CHAPTER 3

COMPUTATIONALISM UNDER ATTACK

Roberto Cordeschi and Marcello Frixione

Since the early eighties, computationalism in the study of the mind has been "under attack"[1] by several critics of the so-called "classic" or "symbolic" approaches in AI and cognitive science. Computationalism was generically identified with such approaches. For example, it was identified with both Allen Newell and Herbert Simon's Physical Symbol System Hypothesis and Jerry Fodor's theory of Language of Thought, usually without taking into account the fact ,that such approaches are very different as to their methods and aims.[2] Zenon Pylyshyn, in his influential book *Computation and Cognition*, claimed that both Newell and Fodor deeply influenced his ideas on cognition as computation.[3] This probably added to the confusion, as many people still consider Pylyshyn's book as paradigmatic of the computational approach in the study of the mind. Since then, cognitive scientists, AI researchers and also philosophers of the mind have been asked to take sides on different "paradigms" that have from time to time been proposed as opponents of (classic or symbolic) computationalism. Examples of such oppositions are:

computationalism *vs*. connectionism,
computationalism *vs*. dynamical systems,
computationalism *vs*. situated and embodied cognition,
computationalism *vs*. behavioural and evolutionary robotics.

Our preliminary claim in section 1 is that computationalism should not be identified with what we would call the "paradigm (based on the metaphor) of the computer" (in the following, PoC). PoC is the (rather vague) statement that the mind functions "as a digital computer". Actually, PoC is a *restrictive* version of computationalism, and nobody ever seriously upheld it, except in some rough versions of the computational approach and in some popular discussions about it. Usually, PoC is used as a straw man in many arguments against computationalism. In section 1 we look in some detail at PoC's claims and argue that computationalism cannot be identified with PoC. In section 2 we point out that certain anticomputationalist arguments are based on this misleading identification. In section 3 we suggest that the view of the levels of explanation proposed by David Marr could clarify certain points of the debate on computationalism. In section 4 we touch on a controversial issue, namely the possibility of developing a notion of analog computation, similar to the notion of digital computation. A short conclusion follows in section 5.

M. Marraffa, M. De Caro and F. Ferretti (eds.), Cartographies of the Mind, 37-49.

We do not debate other controversial issues here, e.g. that of so-called "pancomputationalism", which, albeit related to the topic of this chapter, would deserve a deeper analysis, and is not directly relevant to our argument. Actually, the aim of this chapter is not to deal fully with the different issues of computationalism, but to put forward a preliminary investigation of the topic, which might free it of certain common misunderstandings.

1. THE "PARADIGM OF THE COMPUTER" AND COMPUTATIONALISM

According to PoC, digital computers, considered to be the basis for explaining mental phenomena, are characterised by (at least one of) the following features:

(1) They are *sequential* machines, inspired by the von Neumann architecture. Also concurrent computers with a limited number of processing units can be accommodated within PoC. In any case, PoC's computers are machines based on a rigid distinction between memory and processing units (as is the case with von Neumann-style computers).

(2) They are *general purpose* (i.e., *universal*) computers. That is to say, they are programmable computing machines that in principle (i.e., if not subjected to temporal constraints, and if their memory is supposed to be unlimited) can compute all computable functions according to Church's Thesis.

Opponents of computationalism have an easy time criticising PoC, in that both (1) and (2) can hardly be accommodated with available data about the mind/brain. As far as (1) is concerned, the nervous system is characterised by a high degree of parallelism: in the brain, a high number of interconnected units work in parallel. Therefore, a serial model of computation would be unsatisfactory in modelling many aspects of cognition. Moreover, there is no evidence in favour of the psychological or anatomical plausibility of an architectural distinction between storage and processing of information. As far as point (2) is concerned, empirical data favour the claim that at least some parts of the cognitive architecture consist of specialised and possibly anatomically localised modules. These, from a computational point of view, can be considered *dedicated* computational devices rather than processes implemented on a universal computer.

