Beyond Computationalism

Marco Giunti
Dept. of History and Philosophy of Science
Goodbody Hall 130, Indiana University
Bloomington, IN 47405
giunti@dada.it

Abstract

By *computationalism* in cognitive science I mean the view that cognition essentially is a matter of the computations that a cognitive system performs in certain situations. The main thesis I am going to defend is that computationalism is only consistent with symbolic modeling or, more generally, with any other type of computational modeling. In particular, those scientific explanations of cognition which are based on *(i)* an important class of connectionist models or *(ii)* nonconnectionist continuous models cannot be computational, for these models are not the kind of system which can perform computations in the sense of standard computation theory. Arguing for this negative conclusion requires a formal explication of the intuitive notion of computational system. Thus, if my thesis is correct, we are left with the following alternative. Either we construe computationalism by explicitly referring to some nonstandard notion of computation, or we simply abandon the idea that computationalism be a basic hypothesis shared by all current research in cognitive science. I will finally suggest that a different hypothesis, *dynamicism*, may represent a viable alternative to computationalism. According to it, cognition essentially is a matter of the state evolutions that a cognitive system undergoes in certain situations.

Introduction

By *computationalism* in cognitive science I mean the view that cognition essentially is a matter of the computations that a cognitive system performs in certain situations. The main goal of this paper is to assess whether this view may represent a basic hypothesis shared by the three current approaches to cognition: the symbolic (or classic) approach, connectionism, and nonconnectionist dynamics.

If we look at the models actually used in cognitive science, we see that a different type of model corresponds to each approach. The symbolic approach (Newell and Simon, 1972; Newell, 1980; Pylyshyn, 1984; Johnson Laird, 1988) employs symbolic processors as models. As a first approximation, we may take a symbolic processor to be any device that operates effective transformations of appropriately defined symbol structures. The connectionist approach (Rumelhart and McClelland, 1986), on the other hand, employs connectionist networks, while nonconnectionist dynamicists use other kinds of continuous systems specified by differential (or difference) equations. Nonconnectionist researchers favoring a dynamical perspective are active in many fields. For examples see Port and van Gelder (1995).

The main thesis I am going to defend is that computationalism is only consistent with symbolic modeling or, more generally, with any other type of computational modeling. In particular, those scientific explanations of cognition which are based on *(i)* an important class of connectionist models or *(ii)* nonconnectionist continuous models cannot be computational, for these models are not the kind of system which can perform computations in the sense of standard computation theory.

The thesis that computationalism is only consistent with computational modeling is empty unless one gives a sufficiently precise characterization of what a *computational model* of a cognitive system is. By this term, I mean any computational system that describes (or, at least, is intended to describe) some cognitive aspect of the cognitive system. Intuitively, by the term *computational system* I refer to any device of the kind studied by standard computation theory. Thus, for example, Turing machines, register machines, and finite state automata are three different types of computational systems. By contrast, so-called analog computers are not computational systems. I will propose later a formal explication of this intuitive notion of a computational system.

Thus, if my thesis is correct, we are left with the following alternative. Either we construe computationalism by explicitly referring to some nonstandard notion of computation, or we simply abandon the idea that computationalism be a basic hypothesis shared by *all* current research in cognitive science. In the last section of this paper, I will also suggest that a different hypothesis, *dynamicism*, may represent a viable alternative to computationalism. According to it, cognition essentially is a matter of the state evolutions that a cognitive system undergoes in certain situations.

The Argument

The main thesis of this paper is that computationalism is only consistent with symbolic modeling or, more generally, with any other type of computational modeling. The argument I am going to propose is based on two premises. The first one affirms that all models currently employed in cognitive science are *mathematical dynamical systems*. The second premise, on the other hand, affirms that a computation (in the sense of standard computation theory) can only be performed by that special type of mathematical dynamical system which I have called a *computational system*. Having established these two premises, I will then show that *(a)* an important class of connectionist models, and *(b)* nonconnectionist continuous models are not computational systems. Hence, these models cannot perform computations in the standard sense. But then, if our scientific explanations of cognition are based on these models, we cannot maintain that cognition is, essentially, a matter of the computations performed by the cognitive system which these models are intended to describe. On the other hand, *(c)* all symbolic models are computational systems. Therefore, computationalism is only consistent with symbolic modeling or, more generally, with any other approach which employs computational systems as models of cognition.

