers (and their processing units) have the function of transforming an input real variable into an output real variable, which stands in some specific functional relation to the input variable. Whilst their operations can also be understood in terms of analogue representations, they need not be (Piccinini 2008b: p. 48). I have argued elsewhere that connectionist computation is best classified as analogue computation without invoking any extrinsic representational properties (REMOVED_FOR_ANONYMITY 2010). However, this conclusion is too strong. There are certainly good reasons to classify *discrete* neural nets, which process binary-valued or integer-valued quantities, as digital computing systems (still without invoking any extrinsic representations). That would certainly be the case, if we adopted, say, the mechanistic account of concrete digital computation.

Some connectionist networks perform digital computation, while others perform analogue

computation. The idea of discrete binary networks goes back to the seminal 1943 paper by Warren McCulloch and Walter Pitts. On their model, each neuron was modelled as a linear threshold element with a binary output. This was the first model of a discrete neural net exhibiting all-or-none firing patterns. When both the inputs and the outputs of such neural nets are binary the result is a Boolean circuit (Siu et al. 1995: pp. 1-2). Since the McCulloch and Pitts networks can be used to build digital computers, these (and similar discrete) neural networks are best classified as digital computing systems¹².

Incidentally, Daniel Dennett has pointed out that connectionist networks should not be regarded as a “shift to some ‘qualitatively different’ mode of operation. [For] at the heart of [the connectionist] system lies a von Neumann engine [...] computing a computable function” (1991: p. 269). I do not know whether Dennett meant it literally (or just metaphorically). But if we analyse individual units of a discrete neural network as simple physical computing systems, strictly they need not have von Neumann architecture.

For von Neumann architecture implies a general-purpose computing system, whereas individual units of the neural network are special-purposed. John von Neumann and colleagues argued that for the “device [to] be a general-purpose computing machine it should contain certain main organs relating to arithmetic, memory-storage, [and] control” (Burks et al 1946: p. 399). The arithmetic logic unit in the von Neumann architecture must be capable of the basic elementary operations of addition, subtraction, multiplication and division. But each individual neural unit only performs addition and multiplication of all the weighted connections leading to that particular unit. Besides, these units need not have any built in memory for storing multiple instructions and other data. But perhaps most importantly, a

¹² Otherwise, if McCulloch and Pitts networks were classified as analogue computing systems, then digital computers would be analogue too.

general-purpose computing system can be programmed to perform any function that some special-purpose computing system can perform. Yet, each neural unit is a special-purpose system, whose instruction is an integral part of that system and constitutes a part of its design structure. Each unit can be described as either an IF-THEN equivalent (if threshold exceeded, then “fire”) or a Boolean circuit (when its inputs and output are binary).

Furthermore, we can distinguish between two types of neural nets based on their dynamics. According to the mechanistic account of computation, the first type of networks takes strings of digits as inputs and outputs, has discrete dynamics and does not change its structure over time. The second type of networks takes strings of digits as inputs and outputs, but has continuous dynamics or changes over time (Piccinini 2008c). Whilst only the first type belongs to the class of classical digital computing systems, both these types of neural nets perform digital computation on the mechanistic account of computation.

Yet, there exists another class of neural nets, which process continuous real-valued quantities, that *do not* perform digital computation. These networks turn their input into their output in virtue of their continuous dynamics and do not compute by manipulating strings of digits (ibid: p. 319). Continuous variables are not strings of digits and this suffices to rule out these networks as digital computing systems in the sense of computation employed in computer science. Nevertheless, these neural nets can be correctly classified as analogue computers, for they satisfy the following five plausible criteria. First, the network’s operations take a continuous range of values over time. Second, its physical dynamics are governed by operations on real variables. Third, the functional relation between inputs and outputs of the net is best described by a set of differential equations. Fourth, the network’s inputs and outputs are distinguished from one another up to a limited degree of precision.

Lastly, the net may be subject to varying levels of noise (REMOVED_FOR_ANONYMITY 2010).

Moreover, on other accounts of concrete digital computation, (discrete) neural nets do not straightforwardly qualify as performing digital computation. Unsurprisingly, on both the FSM and PSS accounts, connectionist networks do not compute, because they do not operate algorithmically on structured symbolic representations. This has been the source of much debate in the late 80's and throughout the 90's (for just the tip of the iceberg see, for example, Fodor and Pylyshyn 1988; Smolensky 1991, 1995; Clark 1990; Chalmers 1993; Matthews 1997; Bechtel 2001). But interestingly, even on the algorithmic execution account above, it is not immediately clear that discrete neural networks compute, for it is not obvious whether they execute algorithms in the classical sense of computability theory. This may seem bizarre at first, since neural nets are typically simulated on digital computers. But that is not the point. The question is whether discrete neural nets *perform* digital computation not whether they can be simulated on digital computers.

To answer this question one needs to judge (discrete) neural nets on their own merits as stand-alone non-simulated systems. When a neural net is implemented as a physical collection of interconnected simple processors (each one being an individual unit), it still needs to be trained to perform its designated task. The most common method of doing that is using the backpropagation learning procedure. Whilst this procedure can be described as an algorithm in the classical sense used in computer science, it still does not imply that connectionist computation is algorithmic.

Once the system is trained and performs its task successfully, the question remains: does the network operate algorithmically? Connectionist networks do not operate by following the same type of “hard” predefined rules that are programmed on conventional

digital computing systems¹³. Their operation can rather be described as the satisfaction of soft-constraints, where each connection between two units represents a soft-constraint. Whether a unit actually fires or not depends on a simple summation function of all the weighted signals received by any particular unit. This activation is commonly known as a *spreading activation algorithm*, where it is distributed over the network, based on some mathematical function of the connections weights (Waltz and Pollack 1985: pp. 54-55).

Granted that connectionist networks compute, connectionism is (at least) a subclass of generic computationalism. If *discrete* connectionist networks are *sufficient* for explaining cognition, then connectionism (does not just overlap with, but) is a subclass of digital computationalism¹⁴. If *continuous* neural nets are *sufficient* for explaining cognition, then connectionism is a subclass of analogue computationalism. However, if the full range of connectionist networks is required for explaining cognitive phenomena, then connectionism is a subclass of generic computationalism. At any rate, on the preceding analysis, none of these three options has to presuppose extrinsic representations for connectionist network computation.