Both (1) and (2) have been preferred targets of various opponents of classic AI and cognitive science, starting with the early supporters of connectionism in the eighties[4]. However, it is worth noting that nobody in the field of classic AI or cognitive science has ever seriously claimed that the mind/brain functions "as a von Neumann computer" (or "as a Turing machine"). Even Fodor and Pylyshyn widely argued that the issue of the debate was not this "absurd assumption" (as a matter of fact, a mere metaphor), but whether a connectionist *explanation* of the mind is possible, and perhaps more suitable than the classic one.[5]

Thus *computationalism is not PoC*. The central claim of computationalism is that mental processes are computations. More precisely, the theoretical constructs of a theory of the mind are both the computational processes that are supposed to occur in the mind and the data structures ("representations") that such processes manipulate. According to the orthodox version of computationalism, mental processes are effective or algorithmic processes in the sense of computability theory, i.e. processes that—according to Church's Thesis—compute partial recursive (or, equivalently, Turing-computable) functions[6].

Algorithmic, or effective, computations in the above sense are *digital* processes that manipulate discrete entities. In principle, some extended notion of computation could also be considered, which includes *analog computation*. We shall discuss this point later in the chapter (see section 5). At the moment, we shall consider only digital computations.

Even in its orthodox form (that takes into account only digital computations, and identifies computability with Turing-computability) computationalism can fully negate both (1) and (2) above. As regards (1), parallel, non von Neumann architectures are compatible with a computational stance, as results, for example, from the analysis of the notion of algorithm developed by Robin Gandy[7]. As for (2), the claim that a certain device performs an algorithmic computation is fully legitimate, even if it is not a general purpose computer. Computationalism is compatible with the thesis that the mind/brain is (entirely or in part) built up of modules that are special purpose computational devices. David Marr, who probably developed the first full-blown version of computationalism, was a strong supporter of modularism (we shall take Marr's computationalist stance into account in greater detail below).

It is not our aim to take sides here on the role of computational universality in cognitive science. For our present purposes, it is relevant that a computationalist *can* deny that the human mind/brain, or some part of it, is a universal computer. Different positions are possible on this issue. Followers of weak modularism would agree that *certain* parts of the human mind/brain (for example, the input and output modules) are dedicated computational devices, but that this is not true for central cognition.[8] Followers of strong modularism would claim that the *entire* human mind/brain is made up of dedicated computational components. Putting aside human cognition, one might agree that a computational explanation of the abilities of simple cognitive systems (e.g., insects) is possible, without assuming that such systems are universal computational devices.

2. TROUBLES WITH COMPUTATIONALISM

Thus, we have concluded that computationalism is not PoC. However, such oppositions as those mentioned at the beginning of the present chapter are often based on the identification of computationalism with PoC. In our opinion, these oppositions are based on the fact that a *restrictive* view of computationalism is assumed as a polemical target. As an example we consider here the opposition between dynamical and computational explanations in cognitive science, as put forward by Tim van Gelder. On the one hand, van Gelder states:

(a) dynamical systems “can compute, i.e., be computers”, but “effective computation is a specific kind of computation, resulting from a certain kind of constraint on the processes involved”; moreover “it can be proved that certain classes of dynamical systems are *more* powerful—can compute a wider class of functions—than Turing Machines. So, dynamical systems can compute, i.e., be computers, without needing to be digital computers”.[9]

On the other hand, he acknowledges that:

(b) “most if not all dynamical systems of practical relevance to cognitive science are effectively computable”, in the sense that their behaviour “is governed by some computable function”, i.e., some partial recursive, or Turing-computable, function.[10] So, “no dynamicist in cognitive science (to my knowledge) […] has taken up dynamical modelling on the promise of super-Turing capacities”.[11]

Summing up, on the basis of (a) and (b), we can conclude that dynamical systems *can* be computers, and that their behaviour (at least as far as cognition is concerned) can be described in terms of (Turing-)computable functions. However, van Gelder claims that dynamicism in cognitive science is not compatible with a computational approach. This is because, for van Gelder, “dynamic” computation is profoundly different from common digital computation. Dynamical systems “compute” recursive functions; but they perform such computations in a different way from digital computers. How could this claim be justified? As far as we can see, it might be done in one of the following ways.