The First Premise

The first premise of my argument is that all models currently employed in cognitive science are mathematical dynamical systems. A *mathematical* dynamical system is an abstract mathematical structure that can be used to describe the change of a real system as an evolution through a series of states. If the evolution of the real system is deterministic, that is, if the state at any future time is determined by the state at the present time, then the abstract mathematical structure consists of three elements. The first element is a set T that represents time. T may be either the reals, the rationals, the integers, or the nonnegative portions of these structures. Depending on the choice of T, then, time is represented as continuous, dense, or discrete. The second element is a nonempty set M that represents all possible states through which the system can evolve; M is called the *state space* of the system. The third element is a set of functions $\{g^t\}$ that tells us the state of the system at any instant t provided that we know the initial state; each function in $\{g^t\}$ is called a *state transition* of the system. For example, if the initial state is $x \in M$, the state at time t is given by $g^t(x)$, the state at time $u > t$ is given by $g^u(x)$, *etc.* The functions in the set $\{g^t\}$ must only satisfy two conditions. First, the function g^0 must take each state to itself and, second, the composition of any two functions g^t and g^w must be equal to the function g^{t+w}

An important subclass of the mathematical dynamical systems is that of all systems with discrete time. Any such system is called a cascade. More precisely, a mathematical dynamical system $<T, M, \{g^t\}>$ is a *cascade* just in case T is equal to the nonnegative integers (or to the integers).

As mentioned, the models currently employed in cognitive science can basically be classified into three different types: *(1)* symbolic processors, *(2)* neural networks, and *(3)* other continuous systems specified by differential (or difference) equations. That a system specified by differential or difference equations is a mathematical dynamical system is obvious, for this concept is expressly designed to describe this class of systems in abstract terms. That a neural network is a mathematical dynamical system is also not difficult to show. A complete state of the system can in fact be identified with the activation levels of all the units in the network, and the set of state transitions, on the other hand, is determined by the differential (or difference) equations that specify how each unit is updated. To show that all symbolic processors are mathematical dynamical systems is a bit more complicated.

The argumentative strategy I prefer considers first a special class of symbolic processors (such as Turing machines, or monogenic production systems, *etc.*) and it then shows that the systems of this special type are mathematical dynamical systems with discrete time, *i.e.*, cascades. Given the strong similarities between different types of symbolic processors, it is then not difficult to see how the argument given for one type could be modified to fit any other type (Giunti, 1992, 1996). We may thus conclude that all models currently employed in cognitive science are mathematical dynamical systems.

The Second Premise

The second premise of my argument affirms that a computation (in the sense of standard computation theory) can only be performed by a *computational system*. Intuitively, by this term I refer to any device of the kind studied by standard computation theory (*e.g.*, Turing machines, register machines, cellular automata, *etc.*) I call any computation performed by any such device a *standard computation*. According to this terminology, then, my second premise affirms that *a standard computation can only be performed by a computational system*. It is thus clear that I in fact take this premise to be true by definition.

Someone might object that, given my definitions, my second premise is not only true, but also trivial. According to my imaginary critic, the important question is not whether a standard computation can be performed by a noncomputational system but, rather, whether standard computational methods are sufficient to accurately describe the behavior of *all* models employed in cognitive science (be these models computational or not). I will give an answer to this kind of objection later. Before I can proceed with my argument, however, I need to give a formal explication of the intuitive concept of a computational system.

A Formal Definition of a Computational System

To this extent, let us first of all consider the mechanisms studied by standard computation theory and ask *(i)* what type of system they are, and *(ii)* what specific feature distinguishes these mechanisms from other systems of the same type.

As mentioned, standard computation theory studies many different kinds of abstract systems. A basic property that is shared by all these mechanisms is that they are *mathematical dynamical systems with discrete time*, that is *cascades*. However, standard computation theory does not study all cascades. The specific feature that distinguishes computational systems from other mathematical dynamical systems with discrete time is that a computational system *can always be described in an effective way*. Intuitively, this means that the constitution and operations of the system are purely mechanical or that the system can always be identified with an idealized mechanism. However, since we want to arrive at a formal definition of a computational system, we cannot limit ourselves to this intuitive characterization. Rather, we must try to put it in a precise form.

