Why Everything Doesn't Realize Every Computation

RONALD L. CHRISLEY
School of Cognitive & Computing Sciences, University of Sussex, United Kingdom
(ronc@cogs.susx.ac.uk)

Abstract. Some have suggested that there is no fact to the matter as to whether or not a particular physical system realizes a particular computational description. This suggestion has been taken to imply that computational states are not "real", and cannot, for example, provide a foundation for the cognitive sciences. In particular, Putnam has argued that every ordinary open physical system realizes every abstract finite automaton, implying that the fact that a particular computational characterization applies to a physical system does not tell one *anything* about the nature of that system. Putnam's argument is scrutinized, and found inadequate because, among other things, it employs a notion of causation that is too weak. I argue that if one's view of computation involves embeddedness (inputs and outputs) and full causality, one can avoid the universal realizability results. Therefore, the fact that a particular system realizes a particular automaton is not a vacuous one, and is often explanatory. Furthermore, I claim that computation would not necessarily be an explanatorily vacuous notion even if it were universally realizable.

Key words. Computation, philosophy of computation, embeddedness, foundations of cognitive science, formality, multiple realization.

1. Introduction

A specific worry about our current understanding of computation arises out of the observation that our formal notions of computation, such as those expressed in the formalisms of Turing Machines and recursive function theory, seem so abstract as to deem computational any physically realizable system. The worry focuses on the lack of utility of a concept of computation that is as universally applicable as physical realization. If *any* physical system can be characterized as computational, how can it be interesting that a *particular* system is computational? How can the fact that that system is computational be explanatory? In particular, how can the notion of computation be used to explain cognition, to distinguish thinking beings from mere inert matter? It seems we need a more restricted notion of computation.

Both Putnam (Putnam 1988, pp. 95–96; 121–125) and Searle (Searle 1990; 1992, ch. 9) have presented arguments for the claim that computational states are universally realizable, in the sense that we could interpret any physical system as instantiating any computational characterization. They both argue that this has dire consequences for the computational view of the brain and mind that is a working hypothesis in cognitive science. For example, Searle puts it this way: just as one can argue (via the Chinese Room argument) that semantics is not intrinsic to syntax, so also can one argue that syntax itself is not even intrinsic to physics

Minds and Machines **4**: 403–420, 1995.

(Searle 1992, p. 210). But whereas Searle admits (Searle 1992, p. 209) that the threat of universal realizability could be avoided if our notion of computation is modified to include causal and counterfactual notions (implying that these are lacking at present), Putnam thinks that the universality, and hence vacuity, of the notion of computation remains, even if one requires computational state transitions to be causal.

In the following, I analyze Putnam's argument and find it inadequate, because, *inter alia*, it employs a notion of causation that is too weak. Therefore, the fact that a particular system realizes a particular automaton is not a vacuous one, and is often explanatory. But also I claim that computation would not necessarily be an explanatorily vacuous notion even if it were universally realizable. Thus, claims such as "the brain is a computer executing program P," are not meaningless or incoherent, as Putnam would have us believe.

Before turning to Putnam, further consideration of Searle's position is required. Despite what I said two paragraphs before, I do not mean to suggest that Searle thinks all is rosy about the ontological status of computational states. He says (Searle 1992, p. 209) that perhaps one can't interpret any physical system to be any computer, but that doesn't matter, since the *real* problem with computation is that it involves a notion of interpretation in the first place. This makes computation observer-relative, and therefore unsuitable as a foundation for cognitive science. I think there are two ways in which Searle thinks that even a causal notion of computation is observer-relative, but I think neither should worry anyone who wishes to found an understanding of the mind on computation:

(1) First, there is an objection to (even a causal notion of) computation that arose in personal discussions that I have had with Searle (but of course, he is not *committed* to the views I ascribe to him here). I believe Searle would consent to the following: if one adopts a causal notion of computation, then every system will *not* realize every computation, but every system *will* realize multiple (perhaps infinitely many) computations simultaneously.

I agree with that, for pretty much the same reasons Chalmers does (see Chalmers, this issue). So far so good. The disagreement between Searle and me comes next: he thinks that this realization of a multitude of computational descriptions is still a problem for a computational foundation for cognitive science. Why? Presumably because he thinks cognitive science requires that there be a unique computational description for a system that is to be explained. And to single out a particular computational characterization in such a way is to make cognitive science observer-relative: one could have been just as justified in choosing a *different* computational characterization for the *same* system.