(i) A first possibility is to suppose that the functioning of cognitive dynamical systems depends on some “hidden” non recursive process. In other words, given a dynamical system DS as in Figure 3.1, the functioning of DS would depend on at least one component S, whose behaviour exceeds the limits of effective computability (in the sense that the behaviour of S can not be described by a partial recursive function). However, this “non recursive” behaviour is not visible from the outside. For example, let us suppose that S computes the values of a function f_S: $\mathbf{N} \rightarrow \mathbf{N}$ such that, for every x, $f_S(x)$ is the x^{th} decimal figure of a certain non (Turing-) computable real number n. By definition, f_S is not a Turing-computable function. Therefore, DS functioning is not algorithmic. However, let us suppose that the output of S has some effect on the overall behaviour of DS if and only if the input of S is in a certain finite range (say, is less or equal to 10). In the other cases, the operations of S have no effect on the output of DS. All the other processes in DS are fully algorithmic. Therefore, DS computes an effectively (Turing-)computable function (S could be replaced by a look-up table).

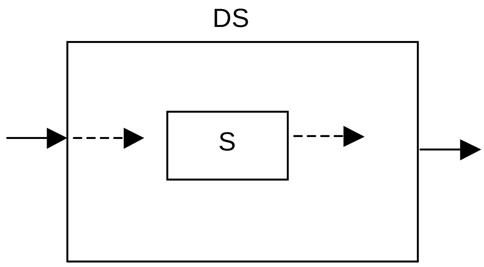

Figure 3.1. A dynamical system (DS). See text for full explanation.

(ii) A second possibility is that van Gelder's opposition is based on a restricted interpretation of what counts as digital computation, i.e., on what we called above a restricted view of computationalism. In this case, the notion of "digital computation" that van Gelder contrasts with dynamicist computation would include some specific, restricted architectural assumption, and would be heavily biased towards PoC.

Position (i) should be supported by very strong justifications; otherwise the hypothesis of "hidden" non algorithmic processes (i.e., non algorithmic process that have no influence on the input-output behaviour of the system) would fall under Occam's razor. In any case, it is unlikely that van Gelder endorses this thesis. According to his words, he does not even consider it crucial that dynamical systems adopted in cognitive science be continuous rather then discrete systems.[12] Therefore, it seems that his claims about computationalism can be traced back to (ii), i.e., to a restricted view of computationalism.

The comparison between Watt's regulator and a computational regulator in van Gelder (1995) shows that many of van Gelder's criticisms to computationalism stem from a restricted view of such an approach.[13] One of the features that would distinguish a computational version of the regulator from Watt's regulator is that the former, and not the latter, is "sequential and cyclic", in van Gelder's words. But, in general, algorithmic processes are not necessarily sequential. William Bechtel argues that frequently the explanation of complex processes has an initially sequential structure, and only later – when a better understanding of the process is achieved – are more complex interaction schemes developed (for Bechtel, models of fermentation give an example of this kind of evolution).[14] This, however, does not mean that a given explanation ceases to be mechanistic in nature. Also in this case, an aspect of a restricted class of algorithms (i.e., sequentiality) is assumed to be characteristic of computationalism (probably, it is not by chance that sequential computation is seen as a distinctive feature of PoC).

3. KINDS OF COMPUTATIONAL EXPLANATIONS

As pointed out above, many arguments against computationalism are based on disagreements concerning computational architecture. The same could be said of the format of representations. Sometimes, computationalism is identified with the choice of a particular kind of representation, typically "language-like" representations characteristic of classic AI and cognitive science (logic-based representations, production rules, semantic networks, frames, and so on). These representations are processed by explicit manipulation rules. Representations with a less "linguistic" structure (firstly, distributed or "subsymbolic" connectionist representations) have been considered less akin to computationalism. These claims, too, usually stem from a restricted view of computationalism. Computationalism, *per se*, is not compromised by any particular kind of representation or process (once accepted that they are effective processes). Frequently, these disputes come from confusion concerning different levels of explanation. The analysis of the levels of explanation in cognitive science developed by David Marr may be useful here.