Since I have informally characterized a computational system as a cascade that can be effectively described, let us ask first what a *description* of a cascade is. If we take a structuralist viewpoint, this question has a precise answer. A description (or a representation) of a cascade consists of a second cascade *isomorphic* to it where, by definition, a cascade $MDS_1 = <T, M_1, \{h^t\}>$ is isomorphic to a given cascade $MDS = <T, M, \{g^t\}>$ just in case there is a bijection $f: M \rightarrow M_1$ such that, for any $t \in T$ and any $x \in M, f(g^t(x)) = h^t(f(x))$.

In the second place, let us ask what an *effective* description of a cascade is. Since I have identified a description of a cascade $MDS = <T, M, \{g^t\}>$ with a second cascade $MDS_1 = <T, M_1, \{h^t\}>$ isomorphic to MDS, an effective description of MDS will be an *effective cascade* MDS_1 isomorphic to MDS. The problem thus reduces to an analysis of the concept of an effective cascade. Now, it is natural to analyze this concept in terms of two conditions: *(a)* there is an effective procedure for recognizing the states of the system or, in other words, the state space M_1 is a *decidable* set; *(b)* each state transition function h^t is effective or *computable*. As it is well known, these two conditions can be made precise in several ways which turn out to be equivalent. The one I prefer is by means of the concept of Turing computability. If we choose this approach, we will then require that an effective cascade satisfy: *(a')* the state space M_1 is a subset of the set $P(A)$ of all finite strings built out of some finite alphabet A, and there is a Turing machine that decides whether an arbitrary finite string is member of M_1; *(b')* for any state transition function h^t, there is a Turing machine that computes h^t.

Finally, we are in the position to formally define a computational system. The following definition expresses in a precise way the informal characterization of a computational system as a cascade that can be effectively described.

Definition (computational system)
MDS is a computational system iff $MDS = <T, M, \{g^t\}>$ is a cascade, and there is a second cascade $MDS_1 = <T, M_1, \{h^t\}>$ such that MDS_1 is isomorphic to MDS and

(1) if $P(A)$ is the set of all finite strings built out of some finite alphabet A, $M_1 \subseteq P(A)$ and there is a Turing machine that decides whether an arbitrary finite string is member of M_1;

(2) for any $t \in T$, there is a Turing machine that computes h^t

It is tedious but not difficult to show that all systems that have been actually studied by standard computation theory (Turing machines, register machines, monogenic production systems, cellular automata, *etc.*) satisfy the definition (Giunti, 1992, 1996).

Two Sufficient Conditions for a System not to Be Computational

The definition of a computational system allows us to deduce two sufficient conditions for a mathematical dynamical system not to be computational. Namely, a mathematical dynamical system $MDS = <T, M, \{g^t\}>$ is not computational if it is continuous in either time or state space or, more precisely, if either *(i)* its time set T is the set of the (nonnegative) real numbers, or *(ii)* its state space M is not denumerable.

An immediate consequence of condition *(ii)* is that *any finite neural network whose units have continuous activation levels is not a computational system.* Also note that *the same conclusion holds for any continuous system specified by differential (or difference) equations.* Since all these systems are continuous (in time or state space), none of them is computational.

Summing up the Argument

We have thus seen that *(I)* all models currently employed in cognitive science are mathematical dynamical systems; *(II)* a standard computation can only be performed by a computational system; *(III)* any finite neural network whose units have continuous activation levels or, more generally, any continuous system specified by differential (or difference) equations is not a computational system. Hence, all connectionist models in this class and all nonconnectionist continuous models cannot perform standard computations. But then, if our scientific explanations of cognition are based on these models, we cannot maintain that cognition is, essentially, a matter of the standard computations performed by the cognitive system which these models are intended to describe. On the other hand, it is obvious that *(IV)* all symbolic models are computational systems. Therefore, computationalism is only consistent with symbolic modeling or, more generally, with any other approach which employs computational systems as models of cognition.

A word of caution is needed here. Somebody might object to this conclusion in the following way. It is well known that the behavior of virtually all continuous systems considered by physics can be simulated, to an arbitrary degree of precision, by a computational system, even though these systems are not computational systems themselves (Kreisel, 1974). Why should the continuous systems considered in cognitive science be different in this respect? As long as the behavior of a continuous model of a cognitive system can be simulated (to an arbitrary degree of precision) by a computational system, there is nothing, in the model, which is beyond the reach of standard computational methods. Therefore, it is false that computationalism is only consistent with computational modeling.