5 Computational Neuroscience, and Neural Computation

Already in the 90's, but particularly in the past decade, computational modeling of cognition has become an active area in neuroscience in an attempt to disclose how neurons give rise to cognitive functions. This research program now wears the title computational neuroscience

¹³ By implementing soft constraints, connectionist networks arguably allow the task demands, rather than the designer's biases (like in rule-driven digital computing systems) to be primary driver shaping the operation of the network. To some extent, this approach reflects a shift in methodology when compared with Marr's classical top-down approach (which is overtly endorsed by classicists).

¹⁴ Still, this does not completely resolve the classicist main beef with connectionist networks, which do not process structured symbolic representations. Fodor and Pylyshyn think that cognition is syntactically governed manipulation of structured representations. Connectionism, so they conclude, is hopeless as a (competence) theory of cognition (REMOVED_FOR_ANONYMITY 2010).

and employs a broad range of techniques also using some tools from the domain of computer science. It is worth noting that computational neuroscience should not be identified with connectionism. The latter typically refers to models based on behavioural data, whereas the former refers to models based on *both* behavioural and neuroscientific data. Besides, the backpropagation method, which is typically used to train the system, depends on the units being able to relay signals bi-directionally. However, the dendrites and axons, which act as input and output channels to and from brain neurons typically allow nerve impulses to travel in one direction only. And whilst individual units in connectionist networks are homogenous, brain neurons are physiologically specialised.

Furthermore, computational neuroscience downplays the explanatory role of the standard digital computer metaphor and connectionist networks in cognitive science. Computational cognitive science attempts a fairly close integration of psychological, neurophysiological and neurobiological data and theories of cognition (Boden 2008). Most existing connectionist networks are hugely different from the anatomy of the brain. The units of connectionist networks are computationally far too simple when compared with real neurons¹⁵, though some attempts have been made to model brain neurons more faithfully (cf. the discussion about models that do not impose the simplification or homogenisation of the computational units in Maass and Markram 2004).

Moreover, Patricia Churchland and Terrence Sejnowsky argue that Marr's tripartite computationalist analysis aligns poorly with the levels of organisation in the nervous system (1992: pp. 18-19). On Marr's analysis, the top-level competence function can be examined independently of understanding the algorithm that is performed in the brain and similarly the

¹⁵ Of course, some degree of simplification is needed to make *any* model viable, since models, by definition, abstract away from some of the particulars of the modelled system. The question here is whether connectionist networks simplify *too much* in the process of modelling cognition.

problem of discovering the algorithm at work is independent of its underlying physical realisation. This top down approach makes neurobiological facts about the nervous system less relevant, since they are just details at the implementation level. Later research in computational neuroscience suggested that knowledge of the brain architecture plays a vital role in understanding those “algorithms that have a reasonable shot at explaining how in fact neurons do the job” (ibid: p .19).

Unlike digital computationalism, computational neuroscience studies cognition in a bottom-up approach, whilst still being informed by top-down theories. Research from neuropsychology, neuroethology and psychophysics provides the details about the relevant lower level mechanisms. But lower level research remains incomplete in the absence of top-level analyses of the very cognitive capacity, whose mechanisms are studied at the lower level. Neuroscientific research can profit, for instance, from abstract discoveries in computability theories and discoveries in the construction of physical computing systems (ibid: pp. 11-12). Unlike other cognitive scientific research programs, neuroscience attempts to do more than “merely reproduc[e ...] a function of the brain (such as playing chess)” whereas this may be sufficient in AI research (Eliasmith and Anderson 2003: p. 1). Yet, as the name suggests, computational neuroscience is committed to the view that the brain is an implemented computing system (Churchland et al. 1988; Churchland and Sejnowsky 1992; Dayan and Abbott 2001; Eliasmith and Anderson 2003; Trappenberg 2010).

Nevertheless, computational neuroscience is not committed to cognition being either symbolic computation or sub-symbolic computation, for that matter. Neurons are taken to be computational units that process information to solve complex tasks, such as perception. But what neuroscientists take ‘computation’ or ‘neural computation’ to be is another matter

entirely. One approach is to agree that whilst there is no precise definition of the computation performed in the brain, it certainly is broader than the notion of digital computation. They take a physical computing system to be one whose physical states can be described as representing states of some (other) systems, where transitions between states are operations on representations (Churchland and Sejnowsky 1992: p. 61-62; Eliasmith 2003). More specifically, neural computation amounts to the encoding and decoding of neural spike trains (Eliasmith 2007: pp. 326-327).

Arguably, neural computation so characterised may be a *sui generis* type of computation. This has been a recent thesis of some researchers, who argue that neuroscientific evidence shows that, on the one hand, typical neural signals (e.g., spike rates) are continuous, yet, on the other, these signals are constituted by spikes, which are discrete elements (Piccinini and Bahar 2011). Whilst this thesis is not uncontentious, it is compatible with some other characterisations of neural computation in neuroscience according to which neural computation is neither digital computation nor analogue computation (Churchland et al 1988: pp. 47-50; Eliasmith 2007: pp. 326-327; Poggio and Koch 1985). Others have proposed *natural computation* as an alternative notion of computation that is more suitable for describing the behaviour of biological systems (MacLennan 2004; Hamann and Wörn 2007)¹⁶. The claim that neural computation (as it is invoked in computational neuroscience) is a *sui generis* type needs unpacking and I lack the space to discuss it further here.

6 Extreme Dynamicism and the Non-Computational Shift

Various “anti-representationalist” approaches are included under this heading starting with “radical” dynamicism (e.g., Thelen and Smith 1994; van Gelder and Port 1995) through

¹⁶ For one thing, neural activity has many sources of noise making the underlying computation imprecise sometimes. This suggests that, unlike digital computation, natural computation itself is noisy and imprecise (MacLennan 2004: p. 129).

embodied and embedded dynamicism (e.g., Pfeiffer and Scheier 1999) to the enactivist approach (e.g., Varela et al. 1991; Thomson 2007). Whilst there are important differences amongst these approaches and grouping them together certainly does them an injustice by blurring those differences, they all share a similar trait. They all reject representation and computation as being key to understanding cognition.¹⁷ Instead, according to this new “post-cognitivist” paradigm, cognition is not computational (Wallace et al. 2007: p. 26). This new paradigm distances itself from both computationalism and connectionism by broadening its research focus on the brain and including the body and its relationship to the “outside” world. The purpose of my exposition here is to reveal any misconceptions about computation and so specific details about the different approaches are omitted for brevity¹⁸.