According to Marr (1982, chapter 1), a computational explanation can be stated at three different levels, the *level of the computational theory*, the *algorithmic level* and the *implementation level*.[15] The level of the computational theory is the most abstract; it is concerned with the specification of the task of a certain cognitive phenomenon. At this level, cognitive tasks are characterised only in terms of their input, their output, and the goal of the computation, without any reference to specific cognitive processes and mechanisms. In other words, at the level of computational theory a cognitive task is accounted for in terms of a functional mapping between inputs and outputs. The algorithmic and the implementation levels deal, at different levels of abstraction, with the specification of the task identified at the computational level. The algorithmic level explains "how" a certain task is carried out: it deals with the computational processes and with the processed data structures (i.e., the "representations"). The implementation level deals with the physical features of the device (e.g., neural structures) implementing the data structure and the procedures singled out at the algorithmic level. The relationship between computational theory and algorithmic level is the same as that existing between a mathematical function and an algorithm that computes its values.

The aim of a computational theory is the individuation of a (computable) function f as a model of a given cognitive phenomenon. At the computational theory level, no assumption is made on the algorithms that compute f, nor, *a fortiori*, on their implementation. The role of the computational level is to allow a more abstract understanding of cognitive phenomena: the computational explanation of a cognitive phenomenon cannot be reduced to the exhibition of an algorithm (or, worse, a computer program) that simulates its behaviour (as happens, for Marr, in many alleged cognitive models developed in AI).

Summing up, at the most abstract level (the level of computational theory, in Marr's terminology) a computational account is completely neutral with respect to the mechanisms adopted (kinds of representation, data structures, algorithms and processes). For example, at this abstract level classic and connectionist theories cannot be discriminated on the basis of the fact that the former adopt "linguistic" representations while the latter are based on, say, "subsymbolic" representations. Nor is appealing to parallel rather than serial computation relevant[16]. Adopting certain kinds of representations and processes instead of others (provided that they are effective processes) is not sufficient, *per se*, to exceed the boundaries of computationalism.

There is a further aspect according to which Marr's analysis can be relevant here. The opposition between computational and dynamicist approaches has sometimes been formulated in terms of different kinds of explanation. Mechanistic explanations that are typical of the computational approach aim at explaining cognitive phenomena in terms of the mechanisms (representations or data structures and processes) determining them. According to the dynamicist approach, to explain a given phenomenon is equivalent to identifying the laws governing it, i.e., the equations that describe its evolution through time[17]. In this sense, the dynamicist explanation would be more homogeneous to traditional nomological-deductive explanations adopted, for example, in physics[18].

However, the computational approach is not in principle incompatible with the kind of explanation favoured by dynamicists. In Marr's hierarchy, the computational explanation – in which cognitive phenomena are characterised solely in terms of the functional correspondences between inputs and outputs – is homogeneous with the "traditional" explanation stated in terms of systems of equations. Therefore (if we do not assume that the equations governing the dynamics of cognitive systems are *not computable*, but as seen above, this is not the case for van Gelder), the dynamicist approach *per se* does not offer a different kind of explanation. It gives up a further advantage that the computational approach can offer us, i.e., mechanistic explanations in terms of algorithms and representations.

One could ask whether dynamicists (and other opponents of classic cognitive science, e.g. situated cognition theorists) can avoid the level of representations and algorithms. In other words, for which phenomena would a truly dynamicist explanation that makes no hypothesis on underlying processes turn out to be really satisfactory? And to what extent do examples of explanation proposed by dynamicists really leave mechanisms or processes out of consideration? These questions are at the core of a lively debate in cognitive science, but are beyond the aims of this chapter.

Bechtel's view seems to support our claim on dynamicist explanation. Let us suppose that a dynamicist theory for some cognitive phenomenon has been developed. At this point, according to Bechtel, a further question arises: "How is the underlying system able to instantiate the laws identified in these [dynamicist] accounts? One way to answer this question is to pursue a mechanistic explanation".[19] Here dynamicist and mechanistic explanations complement one another. A further role of dynamicist explanations with respect to mechanistic ones would be that of providing a preliminary understanding of the behaviour of the system being studied: "It is helpful to have a good description of what a system is doing before trying to explain how it does it".[20] Such a preliminary understanding can be given by a dynamicist explanation. From the above quotations the analogies clearly emerge between (a) the role of dynamicist explanations with respect to mechanistic models in Bechtel's view, and (b) the role of computational theories in Marr's methodology[21].