This objection is confused because it blurs the distinction between the standard computations *performed* by a system, and the *simulation* of its behavior by means of standard computations performed by a different system. In the first place, this distinction is essential for the formulation of the computational hypothesis itself. If computationalism is intended as a very general hypothesis that indicates the appropriate style of explanation of cognitive phenomena (namely, a computational style), it is crucial to affirm that cognition depends on the standard computations *performed* by the cognitive system we are studying, for it is precisely by understanding the particular nature of these computations that we can produce a detailed explanation of cognition. But then, in formulating the computational hypothesis, we are in fact implicitly assuming that the cognitive system *is* a computational system, we are not just claiming that its behavior can be simulated by a computational system. In the second place, I have argued that any continuous model is not a computational system, and thus it *cannot perform* standard computations. But then, if our scientific explanations of cognition are based on continuous models, we cannot maintain that cognition is, essentially, a matter of the standard computations performed by the cognitive system which these models are intended to describe. Therefore, computationalism is indeed inconsistent with continuous modeling.

Concluding Remarks

My argument shows that, unless we construe computationalism by explicitly referring to some nonstandard notion of computation, we cannot maintain that computationalism is a basic hypothesis shared by *all* current research in cognitive science. In view of this fact, however, we should consider at least two further questions.

First, what kind of nonstandard notion of computation would be needed for an adequate generalization of the computational hypothesis? And, second, is there some other hypothesis that might play this unifying role as well?

As regards the first question, I will limit myself to just one preliminary remark, for a critical discussion is beyond the scope of this paper. Even within these limits, however, it seems quite reasonable to maintain that a generalized version of the computational hypothesis should be based on a theory of computation that *(i)* applies to continuous systems and standard computational systems as well; *(ii)* in the special case of standard computational systems, this more general theory reduces to the standard one, and thus *(iii)* all the standard computability results should turn out to be special cases of the more general theory. I leave it up for further discussion whether these conditions are indeed well chosen, or whether they are in fact satisfied by some theories which intend to generalize various aspects of standard computation theory (Blum, Shub, and Smale, 1989; Friedman, 1971; Shepherdson, 1975, 1985, 1988; Montague, 1962).

As for the second question, we have seen that all models currently employed in cognitive science are mathematical dynamical systems. Furthermore, in general, a mathematical dynamical system changes its behavior according to the particular state evolution that the system undergoes. But then, if our aim is to model cognition by means of appropriate mathematical dynamical systems, we may very well claim that *cognition is, essentially, a matter of the particular state evolutions that a cognitive system undergoes in certain situations*. I call this hypothesis *dynamicism*. For two, quite different, articulations and defenses of dynamicism see van Gelder and Port (1995) and Giunti (1995, 1996).

It is thus clear that dynamicism, unlike (standard) computationalism, is consistent with symbolic, connectionist, and nonconnectionist continuous modeling as well. Therefore, all research on cognition might end up sharing this new hypothesis, independently of the type of model employed. The question remains, however, whether this possibility will really obtain. I believe that the answer to this question depends on whether the explicit assumption of a dynamical perspective can sharply enhance our understanding of cognition. This issue, however, will ultimately be settled by detailed empirical investigation, not by abstract argument.

On the other hand, it is also quite obvious that the dynamical hypothesis, as stated above, only gives us an extremely general methodological indication. Essentially, it only tells us that cognition can be explained by focusing on the class of the dynamical models of a cognitive system, where a dynamical model is *any* mathematical dynamical system that describes some cognitive aspect of the cognitive system. Now, a standard objection against this version of dynamicism is that this methodological indication is so general as to be virtually empty. Unfortunately, a detailed rebuttal to this charge goes beyond the scope of this paper. Therefore, I must limit myself to briefly outline the three defenses that have been adopted by the proponents of the dynamical approach.

The first line of defense points out that dynamicism, just like computationalism, has in fact two aspects. The first one is the specification of a particular class of models (dynamical *vs.* computational models), while the second is the proposal of a conceptual framework (dynamical systems theory *vs.* computation theory) that should be used in the study of these models. Therefore, if we also consider this second aspect, we see that the mathematical tools of dynamical systems theory provide dynamicism with a rich methodological content, which clearly distinguishes this approach from the computational one (Giunti 1995, 1996; van Gelder and Port 1995; van Gelder 1995).