But cognitive science doesn't require that there be a unique computational description for a system. Consider a cognitive science that uses computation in the following reductive sense: mental states *are* computational states. On this view, there are a host of laws of the form: anything in computational state C (individuated by appealing to a computational description) is thereby in mental

state *M*. (I suspect that *identity* is too strong to be the right relation between computational and mental states, but if Searle's objection fails for even this extreme form of computationalism, it will *a fortiori* for weaker positions.) Presumably, Searle's thought is this: since there are multiple computational characterizations of a system, it will follow that the antecedents of more than one of these laws will be satisfied, and therefore there will be some indeterminacy as to which of the several mental states mentioned in the consequents of the activated laws is the *real* mental state of the system. This indeterminacy can only be resolved by arbitrarily choosing to employ one computational description over the others. Thus, mental states would be unacceptably observer-relative.

Some might not think that this result would be objectionable; but I share Searle's desire to avoid such observer-relativism of the mental. Fortunately, such indeterminacy doesn't follow from the fact that any system realizes a multitude of computational descriptions. It does not follow for at least two reasons:

- Clearly, not *every* computational state will appear in the left hand side of one of these laws; as Chalmers (this issue) points out, every physical system can be correctly characterized as the one state finite automaton, but nothing should have any mental states in virtue of realizing *that* computational description. In fact, it might be that out of all the computational descriptions that a given system realizes, *only one* will appear in the antecedent of a computational/psychological bridging law; or, it might be that all the computational descriptions appear on the left hand side of the *same* law. In such cases, there would be no multiple assignment of mental states, no indeterminacy, and thus no observer-relativity.
- Even if more than one of the computational descriptions appears on the left hand side of a bridge law, and even if they appear in *different* laws, the multiple mental states so assigned might not be incompatible, either because the multiple mental states are hierarchically related (e.g. I'm happy, and I'm happy that today is Friday; no indeterminism there) or because the mental states just simply *can* be possessed at the same time (e.g. I'm happy that today is Friday, and I believe that it's raining).

(2) The second reason why one might think that computation is observer-relative, the one Searle gives in his book, is this:

> We can't, on the one hand, say that anything is a digital computer if we can assign a syntax to it, and then suppose that there is a factual question intrinsic to its physical operation whether or not a natural system such as the brain is a digital computer. (Searle 1992, pp. 209–210.)

This brings us to issues of realism and instrumentalism in science that are too large to be addressed in this digression, but I have a quick reply. That an object is *interpreted* by someone as being *C* is a deeply observer-relative fact; that an object is *interpretable* by someone as being *C* need not be observer-relative, if enough constraints are put on the conditions of interpretation. Whether or not a particular phenomenon is interpretable by us in a certain way does not just depend on *us*; it also depends on the phenomenon. Many, many things can be

interpreted by us as being, say, a particular 25-state Turing Machine. But *vastly* many more will not be so interpretable. That suggests that there is something that those interpretable things have in common, something objective, even though that objective commonality happens to have a convenient expression in terms of our abilities to interpret.

Furthermore, on the broad notion of "observer-relative" that Searle's discussion requires, don't our *other* scientific physical properties (e.g., biological ones) involve, at root, some notion of interpretation? If *they* are observer-relative, then what's so wrong with being observer-relative?

2. Putnam's Argument for the Universal Realizability of Finite Automata

Putnam has provided a meticulous and concrete expression of the claim that computation is so abstract as to be vacuous. His "theorem", if its complex derivation is sound, establishes that "every ordinary open system is a realization of every abstract finite automaton." In order to establish his conclusion, Putnam appeals to two physical principles:

> *The Principle of Continuity*. The electromagnetic and gravitational fields are continuous, except possibly at a finite or denumerably infinite set of points. (Since we assume that the only sources of fields are particles and that there are singularities only at point particles, this has the status of a physical law.)
> *The Principle of Noncyclical Behavior*. The system S is in different maximal states at different times. This principle will hold true of all systems that can "see" (are not shielded from electromagnetic and gravitational signals from) a clock. Since there are natural clocks from which no ordinary open system is shielded, all such systems satisfy this principle. (N.B.: It is not assumed that *this* principle has the status of a physical law; it is simply assumed that it is in fact true of all ordinary macroscopic open systems.) (Putnam 1988, p. 121.)

The Principle of Continuity claims that the electrical and gravitational fields are continuous; the Principle of Noncyclical Behavior states that every system is in different states at different times. The first principle I will not dispute, other than to point out that as Putnam admits (Putnam 1988, p. 121), and as one anonymous reviewer points out, the Principle of Continuity appears to assume classical, as opposed to quantum, physics. The impact of this assumption on the success of Putnam's argument I leave to those who can speak on such matters with authority.