Summing up, different positions are possible, which can be summarised by the following table:

van Gelder	Bechtel	Marr
Dynamic-equation systems	*Dynamic-equation systems*	*Computational theory*
	Mechanistic explanation	*Algorithms and representation level*

In conclusion, according to van Gelder and the supporters of the dynamical systems theory, dynamicists' explanations are completely unrelated to the computational/mechanistic approach, and incompatible with it. Dynamicist explanations have no mechanistic counterpart. According to Bechtel, dynamicist and computational explanations, far from being incompatible, are complementary and can be fully integrated. Finally, in a more radical way, the level of explanation of dynamicism can be considered as part of a computational explanation. This is the case of Marr's methodology, that is to say, of a version of computationalism that does not reduce computational explanations to the mere individuation of algorithms, or, even worse, to the mere design of computer programs.

4. ANALOG COMPUTATIONS

The possibility of adopting analog processes and representations in cognitive explanations is a particularly tricky problem for the computationalist[22]. By *analog processes* (*representations*) we mean processes (representations) based on continuous quantities. This topic also plays some role in many debates opposing classic cognitive science in favour of various alternative "paradigms". The adoption of some notion of analog computation has been discussed from different points of view, for example by supporters of the dynamicist approach, or by connectionists such as Churchland and Sejnowski (1992).

The transition to analog computation involves a great discontinuity with traditional computability theory which, as said above, is based on the hypothesis that computational notions are defined in terms of operations on discrete quantities.

Now the shift from digital to analog seems not to involve giving up a computational approach. Consider the case of Kenneth Craik, the Cambridge psychologist who is usually considered as a forerunner of computationalism in the study of mental processes. In 1943, he formulated his "symbolic theory of thought" in terms of computations on analog symbols. His definition of models, that later became popular within classic cognitive science, is associated to the notion of *simulation* as performed by analog computers, which were the prevailing computational devices at the time. An analog computer can "imitate" a natural or mental phenomenon by reproducing certain "essential features" (in Craik's words), and ignoring others, that are not essential for simulation.

Craik's examples are Vannevar Bush's differential analyser, Lord Kelvin's tide predictor, self-directing anti-aircraft guns, and small-scale models of human-made artefacts, such as a bridge or a boat. An external process, such as the design of a bridge or the rising of tides, is physically realised as a device in which states of the process are "translated" as "representations by symbols" or "representatives" in input, and then manipulated by suitable rules or procedures. Finally, as output one has a "'retranslation' of these symbols into external processes (as in building a bridge to design) or at least [a] recognition of the correspondence between these symbols and external events (as in realising that a prediction is fulfilled)".[23] Such a device is the model (or rather, the *working model*) of the external process, and symbols must be meant to have a very general sense: symbols are not only words or numerals, but can also be, for example, positions of gears in a mechanism, whose "mechanical process" parallels the external process, thus *causing* the transition from one state to another.[24]

A remark is needed here about the use of the term "analog". This term can be used with at least two different meanings that are in some way related, but that do not fully coincide.[25] According to the first, analog processes are based on the manipulation of continuous quantities. Here "analog" is opposed to "digital". According to the second, closer to Craik's, analog models depend on some "resemblance" relation existing between the representations and what is represented.[26] Here "analog" is opposed to "propositional" or "symbolic" (though, as seen above, this use of "symbolic" does not coincide with Craik's terminology).

If a system is analog in the latter meaning but not in the former (i.e., if its representations are based on some form of "resemblance", but they are made up by discrete elements), then there is no problem in considering it a computational system in all respects (taken for granted, of course, that its evolution is governed by effective processes according to Church's thesis). Johnson-Laird's mental models are an example of this position: they are analog representations in that they "resemble" what they represent; however, they are digital, and therefore they are *not* analog according to the former of the meanings mentioned above.[27]

A more complex issue is the case of systems that are analog in the first meaning (i.e., systems based on the manipulation of continuous quantities, leaving aside the fact whether they are analog according to the second meaning or not). All computable processes according to Church's thesis are discrete. Continuous quantities can at best be approximated in digital terms, but cannot be coded without error in digital terms (the cardinality of the continuum is greater than the cardinality of countable sets). As a consequence, processes based on continuous quantities exceed the limits of the orthodox notion of computation (i.e., the notion of computation based on Church's thesis).