An underlying claim of the extreme dynamicist approaches is that cognition is not computational. Advocates of these approaches think that it is time for cognitive science to embrace a non-computational paradigm. In the early 90’s, Rodney Brooks designed the mobots, which were robots capable of functioning in a messy and unpredictable environment. He claimed that these robots “do not have traditional AI representations [...] which have any semantics that can be attached to them” (Brooks 1991: p. 149)¹⁹. “Radical” dynamicists, Tim van Gelder and Robert Port argued that “[t]he cognitive system is not a discrete sequential manipulator of static representational structures” (1995: p. 3). Similarly, embodied dynamicists, Rolf Pfeiffer and Christian Scheier, criticised the “analogy between human

¹⁷ The label ‘extreme dynamicism’ is used to alert the reader that in some sense, any cognitive scientist is by definition a dynamicist. For there seems to be a consensus that cognition is a dynamical phenomenon, and as such it requires some application of dynamical systems theory. So, for clarity, the label ‘extreme dynamicism’ is chosen to denote the anti-computationalist position.

¹⁸ To be sure, these different approaches are logically autonomous. One can subscribe to any particular approach without necessarily subscribing to the others. For a nice discussion on the history and differences amongst those approaches see, for example, Evan Thomson (2007: pp. 3-15).

¹⁹ More precisely, Brooks only rejects what I dubbed extrinsic representations for the computations performed by these mobots. “[T]here need be no explicit representation of either the world or the intentions of the system” (Brooks 1991: p. 149).

thinking and processes running in a computer, that is, information processing as the manipulation of symbols” (1999: p. 47).

Researchers endorsing one (or more) of these approaches have rejected the cognitivist paradigm, which gives rise to some form of a “Cartesian theater” (Spivey 2007: p. 313) and relies on a metaphor of the “mind as a computer” (ibid: p. 29). Still, the common interpretation of a computer is as a serial digital system (ibid; Wallace et al. 2007: p. 10; Froese 2011: p. 118) that performs information processing on representations (Wallace et al. 2007: p. 10; Thomson 2007: p. 186). For the extreme dynamicist, representation is not a mandatory concept for explaining cognitive phenomena, which are seen as the simultaneous, mutually influencing unfolding of complex temporal structures. The digital computationalist, on the other hand, supposedly explains cognitive phenomena as simple transformations of static representations (Thelen and Smith 1994: pp. 164-165; van Gelder 1998: pp. 621-622).

It seems then that advocates of the various extreme dynamical approaches share a common (mis)conception of computation. This conception leads them to reject computational research programs in cognitive science. Rather than relying on computer science as the foundation for traditional cognitive science, they promote dynamical systems theory as the foundation for an *alternative* cognitive science. For dynamical systems theory provides a general mathematical theory (which supposedly is already the standard language of the natural sciences) and it allows us to do better justice than computability theory to the continuous temporal changes of cognitive phenomena at multiple timescales (Froese 2011).

Moreover, extreme dynamicists take computation to be a serially digital process that is carried out over extrinsic representations. Yet, Gandy machines, cellular automata and (discrete) connectionist networks, perform parallel digital computation and violate this

narrow characterisation. As well, computational neuroscience invokes the notion of neural computation that is (possibly) different than digital computation proper. Further, the assertion that digital computation is carried out over extrinsic representation is unsupported. Symbolic computation is merely a narrow class of digital computation. So, the dynamicist rejection of “information-processing on representations” as the basis of an adequate model of cognition only applies to models that are based on, say, the FSM and PSS accounts. However, broad digital computationalism is not susceptible to a similar criticism.

Extreme dynamicism is advanced as a non-computational more biologically plausible framework. Nevertheless, it is not obvious why this is the case, as extreme dynamicists tend to ignore the practical details of the underlying mechanisms of the cognitive systems in question. That brings us to the next section.

7 Mechanistic vs. Non-Mechanistic Explanatory Frameworks

Before turning to evaluate whether dynamicism, connectionism and digital computationalism should be viewed as competing or complementary frameworks, let us briefly examine the main aspects of mechanistic explanations. Mechanisms typically have four characteristics: phenomenal, componential, causal and organisational. First, they are phenomenal in the sense that they perform tasks.²⁰ The phenomenon is explained by appealing to the tasks performed as a whole and it partially determines the boundaries of the mechanism. Second, all mechanisms have at least two components. The components of a mechanism are those that are relevant to the explanandum. Third, these components are causally interrelated, that is, they interact with one another. Four, the spatial organisation of the components (in terms of their locations, shapes, orientations, etc.) as well as their temporal organisation (in terms of

²⁰ I follow Carl Craver and William Bechtel in labelling this characteristic ‘phenomenal’ in a manner unrelated to phenomenology (2006: p. 469).

the order, rates, and durations of the activities in the mechanism) play a key role in generating the phenomenon (Craver and Bechtel 2006: pp. 469-470).

Moreover, a mechanistic explanation requires isolating some aspect of the phenomenon to be explained and positing a mechanism that is capable of producing this phenomenon. A mechanistic explanation of a system is achieved by virtue of identifying the relevant subcomponents of the mechanism and the corresponding activities (i.e., localisation) that are organised in the right way so as to produce the phenomenon in question. The localisation of the relevant components and corresponding activities is accomplished by means of *structural* and *functional* decomposition respectively. A structural decomposition begins by breaking the mechanism apart into subcomponents and then investigating what they do. A functional decomposition is accomplished by analysing the phenomenon into activities that, when properly organised, exhibit the phenomenon. For example, the chemical process of fermentation may be decomposed into a set of more basic chemical reactions, including oxidation and phosphorylation (ibid: p. 473).

In the context of mechanistic explanations a distinction is typically made between mechanistic *sketches* (or mechanistic *schemata*) and *complete* mechanistic models. A mechanistic sketch (or scheme) is a functional analysis in which some structural details of a mechanistic explanation are excluded. But once the omitted details are filled in, the functional analysis becomes a full-blown mechanistic explanation. A *complete* mechanistic model identifies the functional properties of the components and must respect constraints imposed by those components. It also does not leave any crucial gaps regarding how the mechanism works (Piccinini and Craver 2011). With this brief exposition in mind, let us return to examine the relation among dynamicism, connectionism and digital

computationalism as explanatory frameworks.