The second principle, however, is more problematic, as is the way that Putnam attempts to employ it. Briefly, the only way Putnam can guarantee the truth of the second principle is for him to individuate states by their absolute position in time; but this prevents him from using the principle in the way he intends: to demarcate states that are causally related in such a way as to realize a particular finite automaton (cf. Section 5, below).

Putnam's argument proceeds as follows. He sees it as sufficient to show how any physical system can realize some arbitrary finite automaton, such as one that goes through "the following sequence of states in the interval (in terms of

'machine time') that we wish to simulate in real time: *ABABABA*" (Putnam 1988, p. 122). The goal is to come up with a definition, in terms of the physical properties of an arbitrary system S, of the states A and B such that the system goes through the sequence of states *ABABABA* in a particular time interval. Let $t_1, t_2, \ldots, t_7$ be the times corresponding to the beginning of each of these automata states, with t_8 being the time of the end of the last state. Let s_i be the region of physical state space that S occupies between t_i and t_{i+1}. The definitions for A and B in this particular case (and therefore, in principle, in general) are easy to state: $A = s_1$ OR s_3 OR s_5 OR s_7 (i.e., the system is in computational state A if its physical state lies in any of the parts of state space denoted by s_1, s_3, s_5, and s_7); $B = s_2$ OR s_4 OR s_6. This will entail that S is in states A and B at the right times to result in the sequence *ABABABA* for the temporal interval in question.

We can see immediately an example of Putnam's need to appeal to his physical principles. Without the Principle of Noncyclical Behavior, one cannot assume that the s_i will be disjoint, and if that is so, then some of the conditions sufficient for A might turn out to be sufficient for B. For example, if s_2 were not disjoint from s_3, then there would be at least one point in state space that is in both s_2 and s_3, implying that when the system was in that physical state, it would also be in both computational states A and B. This would yield an ambiguous interpretation function from the physical states of S to the computational states of S, whereas automata states are exclusive.[1]

So the stakes for the s_i being disjoint are high. If they are not, Putnam can't ensure that he will always be able to construct a proper, non-ambiguous interpretation function from physical to computational states. That's where the Principle of Noncyclical Behavior comes in: the disjointness of the s_i follows directly from the purportedly noncyclical behavior of S. If S never makes transitions to states in which it has been previously, then there is no way that the temporally disjoint s_i (which are just time-slices of S) could fail to be disjoint in state space. Thus the stakes are moved from the disjointness claim to the second principle which supports it. But, as I will argue below (in Section 5), Putnam gives us no good reason to believe that systems can never be in the same state twice.

3. Is Computation Essentially Causal?

Ignoring, for now, the problems with the disjointness of the s_i, the only thing then left for Putnam to show is that the sequence of state transitions is causal; that the fact that the system is in state A (and receives the input that it does at that time; this is discussed in Section 6 below) *causes* the system to go into state B (and emit the outputs that it does). Putnam has to show that his arbitrary computational interpretations of a state are causal; otherwise (as Searle admits) one could

prevent universality by only considering the causal characterizations to be the ones that are truly computational.

Some might deny that causal connectedness is an essential property of computational states. Turing Machines themselves, after all, are completely formal; they are abstractions, and are therefore not the kinds of things that can have internal causal structure. However: even if the formal abstractions themselves are not causal, it is a mistake to think that there can be no causal requirements which a physical system must meet in order to be a realization of a formal abstraction. The very fact that they are called Turing *Machines* suggests that the transitions between the realizing states must be mechanizable, or at least causal.

Furthermore, consider an animated display of a Turing Machine on a computer screen. Since, *ex hypothesi*, there is a one-to-one correspondence between the states of the display screen and the states of some Turing Machine, Searle and Putnam would apparently claim that the screen realizes the Turing Machine, if anything does. But it seems clear that we would say that the screen *depicts* a Turing Machine, but is not itself one. One reason why we would deny it computational status is because the state of the screen that corresponds, in the putative interpretation function, to a computational state A does not produce, as a causal effect, the screen state that corresponds to the successor computational state B, even though the Turing Machine depicted does make a transition from state A to state B. Computational states must be able to *cause* other computational states to come about.[2]