There are many ongoing research projects on the foundations of analog computation, which aim at characterising it in a rigorous mathematical way.

Different possibilities have been explored. For example, so-called recursive analysis is aimed at extending the class of partial recursive functions to a class of functions with real arguments and values. Another line of research is inspired by the *General Purpose Analog Computer* (GPAC), a model of analog computation proposed in 1941 by Claude Shannon with the aim of giving a precise mathematical characterisation of Bush's differential analyser.[28]

The problem with such attempts is that, contrary to what happens in the case of digital computation, a general notion of analog computation does not emerge. In other words, up to now no class of real functions has been identified, that, regarding analog computation, plays the same role played by the class of partial recursive functions in the case of digital computation. Thus, a class of real functions which is stable and invariant with respect to different ways of characterising analog computations does not exist. Different notions of analog computation result in different classes of real functions.

This state of affairs is rather discouraging. When one claims that a certain analog model (in the sense of a model based on the processing of continuous quantities) is a computational model, it is not immediately clear what this means (or, rather, it is considerably less clear than in the digital case). In other words, it is much more difficult to establish the extent to which we are still within the boundaries of computation, and when such boundaries have been exceeded.

Summing up, if compared with digital computation, at the moment the theoretical framework of analog computation is still rather confused. Therefore, it is our opinion that the choice of relinquishing digital in favour of analog computation should be based on very strong theoretical grounds. Generic considerations such as "the brain is an analog rather than digital device" are not sufficient. The fact that a cognitive system changes in a "smooth" way is not compelling. By adopting functions that take their arguments and values in the set of rational numbers, one can represent changes that are as "smooth" as desired. The set **Q** of rational numbers is *dense*: given any two rational numbers, there is always a rational number between them. But **Q** is still a countable set (any rational number can be represented as a pair of natural numbers), and functions of the type $\mathbf{Q} \rightarrow \mathbf{Q}$ can therefore be computed by a digital device. The need of going over to the set **R** of real numbers, and, therefore, to give up digital quantities in favour of continuous quantities, can be motivated by the need of adopting the methods and the theoretical apparatus of mathematical analysis. The need for such methods in the modelling of cognitive phenomena is still an open question.

5. CONCLUSION

In this chapter we have pointed out some misunderstandings in putting computationalism under attack.

Criticisms have often been based on a particularly rigid, restrictive view of computationalism, considered as a "paradigm" artfully opposed to other alleged "paradigms". It is beyond doubt that there are deep differences between the approaches that have from time to time been proposed, ranging from "classic" cognitive science, to connectionism and dynamicism (we have not dealt with other possible contenders here). However, opposition of the kind "computationalism vs. something else" is misleading. It does not account for the actual differences between those approaches, and for the as yet unresolved problems.

Despite certain restrictive views, computationalism is not as rigid as it was described by some of its early supporters and some of its recent opponents. Probably, the main problem to be resolved is not what cognition might be if not computation, but what *kind* of computation might be cognition[29].

NOTES

[1] We have borrowed this expression from Scheutz (2002).

[2] Suffice it to consider here Fodor's strong criticism to "Wagnerian" (i.e., *classic*) AI and to the very concept of behaviour *simulation* (Fodor, 1983).

[3] Pylyshyn (1984, pp. xxi-xxii).

[4] Consider, for example, the following quotations: "The *dissimilarity* between computers and nervous systems […] have made the metaphor that identifies the brain with a computer seem more than a trifle tin. […] The known parallel architecture of the brain and the suspected distributed nature of information storage has suggested to some researches that greater success in understanding cognitive functions might be achieved by a radical departure from the sequential stereotype. The idea has been to try to understand how interconnected neuron-like elements, simultaneously processing information, might be accomplish such tasks as pattern recognition and learning" (Churchland 1986, pp. 458-459). "In considering the brain as a Turing machine, we must confront the unsettling observation that, for a brain, the proposed table of states and state transitions is unknown […], the symbols on the input tape are ambiguous and have no preassigned meanings, and the transition rules, whatever they may be, are not consistently applied […] It would appear that little or nothing of value can be gained from the application of this failed analogy between the computer and the brain" (Edelman 1992, p. 227).