Recently, some devoted dynamicists have argued that (good) dynamical accounts of cognitive phenomena are genuinely explanatory and not merely descriptive (Stepp et al. 2011). This defence was invoked in response to the contemporary mechanistic philosophy of science that allegedly excludes dynamicists' explanations of cognition (Machamer 2004; Bechtel 2009; Piccinini and Craver 2011). According to the defenders, the reason for this exclusion results from either a theoretical commitment to computational explanations or a normative commitment to a mechanistic philosophy of science. Instead of proposing a complete mechanistic explanation, dynamical explanations seek to model cognitive phenomena by identifying higher-level laws (or law-like principles). Dynamical explanations capture the temporal change of the phenomenon in question by a set of differential equations (Stepp et al. 2011: p. 432).

Other authors have argued that some dynamical explanations are mechanistic. Arguably the fact that dynamical explanations use mathematical tools and concepts of dynamical systems theory does not entail that these explanations are non-mechanistic. On this view, (extreme) dynamicism can also sometimes be used to describe cognitive mechanisms. Carlos Zednik offers two examples that supposedly show that dynamical models and dynamical analyses are in themselves mechanistic throughout (2011). The first one is the infant perseverative reaching model by Esther Thelen and colleagues based on Jean Piaget's classic A-not-B task. What Zednik identifies as most significant for his claim is a tripartite analysis of an input vector, which partakes in this dynamical explanation, into a task input, a trial-specific input and a memory trace, which captures the influence of prior trials. The individual contributions of the task input, trial-specific input and memory trace can be supposedly

construed as the posited component operations of a mechanism for goal-directed reaching. Expressed as variables, these operations are linked in a dynamical equation that captures their role in this mechanism (ibid: pp. 248-249).

The second example is Randall Beer's dynamical explanation of perceptual categorisation in a simulated brain-body-environment system. The simulated system consists of a single minimally cognitive agent, which is equipped with a 14-neuron continuous-time recurrent connectionist network "brain". The system is situated in a simple two-dimensional environment, which features a single circular or diamond-shaped object. This object falls vertically toward the agent in the course of the trial, and the agent responds by moving horizontally to catch circles and avoid diamonds, thereby performing a categorical discrimination. By Zednik's lights, Beer's dynamical explanation features a dynamical analysis that describes the activity of two components, the embodied "brain" and the environment (ibid: pp. 250-252).

Nevertheless, these two examples do not show that extreme dynamicism offers a mechanistic explanation. They rather show that it is compatible with mechanistic cognitive models. Zednik argues that Beer's dynamical analysis relies on the mechanistic heuristic of structural decomposition to identify two components, the embodied brain and the environment. The operations associated with each of the components are described by a detailed dynamical analysis. By doing so, Zednik puts a foot on a slippery slope. For once we allow such simple structural decomposition, any dynamicist explanation, which describes the interaction between a cognitive agent and the environment, is supposedly mechanistic. Beer's model is mechanistic, but only because it includes a connectionist network, which models a part of the brain. His dynamical analysis complements the connectionist network model.

The infant perservative reaching model also does not support the claim that some dynamical models are mechanistic proper. Zednik implies that this model can be considered a relatively abstract mechanistic sketch, which leaves more than enough room for elaborating the possible neuroanatomical components giving rise to the goal-directed reaching phenomenon. At best, this model offers a functional decomposition of low-level processes of perception and action (ibid: pp. 249-250). Even if it were classified as an “abstract mechanistic sketch”, it would be at the “very incomplete” end of the spectrum. For, it lacks any structural decomposition of the underlying relevant components. Absent the identification of the participating components, any possible causal relations among them cannot be specified. As an incomplete mechanistic sketch, this model indeed invites a future development of a mechanistic explanation. Yet, there remains a big gap to be filled, as the model has to identify the causal structure of the system in question (Piccinini and Craver 2011: p. 292).

A dynamical explanation should not be misconceived as an *alternative* to mechanistic explanations. The Hodgkin-Huxley model of spike generation is arguably a good example of a genuine explanation. Still, it simultaneously offers a dynamical description (comprising a set of coupled differential equations to describe the dynamics of the membrane action potential) and a mechanistic one (describing how ion channels and related activities are organised to generate action potentials). These differential equations helped guide the search for the underlying components of the responsible mechanism (Kaplan and Bechtel 2011: p. 439).

It is the non-mechanistic dynamical approach that offers a genuinely different *kind* of cognitive science. Such an approach is not on a par with either connectionism or digital

computationalism. But whether this approach is truly explanatory or merely descriptive remains contentious. The burden of proof is on the extreme dynamicist to show how her approach is explanatory in the absence of a mechanistic description. Predictions based on law-like regularities are at best incomplete explanations.

Whilst digital computationalism and connectionism also make available physical *models* of cognitive architectures, dynamicism proper offers a mathematical *formalism* describing the evolution of complex physical systems over time. Classicism and connectionism, for instance, may be competing for the same prize. But not dynamicism, as it provides a completely different type of epistemological analysis with a different purpose than the modelling one served by the other two. Whether cognition turns out to be a programmable digital computing system, a continuous recurrent network, both or neither, has no critical implications for dynamicism. If we endorse the view defended by Zednik, then some dynamical analyses may be considered incomplete mechanistic sketches. Still, as incomplete sketches they have to be elaborated by means of structural decomposition. Typically, dynamical analyses are complemented by connectionist networks in an attempt to identify the relevant subcomponents generating the cognitive phenomenon.

Nevertheless, dynamicism proper is a non-mechanistic explanatory framework. It explains cognitive phenomena in one of three ways: metaphorically, using a small number of variables or using connectionist models (Thagard 2005: pp. 200-203). When not all influencing variables can be identified and the equations cannot be spelled out, dynamicists describe cognition metaphorically. But a metaphor only goes so far as an explanation. In other cases, where a small number of variables can be identified, dynamicism provides a mathematical description of the overall system state and its predicted changes under certain conditions.

When connectionist networks are employed, dynamicism offers a mathematical framework for analysing the workings of these networks (revealing the overlap between dynamicism and connectionism).

“Radical” dynamicism, in particular, rejects the need to identify the various parts comprising the overall cognitive system and their organisation in a manner that contributes to the overall system activity. It thereby violates the decomposition principle (Bechtel 1998a, 1998b). “Radical” dynamicists seek to identify the laws governing the “highest level relevant to an explanation of cognitive performances, whatever that may be” (van Gelder 1998: p. 619).

By contrast, connectionism and digital computationalism provide a mechanistic explanation of cognition. Models of cognitive architecture are available within each of these paradigms (yet, classicists downplay the importance of the particular physical implementation). There is certainly little reason to insist on a narrow view of computationalism as the basis for computational cognitive science²¹. But if we adopt a broad view of digital computationalism (say, one that follows from either the algorithmic execution account or the mechanistic account of computation) instead, then the result is a mechanistic explanation of cognition.