But those arguments only establish that we do, in fact, take causation to be essential to computation. But why *should* we, other than to avoid the universal realizability results? One reason seems to be this: computational characterizations are not purely descriptive; they are also explanatory and predictive. In virtue of characterizing something computationally, we not only describe its past, but predict its future and explain both. The fact that our notion of computation puts some constraints on the intrinsic, causal properties of the physical systems which realize that computation allows us to use a computational characterization in order to *predict* the behavior of that system. If there were no connection between our computational notions and causation, then we would have no reason to expect a physical system to continue to be interpretable (with a fixed interpretation function) as realizing a particular computation. Of course, one could, in an ad hoc manner, continually modify the interpretation function from physical states to computational states, so as to guarantee that the system will continue to realize a particular computation. This is, in fact, what Putnam suggests we do. But this method, unlike a truly causal understanding of computation, would not allow us to *predict* which intrinsic physical states a system will go through in the future. We can logically guarantee that any physical system will enter the computational state A in the future only by giving up all claims as to the intrinsic nature of the realization of A, and thus giving up all predictions of the behavior of the system based on it being in A.

4. The Causal Efficacy of Computational States

As said before, Putnam accepts that he must establish a causal connection between his constructed computational states. He argues that S being in A and having the boundary conditions that it does when it is in A *causes* S to go into state B. His argument uses the following lemma:

LEMMA. If we form a system S' with the same spatial boundaries as S by stipulating that the conditions *inside* the boundary are to be the conditions that obtained inside S at time t while the conditions on the boundary are to be the ones that obtained on the boundary of S at time t', where t is not equal to t' [note that this will be possible only if the spatial boundary assigned to the system S is the same at t and t'], then the resulting system will violate the Principle of Continuity. (Putnam 1988, p. 121.)

The argument for causal connectedness then proceeds by claiming that given the state of the boundary of S at time t, then, by the lemma and the Principle of Continuity, the inside of S *must* change from the state it was in just before t to a state distinct from any other state it occupies in the time interval under consideration. Thus, the transitions between states are causal.

I think that Putnam's argument for the causal connectedness of his constructed computational states is unconvincing for several reasons:

(1) It relies on the Principle of Continuity;

(2) It relies upon the lemma, which, as I will argue in Section 5, lacks justification, for the same reasons as does the Principle of Noncyclical Behavior and therefore his argument for the disjointness of the s_i;

(3) It manages to establish causal links between the states of arbitrary physical systems only by assuming a very weak notion of causation.

Since I've already expressed some doubts concerning Putnam's continuity assumptions (1), and the lemma (2) is discussed in Section 5, below, we can move on to Putnam's notion of causation (3).

The question is: under what construal of causation will the "connect-the-dots"-style computational descriptions that Putnam constructs entail, in general, causal relations between computational states? Putnam tells us: it is the notion of causation "that commonly obtains in mathematical physics" (Putnam 1988, p. 96). By this, Putnam means a notion of causation that is quite weak:

> In certain respects the notion of causal connection used in mathematical physics is less reasonable than the common sense notion . . . If, for example, under the given boundary conditions, a system has two possible trajectories – one in which Smith drops a stone on a glass and his face twitches at the same moment, and one in which he does not drop the stone and his face does not twitch – then "Mathematically Omniscient Jones" can predict, from just the boundary conditions and the law of the system, that if Smith (the glass breaker) twitches at time t_0, then the glass breaks at time t_1; and this relation is not distinguished, in the formalism that physicists use to represent dynamic processes, from the relation between Smith's dropping the stone at t_0 and the glass breaking at t_1 (Putnam 1988, p. 97).

This is a weak notion of causation in that the conditions, under this notion, that have to be met in order for two events to be causally related, are weaker than the conditions for our common sense notion. For example, our common sense understanding of causation would not deem Smith's twitching and the glass breaking as causally related, while Putnam's understanding would.

In order to support this notion of causation, Putnam attempts to discredit what he considers to be the main alternative: a notion of causation based on possible worlds and counter-factual conditionals:

> . . . one can sum this up as follows: when we consider what would have been the case if Smith had not twitched, we keep such things fixed as that he released the stone. This means that . . . we consider situations in which the boundary conditions themselves (or the initial conditions, or both) are quite other than they actually are (Putnam 1988, p. 97).[3]

Putnam's objection is that any account of causation in terms of counter-factual conditionals is dependent on a prior notion of what range of possible worlds, for each A and B, are to be used for the determination of whether A caused B. And the idea of a similarity metric on possible worlds is in at least as bad shape as the notion of computation which it is supposed to explicate. Putnam also claims that the notion of "possible world" itself is in dire need of explication. But if this is so, it undermines his own favoured theory of causation as well, since that theory appeals to the "possible trajectories" of a system. The difference between Putnam's notion and the counter-factual notion of causation is not that only the latter uses a notion of possibilities; it is that only the latter uses a similarity metric to determine *which* possibilities are to be considered. Putnam's notion, supposedly, considers all possibilities equally.