Second, some proponents of the dynamical approach (van Gelder and Port 1995; van Gelder 1995) have in fact restricted the class of models allowed by the dynamical hypothesis. According to their proposal, dynamical models include most connectionist models and all nonconnectionist continuous models, but they exclude computational models. Thus, under this interpretation of dynamicism, it is no longer true that the dynamical hypothesis is consistent with symbolic modeling. These authors, however, do not take this to be a drawback, for they maintain that all symbolic models give a grossly distorted picture of real cognition.

Finally, my line of defense (Giunti 1995, 1996) also restricts the class of the dynamical models, but in a different way. The heart of my proposal lies in the distinction between two different kinds of dynamical models: simulation models and Galilean ones. This distinction is an attempt to set apart two, quite different, modeling practices. *Simulation models* are mathematical dynamical systems which, to a certain extent, are able to reproduce available data about certain tasks or domains. Besides this empirical adequacy (which sometimes is itself quite weak) it is very difficult, if not impossible, to find an interpretation which assigns a feature (aspect, property) of the real system to each component of the model. By contrast, *Galilean models* are built in such a way that no component of the model is arbitrary. Rather, each component must correspond to a magnitude of the real system. Galilean modeling is in principle consistent with symbolic, connectionist, and nonconnectionist continuous modeling as well. What I have been arguing for is that we should take the *ideal* of Galilean modeling more seriously for, if we are successful, we are going to build a better science of cognition.

References

Blum, L., Shub, M., and Smale, S. (1989). On a theory of computation and complexity over the real numbers: NP-Completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, *21*, *1*, 1-46.

Friedman, H. (1971). Algorithmic procedures, generalized Turing algorithms, and elementary recursion theory. In R. O. Gandy and C. M. E. Yates (Eds.), *Logic colloqium '69* (pp. 361-389). Amsterdam: North Holland.

Giunti, M. (1992). *Computers, dynamical systems, phenomena, and the mind.* Doctoral dissertation. Bloomington, IN: Indiana University, Dept. of History and Philosophy of Science.

Giunti, M. (1995). Dynamical models of cognition. In R. F. Port and T. van Gelder (Eds.), *Mind as motion* (pp. 549-571). Cambridge MA: The MIT Press.

Giunti, M. (1996). *Computation, dynamics, and cognition*. New York: Oxford Univ. Press. Forthcoming.

Johnson-Laird, P. N. (1988). *The computer and the mind.* Cambridge MA: Harvard Univ. Press.

Kreisel, G. (1974). A notion of mechanistic theory.*Synthese*, *29*, 11-26.

Montague, R. (1962). Toward a general theory of computability. In B. Kazemier and D. Vuysje (Eds.), *Logic and language*. Dordrecht: D. Reidel.

Newell, A. (1980). Physical symbol systems. *Cognitive Science*, *4*, 135-183.

Newell, A., and Simon, H. (1972). *Human problem solving*. Englewood Cliffs NJ: Prentice Hall.

Port, R. F., and T. van Gelder (Eds.) (1995). *Mind asmotion: Explorations in the dynamics of cognition*. Cambridge MA: The MIT Press.

Pylyshyn, Z. W. (1984). *Computation and cognition*.Cambridge MA: The Mit Press.

Rumelhart, D. E., and McClelland, J. L. (Eds.) (1986).*Parallel distributed processing*. 2 vols. Cambridge MA: The MIT Press.

Shepherdson, J. C. (1975). Computation over abstract structures: serial and parallel procedures and Friedman's effective definitional schemes. In H. E. Rose and J. C. Shepherdson (Eds.), *Logic colloquium '73* (pp. 445-513). Amsterdam: North Holland.

Shepherdson, J. C. (1985). Algorithmic procedures, generalized Turing algorithms, and elementary recursion theory. In L.A. Harrington, *et al.* (Eds.), *Harvey Friedman's research on the foundations of mathematics* (pp. 285-308). Amsterdam: North Holland.

Shepherdson, J. C. (1988). Mechanisms for computing over abstract structures. In R. Herken (Ed.), *The universal Turing machine: A half century survey* (pp.581-601). Oxford: Oxford Univ. Press.

van Gelder, T. (1995). Connectionism, dynamics, and the philosophy of mind. To appear in the *Proceedings volume of Philosophy and the sciences of mind. The third Pittsburgh-Konstanz colloquium in the philosophy of science. Konstanz, May 1995.*

van Gelder, T., and Port, R. F. (1995). It's about time: an overview of the dynamical approach to cognition. In R. F. Port and T. van Gelder (Eds.), *Mind as motion* (pp. 1-43). Cambridge MA: The MIT Press.