Some authors have recently rejected the view that digital computationalism and connectionism are mechanistic cognitive models. For instance, Daniel Weiskopf argues that though they have some features in common with mechanistic models proper, they crucially differ in the manner in which they relate to the modelled cognitive system (2011: p. 314). By his lights, cognitive models are causally structured, componentially organised, and

²¹ Cf. Hubert Dreyfus (1972), Searle (1980) and Steven Horst (1999) to name just a few critics of classicism.

semantically interpretable. Such cognitive models can be specified at different levels of analysis in a similar manner to full-blown mechanistic models (ibid: p. 327). However, the objection continues, cognitive models need not be mechanistic to be genuinely explanatory. For there need not always be a one-to-one correspondence of every component of the cognitive model to some real entity in the modelled system.

Whilst some cognitive models may be genuinely explanatory, if a one-to-one correspondence does not obtain, then, by the mechanistic standards, they are supposedly inadequate. In some cognitive models, which offer functional layered analyses, what matters is that there is some stable pattern of organisation in the brain that carries out the appropriate processes assigned to each layer of analysis, and has the appropriate sort of causal organisation. For example, there could be a correspondence to a whole set of resources possessed by neural regions, rather than, say, individual neurons.²² Yet, if a simple correspondence among components of the cognitive model and some neural entities in the brain does not obtain, then the model is at best incomplete, and at worst false (ibid: pp. 329-330).

The gist of the objection is that whilst digital computationalist and connectionist models are both *componential* and *causal*, they need not necessarily be mechanistic. For these models often posit elements that do not straightforwardly map onto localised parts of the modelled cognitive system (ibid: p. 332). However, as Weiskopf acknowledges himself, this objection may be countered by distinguishing between mechanistic sketches and complete

²² Weiskopf cites some researchers corroborating this claim. For example:

“[P]sychological primitives are functional abstractions for brain networks that contribute to the formation of neuronal assemblies that make up each brain state” (Lisa Barrett, as cited by Weiskopf 2011: p. 330).

“Almost every cognitive task involves the activation of a network of brain regions (say, 4-10 per hemisphere) rather than a single area” (Marcel Just et al, as cited by Weiskopf 2011: p. 330).

mechanistic models.

Connectionism typically offers cognitive explanations in terms of neural networks that need not correspond to networks of real brain neurons and synapses. A single artificial neural unit may correspond to a single *region* in the brain instead. Connectionist networks implement a task analysis without necessarily decomposing their overall operation into intelligible subtasks performed by individual components (i.e., units), which correspond to either individual brain neurons or regions. Connectionist modellers typically build their network as mechanistic models, yet they cannot give a complete mechanistic analysis of the microfeatures and microactivities that result from its adaptive weight changes during learning (Bechtel and Abrahamsen: p. 268).

Moreover, connectionist networks explain cognitive phenomena without employing localisation and decomposition. The overall performance of the network is typically not decomposable into intelligible subtasks. Instead, such networks emphasise dynamic behaviour that corresponds to the cognitive activity to be explained without the subcomponents of the system performing recognisable subtasks of the overall task. Each one of these subtasks is distributed across the layers of network and cannot be straightforwardly localised in any individual unit. In the absence of explicit rules connectionist networks have structures that are found in the networks' connections (Bechtel and Richardson 2010: pp. 217, 222-223). Yet, these networks are, at the very least, mechanistic sketches.

Why are digital computationalist models mechanistic? In digital computationalist models, functional decomposition is accomplished by modelling the target cognitive phenomenon through a series of algorithmic operations. In principle, it is easier in (non-connectionist) computational models to localise individual operations in corresponding components, due

to the nature of these models. Conventional digital computing systems are typically driven by an explicit set of rules (cf. the mechanistic account or the algorithm execution account and even the FSM and PSS accounts). Data (or symbols, on the classicist view) are manipulated by either hard- or soft-programmed instructions. For the purposes of classifying digital computationalist models as mechanistic, these instructions that manipulate data (or symbols) embody an attempt to account for the performance of the modelled cognitive system by way of decomposing the overall task into simpler subtasks.

Consider, for example, Marr's model of vision and John Anderson's ACT* production model. At the computational level, Marr's analysis specifies what is being computed and why. At the algorithmic level, the visual system is specified by means of the representations being used as well as the algorithm for transforming inputs to outputs. This level provides an explanation of the structure of visual processes. The implementation level specifies the physical realisation of the representations and algorithm (Marr 1982). Marr's tripartite model attempted to identify individual operations with specific neuroanatomical structures²³ (i.e., localising the detection of zero-crossings in cortical simple cells). Anderson's ACT* production model analyses cognitive memory function while also providing a cognitive architecture. This model consists of three components: working memory, declarative (or explicit) memory and production (or implicit) memory. This model exhibits the performance of an action as loop of encoding (into working memory), match (against a rule in production memory) and execution (in working memory) (Anderson 1983).

These computationalist models assume that the modelled mental activity is decomposable

²³ Piccinini and Craver argue that Marr's three levels are not levels of mechanism, since they do not describe relations among components or subcomponents (2011: p. 303). On their interpretation, the computational and algorithmic levels are mechanistic sketches. The computational level describes the mechanism's task and the algorithmic level describes the computational vehicles as well as the processes that manipulate these vehicles.

into a set of operations, each of which is governed by a set of instructions operating on representations (Bechtel and Richardson 2010: pp. 211-212). If a computationalist model also specifies how the relevant components are realised by neuroanatomical structures, then, by the standards of mechanistic explanation, it is a complete mechanistic model. Yet, such a direct localisation is not always practical.

Not all advocates of dynamicism share the view of it being an *alternative* to connectionism and computationalism broadly (hence the label ‘extreme dynamicism’ is used above to denote a narrower subclass of dynamicism). Beer, for one, denies the extreme dynamicist thesis that cognitive systems are best understood *only* using the tools of dynamical systems theory (forthcoming). He asserts that there is no useful mathematical distinction to be drawn among dynamicism, connectionism and (digital) computationalism. For, on the one hand, all dynamical systems can be approximated by TMs and, on the other, TMs defined over the real numbers are equivalent to dynamical systems. Similarly, recurrent neural nets can approximate arbitrary dynamical systems. He also acknowledges that it is probable that connectionism, (digital) computationalism and dynamicism will all be important in any future theory of cognition.