This is not the proper place for a detailed enquiry into the advantages and disadvantages of a possible worlds approach to causation, but a more general point can be made: at most Putnam has only showed that one's account of computation will be as universally realizable as one's account of causation. *If one sees causation everywhere, then one will see computation everywhere.* If, however, one prefers to work with a notion of causation that is more restricted, that conforms more to our common sense notion of causation (even though a full account of such a notion may be a long time in the coming), then one will be able to make sense of the idea that some physical systems instantiate a particular computational system, and some do not. I think there are good reasons for favoring, in science, a distinction between two contiguous events that are related causally (the dropping of the stone and the glass breaking), and two contiguous events whose continuity is merely a matter of coincidence (the twitching and the glass breaking). This is precisely what causation is meant to do; a notion which doesn't do this (such as Putnam's) isn't really a notion of causation at all.[4]

5. Complexity Requirements for Computational Interpretation

Searle seems to be aware of the fact that the physics of a system *do* constrain the possible computational ascriptions to that system when he mentions that a system

must be "sufficiently complex" in order to be understood as instantiating a particular computation (Searle 1992, pp. 208–09). Putnam also realizes this; for example, he would admit that a system cannot be assigned computational state A at t_1 and B at t_2 if its physical state at t_1 is indistinguishable, in terms of its intrinsic properties, from its physical state at t_2. It's just that Putnam believes that every *ordinary* open physical system is, in fact, arbitrarily complex (i.e., can be individuated into the number of distinct states necessary to instantiate any automaton).[5]

The last reason, then, for rejecting Putnam's argument for the causal relatedness of his constructed computational states, and for rejecting his Principle of Noncyclical Behavior, centers on his claims concerning the arbitrary complexity of physical states. Specifically, the problem is the lemma mentioned before: if a system were to have the boundary of S from one time and the interior of S from a different time, it would violate the Principle of Continuity. The problems arise in his unconvincing proof:

> *Proof* (of the lemma): Every ordinary open system is exposed to signals from many clocks C (say, from the solar system or from things which contain atoms undergoing radioactive decay, or from the system itself if it contains such radioactive material – in which latter case the system S itself coincides with the clock C. In fact, according to physics, there are signals from C from which it is not possible to shield S (for example, gravitational signals). These signals from C may be thought of, without loss of generality, as forming an "image" of C on the surface of S. For the same reason, there are also "images" of C *inside* the boundary of S. The "image" of C at, say, $t' = 12$ may be thought of as showing a "hand at the 12 position"; while the "image" of C at, say, $t = 11$ shows a "hand at the 11 position." Thus, for these values of t and t', the system S' would have a "12 image" on its boundary and an "11 image" at an arbitrary small distance inside its boundary; but this is to say that the fields which constitute the "images" would have a discontinuity along an entire continuous area, and hence at nondenumerably many points (Putnam 1988, pp. 121–22).

Why is this not convincing? Because Putnam assumes, without justification, that the "images" on the boundary and interior of S are characteristic of the current time of the clock that generates the images. And he assumes that they are characteristic in a strong sense: the images of the signals that bombard S are dissimilar to such an extent that a system with a boundary image of t and an interior image of any t' distinct from, but arbitrarily close to, t would violate the Principle of Continuity.

Putnam obviously does not intend to use a temporally relational individuation of physical states. If he did, then he wouldn't have had to bring in the empirically questionable Principle of Noncyclical Behavior in order to *argue* that systems are in different states at different times; he could have just *stipulated* this. He must, therefore, be using a relatively intrinsic individuation of physical states. In order

for the argument for the lemma to make any sense, then, it must be that one of the following is what Putnam imagines to be the case:

- All systems have "counters" that take as input the gravitational signals, radiation, etc. they receive and increment their count accordingly. This counting ability must be arbitrarily robust: there can be no limitation on how high a system is able to count if Putnam is to be able to make his claims.
- All clock signals explicitly (i.e., in terms of their intrinsic properties) encode their absolute position in time. Thus, systems that are bombarded by them are never in the same state twice, since they have a new input at each instant.