[5] Fodor and Pylyshyn (1988, pp. 50-64).

[6] Within computability theory, effective processes are characterised in terms of a class of arithmetic functions, the so-called partial recursive, or Turing computable, functions. Sometimes the objection has been raised that identifying computation with computation of the values of a function is restrictive (see for example Scheutz 2002). Such an identification is suitable only when input data are provided at the beginning of the computation, and outputs are produced at the end. The majority of computer implemented algorithms do not work in this way: in most cases computer programs continue to interact with the user and/or with their environment, taking new inputs and producing new outputs, until the computation ends. Similar considerations also hold for the algorithms employed within cognitive science. These phenomena are the topic of the field of research called *interactive computation*. However, these aspects are not strictly relevant for our present argument, and we do not take them into account here.

[7] Gandy (1980) developed a very comprehensive analysis of algorithmic computation, which also includes parallel computing processes. This analysis resulted in a further confirmation of Church's Thesis. Gandy individuated a number of very general constraints that every algorithmic process must satisfy. The computing devices that obey such constraints are called *Gandy machines*. Turing machines turn out to be a special case of Gandy machine. However, it can be proved that a Turing machine can do whatever can be done by a Gandy machine (in other words, all functions that can be computed by a Gandy machine can also be computed by a Turing machine).

[8] See, e.g., Fodor (1983).

[9] van Gelder (1998a, sections 6.3 and 6.10).

[10] Ibid., section 6.4.

[11] van Gelder 1998b, section 1.3.

[12] Ibid., section 1.4.

[13] van Gelder (1995).

[14] Bechtel (1998, section 5).

[15] See this volume, p. 9.

[16] This does not mean that connectionism is simply an implementation theory, as stressed by certain early supporters of "classic" cognitive science. On the one hand, the above mentioned opposition regards representations and algorithms, not implementation issues. On the other hand, other features might be used to distinguish classic and connectionistic explanations at the computational level, e.g. connectionists' emphasis on learning and the statistical side of cognition.

[17] One has differential equations in the case of continuous dynamical systems and difference equations in the case of discrete dynamical systems. As to differential equations, a mechanical (algorithmic) model can only approximate their behaviour. We do not deal with this issue at the moment. As seen above, for van Gelder the distinctive feature of the dynamical approach does not consist in the use of continuous rather than discrete systems (van Gelder 1988b, section 1.4).

[18] See for example van Gelder's (1998a) distinction between "Hobbesian", i.e., mechanistic, and "Humean" explanations in psychology. On this topic see also Bechtel (1998, sections 4 and 5) and Beer (2000, pp. 96-97).

[19] Bechtel (1998, p. 312)

[20] Ibidem.

[21] Bechtel sees the role of dynamical explanations in the development of mechanical models as analogous to that of the ecological requirement stressed by certain cognitivist psychologists (e.g., Ulric Neisser). He observes that "the language of ecological validity is drawn from James Gibson, and it is noteworthy that several of today's DST [Dynamical System Theory] theorists […] are also neo-Gibsonians" (Bechtel 1998, p. 312). It is noteworthy also that Marr considered Gibson to be "perhaps the nearest anyone came to the level of computational theory" (Marr 1982, p. 29). But Gibson "was mislead by the apparent simplicity of vision" (p. 30), so disregarding the mechanistic side of the theory (i.e., explanations in terms of representations and algorithms). Gibson, Marr concluded, "did not understand properly what information processing was, which led him to seriously underestimate the complexity of the information-processing problems involved in vision" (p. 29).

[22] For a recent point of view on this topic see Trautteur (2005).

[23] Craik (1943, p. 50).

[24] See Cordeschi (2002, chapter 4) for further details.