Yet, any mathematical distinctions aside, the mechanistic challenge remains unanswered. Connectionism and digital computationalism also make available models of cognitive architecture besides the mathematical toolbox that comes with the theory, but dynamicism proper does not. Digital computationalism need not be limited only to a specific formalism of computability, such as TMs or the lambda calculus. Formalisms of computability provide the mathematical tools required for determining the plausibility of *computational level theories*. Still, any particular formalism does not specify the relationship between *abstract* and

concrete computation. An algorithm formally specifies the relations between inputs and outputs and it can run on digital computers of various architectures. Any (classical) algorithm can (in principle) be executed on some TM. However, a TM is merely an idealisation and does not specify the physical mechanism(s) by which the algorithm is executed.

Moreover, it is at the physical level that the algorithm is converted to a program and bound by the implementing physical system. An algorithm can, in principle, produce all the natural numbers by iteratively invoking the successor function starting from 0. However, a program that implements that algorithm will eventually fail when it runs out of physical memory. TMs may help us determine whether an algorithm can be implemented on a digital computing system. But it is at the level of the physical implementation that the actual operations are analysed in consideration of the physical architecture and the primitive operations supported as well as the “real time” speed of the executed program (as opposed to the number of discrete steps in a TM). And if cognition is an embodied biological phenomenon (as granted by the dynamicist), it is *concrete* computation that plays a key role in explaining cognition and not just *computability theory*.

As observed above, dynamicism and mechanistic computational explanations are complementary. Understanding a particular mechanism but not its role in the overall dynamics of the cognitive agent (and perhaps the environment) is insufficient. Identifying a clock mechanism, for instance, in a physical computing system without discovering how it affects and is affected by the overall operation of the system only provides a partial explanation. And conversely, understanding the dynamics of the cognitive agents without identifying their constituent components provides a limited explanation at best.

This complementarity principle has yielded some collaborative effort in computational

neuroscience where an understanding of single neurons is supplemented by dynamicism. For instance, Eugene Izhikevich has applied dynamical systems theory tools in studying the relationship among electrophysiology, bifurcations and computational properties of neurons (2007). Chris Eliasmith and Charles Anderson have introduced a framework for the study of cognition in which computation, representation and dynamical systems theory all play a role (2003). They argue that modern control theory is better suited than computability theory for understanding cognition as a biological system. According to their theory, neural computation is the transformation of neural representations.

It certainly seems plausible that cognitive science has much to gain by adopting a broad perspective, which sees the above paradigms as complementary. A bottom up strategy alone will face significant challenges trying to explain how low level mechanisms give rise to high-level cognitive phenomena. A purely top down strategy may yield a viable story that explains certain phenomena without establishing how they are grounded in the human biological substrate. But such a story is difficult to conclusively refute. Still, cognitive science that draws on each of these strategies simultaneously is more likely to overcome those challenges. Time will tell.

8 Conclusion

Cognitive science faces the nontrivial task of explaining cognition. Even setting aside the question of what consciousness is or how it fits in the whole story, human cognition remains largely unexplained. As soon as it seemed that computation might help us in explaining cognition, computation became foundational to the scientific enquiry. But when we are not even clear on what computation is precisely matters only get worse. Sections 3-5 illustrate how three research programs in cognitive science invoke computation differently for

explaining cognition. Section 6 illustrates how a particular construal of computation leads another research program to dismiss it as playing a key role in cognitive science.

I have argued for enhanced clarity on how computation is invoked for explaining cognition. A blanket dismissal of the key role computation plays in cognitive science is unwarranted. Even if classicism, for example, were found untenable, *digital* computationalism could still survive. Moreover, by invoking computation as the basis for their explanatory frameworks, digital computationalism and connectionism gain not only (possible) competence level theories capitalising on mathematical formalisms, but potentially also physical cognitive architectures (or performance level theories). Dynamicism has to be complemented to provide a (complete) mechanistic explanation (see section 7 above). Computation is a general notion that offers great flexibility and its lure for explaining cognition is obvious. However, it comes at the cost of equivocation. When one makes assertions about cognition being computational (or not), one should also explicate what particular notion of computation is employed.

References:

- Anderson, J. R. (1983). The architecture of cognition. Harvard University Press.
- Bechtel, W. (1998a). Representations and cognitive explanations: Assessing the dynamicist's challenge in cognitive science. *Cognitive Science*, 22, 295-318.
- Bechtel, W. (1998b). Dynamicists versus computationalists: Whither mechanists? *Behavioral and Brain Sciences*, 21, 629.
- Bechtel, W. (2001). Representations: from neural systems to cognitive systems. In W. Bechtel, P. Mandik, J. Mundale, and R. S. Stufflebeam (Eds.), *Philosophy and the neurosciences: a reader*. Oxford: Basil Blackwell.

- Bechtel, W. (2009). Constructing a philosophy of science of cognitive science. *Topics in Cognitive Science*, 1, pp. 548–569.
- Bechtel, W. and Abrahamsen, A. (2002). *Connectionism and the mind: parallel processing, dynamics, and evolution in networks*. 2nd Edition. Oxford: Basil Blackwell.
- Bechtel, W. and Richardson, R. C (2010). *Discovering complexity: decomposition and localization as strategies in scientific research*. 2nd Edition. Cambridge: The MIT Press.
- Beer, R. (forthcoming). Dynamical systems and embedded cognition. In K. Frankish and W. Ramsey (Eds.), *The Cambridge handbook of artificial intelligence*. Cambridge University Press.
- Boden, M. A. (2008). Information, computation, and cognitive science. In P. Adriaans and J. van Benthem (Eds.), *Handbook of the philosophy of science, volume 8: philosophy of information*, pp. 741-761. Elsevier.
- Brooks, R. A. (1991). Intelligence without representation, *Artificial Intelligence*, 47, 139–159.
- Burks A. W., Goldstine, H. H., and von Neumann, J. (1946). Preliminary discussion of the logical design of an electronic computing instrument. In B. Randell (Ed.), *The origins of digital computers: selected papers*. 3rd edition (1982). pp. 399-414. New York: Springer-Verlag.
- Chalmers, D. (1992). Subsymbolic computation and the Chinese room. In J. Dinsmore (Ed.) *The symbolic and connectionist paradigms: closing the gap*, pp. 25-47. Hillsdale, NJ: Lawrence Erlbaum.
- Chalmers, D. (1993). Why Fodor and Pylyshyn were wrong: the simplest refutation. *Philosophical Psychology*, 6, pp. 305–319.