It seems that Putnam must take one of these views in order to claim that the "images" of a particular clock time are characteristic of that time. If they are not characteristic, then it might be that the images corresponding to two different times would be the same, and therefore, his lemma would be shown to be false. That is, no discontinuity would occur if the images of those two times were simultaneously present in the boundary and interior. And if that were the case, then Putnam hasn't shown that the system *must*, even given the boundary conditions, make the state transitions that it does. As a consequence, Putnam could not guarantee that the relations between his constructed computational states are causal, *even on his weak notion of causation.*

So he has to appeal to something like the two ideas just mentioned. But both of these options have problems. As far as the first one goes, one has to ask what physical law prevents a system from being a flip-flop? It seems very likely that there are systems that receive a steady stream of qualitatively identical input from some clock, but merely make a transition from one of two states to the other upon receipt of these signals. How could such a system be interpreted to be realizing any automaton with more than two states, without using some ambiguous interpretation function? We saw before that such a move would be of no use, since computational realists could restrict their notion of computation so as to exclude systems with ambiguous or relational interpretation functions. Some physical systems just don't have the complexity to be interpreted as having such counters.

The second option is suggested as the one that Putnam has in mind when he speaks of "the fields which constitute the images". That is, Putnam takes those parts of the gravitational and electromagnetic fields within the boundaries of a physical system to be parts of that system. It is only by making this assumption that the discontinuity of the images could result in a violation of the Principle of Continuity, since the Principle only concerns the continuity of the gravitational and electromagnetic fields.[6]

But if this is what Putnam is assuming, then it is clear why he thinks any physical system is complex enough to realize any formal automaton. It is because he is assuming that all physical systems are continuous (via the continuity of the fields and the inclusion of the fields into the physical system). This again raises an issue from Section 2: is it wise for Putnam to rest his philosophical points on a

particular physics which ignores the discrete (quantum) nature of physical systems?

However, even if we grant continuity, and the existence of clocks which explicitly encode their time (perhaps the background radiation is an electromagnetic example; I can't imagine what Putnam has in mind for a gravitational equivalent), and the *possibility* of systems whose internal states (including the fields) reflect this temporal encoding, that does not mean that all or even any actual physical systems do, in fact, contain such images. The effects of two different clocks can cancel one another out (consider a physical system midway between two clocks that emit complimentary signals); signals can be disturbed, distorted, blocked; they can decay; qualitatively distinct signals might have identical effects on a system; etc. Surely Putnam doesn't want his argument to depend on issues as empirically contingent and contentious as *these*?

Since it seems that Putnam can't, without further justification, appeal to the lemma, he has given us no good reason to believe that his constructed computational states are *even weakly* causally related; and since Putnam can't appeal to the Principle of Noncyclical Behavior, he can't establish the disjointness of the s_i (cf. Section 2).

There is another way (albeit one that requires much more elaboration than can be given here) that complexity considerations might tell against Putnam's argument. This is based on the insight that, roughly speaking, one's theory of a phenomenon should at least be *less complex* than the phenomenon itself. If it isn't, then the theory is in some sense confabulating, or at least not cutting nature at its joints. Suppose I present you with a steel ball, and claim that it is implementing a particular expert system, say Mycin. You ask me to substantiate this outrageous claim. I proceed to do so, by finding strange, relational, disjunctive, and complex characterizations of the steel ball states to identify with each of Mycin's computational states. This characterization would be so complex, in fact, that a text representation of it might take up, say, one thousand times the computer disk space that the Mycin program itself takes up! Anyway, I go on to claim that with this interpretation of the steel ball states, I can tell you how Mycin would respond to any given query. Even if I could, it would only be because of the complexity of the interpretation function, not the steel ball. The steel ball wouldn't be implementing Mycin, *I* would be. The intuition that this type of story is supposed to motivate is that it is natural to put some restrictions on the relative complexity of our interpretations in order to rule out such cases. Such restrictions would, no doubt, rule out Putnam's interpretations as well.[7]

Finally, one anonymous reviewer points out that computational descriptions do not only specify causal transitions that must take place; they also implicitly *prohibit* many transitions. For example, if an automaton is supposed to move causally from state A to state B, then it is supposed to do this *without moving into state C in the process*. Putnam tries to avoid the difficulties that this observation raises by defining the s_i to be the region containing all of the states of S between t_i

and t_{i+1}. This would rule out the possibility of the S moving from A to B via C, but only if the s_i could be shown to be disjoint. But we have already seen that he cannot show this.