[25] See, e.g., Pylyshyn (1984, pp. 199 ff.).

[26] O'Brien (1998) calls this thesis *structural isomorphism*. For a partially similar position, see Trenholme (1993).

[27] Johnson-Laird (1983). Analog models in the second meaning can in turn be combined with both propositional and connectionist representations, still remaining within the boundaries of digital computation. See for example Chella, Frixione and Gaglio (1997, 2000) for a hybrid model in the field of artificial vision and robotics, which combines analog models with both propositional representations and connectionist networks.

[28] On these topics see for example Pour-El and Richards (1989); Weihrauch (2000).

[29] Thanks to Diego Marconi, Massimo Marraffa, Teresa Numerico, Dario Palladino and Giuseppe Trautteur for useful critical remarks on previous versions of this chapter.

REFERENCES

Bechtel, W. (1998). Representations and cognitive explanations: Assessing the dynamicist challenge in cognitive science. *Cognitive Science*, 22, 295-318.

Beer, R.D. (2000). Dynamical approaches to cognitive science. *Trends in Cognitive Sciences*, 4(3), 91-99.

Chella, A., Frixione, M. and Gaglio, S. (1997). A cognitive architecture for artificial vision. *Artificial Intelligence*, 89, 73-111.

Chella, A., Frixione, M. and Gaglio, S. (2000). Undertsanding dynamic scenes. *Artificial Intelligence*, 123(1-2), 89-132.

Cordeschi, R. (2002). *The Discovery of the Artificial: Behavior, Mind and Machines Before and Beyond Cybernetics*, Kluwer, Dordrecht.

Churchland, P.S. (1986). *Neurophilosophy: Toward a Unified Science of the Mind-Brain*, MIT Press, Cambridge, MA.

Churchland, P.S. and Sejnowski, T.J. (1992). *The Computational Brain*, MIT Press, Cambridge, MA.

Craik, K.J.W. (1943). *The Nature of Explanation*, Cambridge University Press, Cambridge.

Fodor, J.A. (1983). *The Modularity of Mind*, MIT Press, Cambridge, MA.

Edelman, G.M. (1992). *Brillant Air, Brillant Fire: On the Matter of the Mind*, Basic Books, New York.

Fodor, J.A. and Pylyshyn, Z.W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28: 3-71.

Gandy, R. (1980). Church's thesis and principles for mechanisms, in J. Barwise, H.J. Keisler and K. Kunen (eds.), *The Kleene Symposium*, North Holland, Amsterdam, 123-148.

van Gelder, T.J. (1995). What might cognition be, if not computation? *Journal of Philosophy*, 91, 345-381.

van Gelder, T.J. (1998a). The dynamical hypothesis in cognitive science. *Behavioral and Brain Sciences*, 21, 1-14.

van Gelder, T.J. (1998b). Disentangling dynamics, computation, and cognition. *Behavioral and Brain Sciences*, 21, 40-7.

Johnson-Laird, P.N. (1983). *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*, Harvard University Press, Cambridge, MA.

Marr, D. (1982). *Vision*, Freeman, New York.

Newell, A. (1990). *Unified Theories of Cognition*, Harvard University Press, Cambridge, MA.

O'Brien, G. (1998). Connectionism, analogicity and mental content. *Acta Analytica*, 22, 111-131.

Pour-El, M.B. and Richards, J.I. (1989). *Computability in Analysis and Physics*, Springer, Berlin.

Pylyshyn, Z.W. (1984). *Computation and Cognition: Toward a Foundation for Cognitive Science*, MIT Press, Cambridge, MA.

Scheutz, M. (ed.) (2002). *Computationalism: New Directions*, MIT Press, Cambridge, MA.

Trautteur, G. (2005). Beyond the Super-Turing Snare: Analog Computation and Digital Virtuality, in S.B. Cooper, B. Löwe and L. Torenvliet (eds.), *New Computational Paradigms*, LNCS 3526, Springer, Berlin, 507–514.

Trenholme, R. (1994). Analog simulation. *Philosophy of Science*, 61, 115-131.

Weihrauch, K. (2000). *Computable Analysis: An Introduction*, Springer, Berlin.