- Chomsky, N. (1992). Language and interpretation: philosophical reflections and empirical Inquiry In J. Earman (Ed.), *Inference, explanation, and other frustrations: essays in the philosophy of Science*. pp. 99–128. Berkeley: University of California Press.
- Churchland, P. S., Koch, C., and Sejnowsky, T. J. (1988). What is computational neuroscience? In E. L. Schwartz (Ed.). *Computational neuroscience*, pp. 46-55. Cambridge, MA: The MIT press.
- Churchland, P.S. and Sejnowsky, T. J. (1992). *The computational brain*. Cambridge, MA: The MIT Press.
- Clark, A. (1990). Connectionism, competence, and explanation. *The British Journal for the Philosophy of Science*, 41, pp. 195-222.
- Copeland, B. J. (1996). What is computation? *Synthese*, 108, pp. 335-359.
- Copeland, B. J. (1997). The broad conception of computation. *The American Behavioral Scientist*, 40, pp. 690-716.
- Cummins, R. (1991). The role of representation in connectionist explanations of cognitive capacities. In Ramsey, W., Stich, S. P., and Rumelhart, D. E. (Eds.), *Philosophy and connectionist theory*, pp. 91-114. New York: Lawrence Erlbaum Associates.
- Cummins, R. (1996). *Representations, targets, and attitudes*. Cambridge: The MIT Press.
- Craver, C. F. and Bechtel, W. (2006). Mechanism. In S. Sarkar & J. Pfeifer (Eds.), *Philosophy of science: an encyclopedia*, pp. 469–478. New York: Routledge.
- Dennett, D. C. (1991). *Consciousness explained*. New York: Little, Brown and Company.
- Diederich, J. (1990). Spreading activation and connectionist models for natural language processing. *Theoretical Linguistics*, 16, pp. 25-64.
- Dodig-Crnkovic, G. and Burgin, M. (2011). *Information and computation: essays on*

scientific and philosophical understanding of foundations of information and computation. Singapore: World Scientific Publishing Company.

- Dreyfus, H. (1972). What computers can't do. New York: Harper and Row.
- Egan, F. (2011). Two kinds of representational contents for cognitive theorizing. Paper presented at the 2011 Philosophy and the Brain conference at the Institute for Advanced Studies, Hebrew University in Jerusalem, Israel. Retrieved May 9, 2011, from <https://sites.google.com/site/philosophybrainias2011/home/conference-papers-1/Egan-TwoKindsofRepContent.pdf?attredirects=0&d=1>
- Eliasmith, C. (1996). The third contender: a critical examination of the dynamicist theory of cognition. *Journal of philosophical psychology*, 9, 441-463.
- Eliasmith, C. (2003). Moving beyond metaphors: understanding the mind for what it is. *The Journal of Philosophy*, 100, pp. 493-520.
- Eliasmith, C. (2007). Computational neuroscience. In Thagard, P. (Ed.) *Philosophy of psychology and cognitive science: handbook of philosophy of science*. Amsterdam: Elsevier.
- Eliasmith, C., and Anderson, C. H. (2003). *Neural engineering: computation, representation and dynamics in neurobiological systems*. Cambridge: MIT Press.
- Fodor, J. A. (1980). Methodological solipsism considered as a research strategy in cognition psychology. *Behavioral and Brain Sciences*, 3, pp. 63–109.
- Fodor, J. A. (1981). The mind-body problem. *Scientific American*, 244, 114-123.
- Fodor, J. A. and Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: a critical analysis, *Cognition*, 28, 3-71.
- Froese, T. (2011). Breathing new life into cognitive science. *Avant*, 2, pp. 113-129.
- Gandy, R. (1980). Church's thesis and principles for mechanisms. In J. Barwise, H. J.

Keisler and K. Kunen (Eds.), *The Kleene symposium*. pp. 123-148. Amsterdam: North-Holland.

- Haugeland, J. (1985). *AI: the very idea*. Cambridge, MA: The MIT Press.
- Hamann, H. and Wörn, H. (2007). Embodied computation. *Parallel Processing Letters*, 17, pp. 287-298.
- Horst, S. (1999). Symbols and computation: A critique of the computational theory of mind. *Minds and Machines*, 9, 347-381.
- Izhikevich, E. M. (2007). *Dynamical Systems in Neuroscience*. Cambridge, MA: The MIT Press.
- Kaplan, D. M., and Bechtel, W. (2011). Dynamical models: an alternative or complement to mechanistic explanations? *Topics in Cognitive Science*, 3, pp. 438–444.
- Kremer, S. C. (2007). Spatio-temporal connectionist networks. In P.A. Fishwick (Ed.). *Handbook of dynamic system modelling*. Chapman & Hall/CRC.
- Maass, W. and Markram, H. (2004). On the computational power of circuits of spiking neurons. *Journal of Computer and System Sciences*, 69, pp. 593–616.
- Machamer, P. (2004). Activities and causation: the metaphysics and epistemology of mechanisms. *International Studies in the Philosophy of Science*, 18, pp. 27–39.
- MacLennan, B. J. (2001). Connectionist approaches. In N.J. Smelser and P.B. Baltes (Eds.), *International encyclopedia of the social and behavioral sciences*, pp. 2568–2573. Oxford: Elsevier.
- MacLennan, B. J. (2004). Natural computation and non-Turing models of computation. *Theoretical Computer Science*, 317, pp. 115–145.
- Matthews, R. (1997). Can connectionists explain systematicity? *Mind and Language*, 12,

pp. 154–177.