To summarize some of the main points so far, Putnam's argument for the universal realizability of finite automata is uncompelling because:

- The disjunctive nature of its individuation of computational states limits Putnam to *post hoc descriptive* states, yet computational characterizations are also *predictive*;
- Its notion of causation is too liberal, in that it would allow as causally related many events that, in everyday life and sciences other than mathematical physics, we would *not* take to be causally related;
- It relies on the Principle of Noncyclical Behavior and the lemma, which both, in turn, rely on an unconvincing and largely empirical account based on "clocks". Thus, it fails to establish that the transitions are even weakly causal, and fails to establish the disjointness of the realizing states;
- The failure to establish the disjointness of the realizing states yields ambiguous interpretation functions, and prevents Putnam from accounting for the fact that computational characterizations *prohibit* certain state transitions.

6. Computation and the World: Inputs and Outputs

But wait; there's more. Computers don't, in general, *just* sit around making state transitions. They receive signals from keyboards, mice, and video cameras, and control displays, printers, and robot arms. They *do* things; they interact with things. Even formal automata include a notion of input and output. Another problem, then, for Putnam's proof is that, strictly speaking, he only establishes it for the case of automata without any inputs or outputs (Putnam admits as much on p. 124). To try to rectify this, Putnam would have to count the state of the boundary of S at a particular time to be the input to, and output of, the automaton. Let $[A:I_i:O_j:B]$ indicate a finite automaton that when in state A, receives input I_i which causes it both to output O_j, and to move into state B. Putnam must define the instantiation of I_i as the disjunction of all the boundaries of S that correspond to states which receive I_i as input, in the interval being interpreted. For example, consider the finite automaton: $[A:I_1:O_1:B]$ $[B:I_2:O_2:C]$ $[C:I_1:O_1:A]$. If physical state s_1 is interpreted as state A, s_3 is interpreted as C, then I_1 could be defined as: boundary (s_1) OR boundary (s_3). Only then can Putnam argue in a way similar to before, that the computational state of the system and the input received in that state *jointly cause* the system to move into the next state, and emit an output.

One problem with this approach is that it isn't faithful to the notion of input and output that is involved in computation. For computational purposes, inputs and outputs are characterized in terms of their intrinsic properties. If we *define* inputs and outputs in a *post hoc* manner, as whatever boundary state a physical

system has at a particular time, then adding inputs and outputs gives Putnam no (further) difficulties.[8]

But if the definition of an output is fixed *in advance* as, say, the display of a character on a video display, then Putnam will not be able to show that a given system, for example my office wall, instantiates any formal automaton with that kind of output. That is because the state transitions of the wall will not causally determine the output, even, presumably, on Putnam's weak notion of causation. Varying the states of the wall (considering the various possible trajectories of the physical system with respect to its input) will not result in a corresponding variation in video display states. Therefore, the output is not *caused* by the state transitions in question. Similar considerations apply in the case of inputs. So only *post hoc* notions of input and output will allow Putnam to maintain his universal realizability thesis, yet *post hoc* notions are unacceptable for predictive and explanatory purposes. If what counts as a physical realization of an output is not fixed in advance, then we can guarantee that any system will emit a given output in the future, but only at the price of having no idea of how that output will be manifested. We will only have a descriptive, not a predictive computational understanding of the system (cf. the end of Section 3).

In fact, Putnam admits that for any given automaton with inputs and outputs, one will be able to restrict the set of systems that instantiate it (Putnam 1988, p. 124). In some sense, then, he admits defeat: not *every* physical system can instantiate every finite automaton. But he doesn't really consider this concession to be a concession of defeat. That's because he believes that one will still have universal realizability of computation within the class of physical systems that get the input and output right:

> Imagine, however, that an object S which takes strings of "1"s as inputs and prints such strings as outputs behaves from 12:00 to 12:07 exactly as *if* it had a certain description D. That is, S receives a certain string, say "111111" at 12:00 and prints a certain string, say "11" at 12:07, and there "exists" (mathematically speaking) a machine with description D which does this (by being in the appropriate state at each of the specified intervals, say 12:00 to 12:01, 12:01 to 12:02, . . . , and printing or erasing that it is supposed to print or erase when it is in a given state and scanning a given symbol). In this case, S too can be *interpreted* as being in these same logical states A, B, C, . . . at the very same times and following the very same transition rules; that is to say, we can find *physical* states A, B, C, . . . which S possesses at the appropriate times and which stand in the appropriate causal relations to one another and to the inputs and outputs. The method of proof is exactly the same as in the theorem just proved (the unconstrained case). Thus we obtain that *the assumption that something is a "realization" of a given automaton description (possesses a specified "functional organization") is equivalent to the statement that it behaves as if it has that description* (Putnam 1988, p. 124, his emphasis).