- Marr, D. (1982). *Vision: a computational investigation into the human representation and processing of visual information*. W. H. Freeman & Co.
- McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5, pp. 115-133.
- Newell, A. (1980). Physical symbol systems. *Cognitive Science*, 4, pp. 135-183.
- Newell, A. and H. A. Simon (1976). Computer science as an empirical enquiry: symbols and search. *Communications of the ACM*, 19, pp. 113-126.
- O'Brien, G. (1999). Connectionism, analogicity and mental content. *Acta Analytica* 22, 111-131.
- O'Brien, G. and Opie, J. (2006). How do connectionist networks compute? *Cognitive processing*, 7, 30-41.
- Pfeiffer, R. and Scheier, C. (1999). *Understanding Intelligence*. Cambridge, MA: The MIT Press.
- Piccinini, G. (2008a). Computation without representation. *Philosophical Studies*, 137, pp. 205-241.
- Piccinini G. (2008b). Computers. *Pacific Philosophical Quarterly*, 89, pp. 32-73.
- Piccinini G. (2008c). Some neural networks compute, others don't. *Neural Networks*, 21, pp. 311-321.
- Piccinini G., and Scarantino, A. (2011). Information processing, computation and cognition. *Journal of Biological Physics*, 37, pp. 1–38.
- Piccinini, G. and Bahar, S. (2011). Neural computation and the computational theory of cognition. Paper presented at the Computation and the Brain workshop 2011 at the Institute

for Advanced Studies, Hebrew University in Jerusalem, Israel. Retrieved June 2, 2011, from https://sites.google.com/site/iascomputationbrainhuji2011/home/previous-lectures/Piccinini%26Bahar_NeuralComputationandtheComputationalTheoryofCognition.doc?attredirects=0&d=1

- Piccinini, G. and Craver, C. F. (2011). Integrating psychology and neuroscience: functional analyses as mechanism sketches. *Synthese*, 183, pp. 283–311.
- Poggio, T. and Koch, C. (1985). Ill-posed problems in early vision: from computational theory to analogue networks. In the proceedings of the Royal Society of London, series B, Biological Sciences, 226, pp. 303-323.
- Putnam, H. (1988). Representation and reality. Cambridge, MA: The MIT Press.
- Pylyshyn, Z.W. (1984). Computation and cognition. Cambridge, MA: The MIT Press.
- Pylyshyn, Z. (1989). Computing in cognitive science. In Posner, M. (Ed.) *Foundations of Cognitive Science*. Cambridge: The MIT Press.
- Pylyshyn, Z. W. (1993). Computers and the symbolization of knowledge. In R., Morelli, W. M., Brown, D., Anselmi, K., Haberlandt and D., Lloyd (Eds.). *Minds, brains and computers: perspectives in cognitive science and artificial intelligence*. Ablex.
- Pylyshyn, Z. W. (1999). What's in your mind? In E. Lepore and Z. W. Pylyshyn (Eds.) *What is cognitive science?* pp. 1-25. MA: Blackwell Publishers.
- Rubel, L. A. (1985). The brain as an analog computer. *Journal of Theoretical Neurobiology*, 4, pp. 73–81.
- Rumelhart, D. E., and McClelland, J. L. (1986). *Parallel distributed processing: explorations in the microstructure of cognition*. Cambridge: The MIT Press.
- Ryle, G. (1949). *The concept of mind*. London: Hutchinson.

- Searle, J. (1980). Minds, brains and programs. *Behavioral and Brain Sciences*, 3, pp. 417–424.
- Searle, J. R. (1990). Is the brain a digital computer? *Proceedings and Addresses of the American Philosophical Association*, 64, pp. 21-37.
- Sieg, W. and Byrnes, J. (1999). An abstract model for parallel computations: Gandy's thesis. *The monist*, 82, pp. 150-164.
- Siegelmann, H. T. (1999). *Neural networks and analogue computation: beyond the Turing limit*. Boston: Birkhauser.
- Siu K.Y., Roychowdhury, V., Kailath, T. (1995). *Discrete neural computation: a theoretical foundation*. Prentice Hall.
- Smolensky, P. (1988). On the proper treatment of connectionism. *The behavioural and brain sciences*, 11, 1-23.
- Smolensky, P. (1991). The constituent structure of connectionist mental states: a reply to Fodor and Pylyshyn. In T., Horgan and J. L., Tienson (Eds.), *Connectionism and the philosophy of mind*. pp. 281-308. Dordrecht: Kluwer.
- Smolensky, P. (1995). Constituent structure and explanation in an integrated connectionist/symbolic cognitive architecture. In C., MacDonald and G., MacDonald (Eds.), *Connectionism: debates on psychological explanation*, pp. 223-290. U.K.: Blackwell Publishers.
- Smolensky, P., and Legendre, G. (2006). *The harmonic mind: from neural computation to optimality-theoretic grammar*. Cambridge, MA: The MIT Press.
- Spivey, M. (2007). *The Continuity of Mind*. Oxford University Press.
- Stanley, J., and Williamson, T. (2001). Knowing how. *Journal of Philosophy*, 98, pp.

411–444.

- Stepp, N., Chemero, A., Turvey, M. T. (2011). Philosophy for the rest of cognitive science. *Topics in Cognitive Science*, 3, pp. 425–437.
- Thagard, P. (2005). *Mind: introduction to cognitive science*. 2nd edition. Cambridge: MIT Press.
- Thelen, E. and Smith, L. B. (1994). *A dynamical systems approach to the development of cognition and action*. Cambridge, MA: The MIT press.
- Thompson, E. (2007). *Mind in life: biology, phenomenology and the sciences of mind*. Cambridge, MA: Harvard University Press.
- Tienson J. L. (1988). An introduction to connectionism. *Southern Journal of Philosophy*, 26, pp. 1-16.
- Trappenberg, T. (2010). *Fundamentals of computational neuroscience*. 2nd edition. Oxford University Press.
- van Gelder, T. (1995). What might cognition be, if not computation? *The Journal of Philosophy*, 91, 345-381.
- van Gelder, T. (1998). The dynamical hypothesis in cognitive science. *The behavioural and brain sciences*, 21, 615-665.
- van Gelder, T. and Port, R. F. (1995). It's about time: an overview of the dynamical approach to cognition. In van Gelder, T. and Port, R. F. (Eds.) *Mind as motion*. Cambridge, MA: The MIT Press.
- Varela, F. J., Thompson, E., Rosch, E. (1991). *The embodied mind: cognitive science and human experience*. Cambridge, MA: The MIT Press.
- Wallace, B., Ross, A., Davies, J., Anderson, T. (2007). *The mind, the body and the world*:

psychology after cognitivism? Exeter: Imprint Academic.

- Waltz, D. L., Pollack, J. B. (1985). Massively parallel parsing: a strongly interactive model of natural language interpretation. *Cognitive Science*, 9, pp. 51-74.
- Weiskopf, D. A. (2011). Models and mechanisms in psychological explanation. *Synthese*, 183, pp. 313-338.
- White, G. (2011). Descartes among the robots: computer science and the inner/outer distinction. *Minds and Machines*, 21, pp. 179-202.
- Zednik, C. (2011). The nature of dynamical explanation. *Philosophy of Science*, 78, pp. 238-263.