Putnam means "behaves" here purely externally: any physical system that, for a given time period, has the same inputs and outputs as a particular finite automaton, instantiates that automaton. Thus, Putnam is claiming that there is no computational difference between the two following systems:

- A program that calculates trajectories for spacecraft on the basis of certain input parameters (position, mass and velocity of the craft and nearby bodies) that

is run, on three successive occasions, on the inputs a, b, c respectively and yields outputs x, y, z respectively;

- A lookup table which only has three entries: $a \rightarrow x$, $b \rightarrow y$, $c \rightarrow z$.

Such an equivalence would be bad enough for our current understanding of computation, but Putnam has even more specific prey in mind. In particular, the reason why he is attempting to undermine computation in general is because he is opposed to its use as a foundation for an understanding of the mental in particular. And if Putnam can show that all behaviorally equivalent systems instantiate the same program, then he will have shown that functionalism implies behaviorism, a conclusion that many who wish to use computation as a foundation for cognition would be loathe to accept.

Of course, the conclusion need not be accepted, since it depends on the central argument of universal realizability, which, as we have seen, doesn't work. Nevertheless, one might think that the computational equivalence of behaviorally identical systems might have held *if* Putnam's original argument were sound. But I don't think even this is correct. Perhaps if one restricts oneself to characterizing a particular temporal interval of a system, then one could get the equivalence of behavior and computation if Putnam's main argument were successful. But this is to make the mistake (again) of seeing computational characterizations as purely descriptive, and not explanatory or predictive (cf. the end of Section 3). Not all systems that have the same inputs and outputs for a short interval will continue to have the same inputs and outputs in the future. Thus, a particular computational characterization will apply, for predictive purposes, only to some small subset of those physical systems.

7. The Worst Case: Universal, but Useful

Input/output issues aside, one might think: OK, so Putnam doesn't show that every system realizes *every* finite automaton. There are, in principle, limits to what can count as an acceptable interpretation. But the fact is that, given the natural complexity of physical stuff out there, there is still a *lot* of room for indeterminacy. Even if every system doesn't instantiate *every* automaton, it might be that every ordinary macroscopic system (like a brain) instantiates an *infinite* number of automata.[9]

As stated in the introduction, such indeterminacy doesn't count against computation. There is a reason why Putnam set his goal to be such a lofty one: it is the only one which can really count against the ontological status of computation. It is only by guaranteeing that every system instantiates *every* computation that one can be sure that no matter what computational account one gives of the brain, it will apply just as well to stones, roads, and walls. If it is admitted that some systems do not instantiate every program, then one will not be able to conclude that everything implements any particular computational characterization of mind that cognitive science puts forward. That is, the modified claim

allows computational characterizations to be non-vacuous, which in turn upholds the coherence of the computational approach to understanding the mind. Which is just what Putnam wishes to reject.

Thus, for computational states to be ontologically sound, one does not have to show that there is only *one*, *unique* computational characterization that applies to a given physical system. In fact, computational practice hinges on just the opposite: that a particular physical system can be understood to be instantiating simultaneously, say, a word-processing program, and a universal Turing Machine. That is, some degree of indeterminacy of computational description is acceptable, or even desirable.

But *what if* computation were universally realizable? What if, barring the just presented arguments to the contrary, any ordinary open physical system could be interpreted as, say, running any program? It is worthwhile to look at just what would follow from what Putnam is trying to establish.[10]

Even if everything is every kind of computer, the brute facts are: (1) we don't actually seek to understand everything in terms of computational properties; and (2) computational explanations, although limited, are actually satisfying in a large number of cases. This just shows that even if computationality is "merely attributed", it can nevertheless be explanatory.[11]

The fact is, it *is* very useful to understand many physical systems (IBM's, Sun workstations, Macintoshes, etc.) in terms of computational properties; and there are many more systems for which such an understanding is *not* useful. If computational properties are universally realizable, this just shows that for some systems, we can always competently assign computational properties in such a way that such assignments will allow us to develop an explanatory and predictive understanding of those systems. If ontology is completely independent of these explanatory concerns, then perhaps claims of the form "physical system x instantiates automaton P" are meaningless in some absolute sense. But if explanatory (or even mere utility) considerations have any say in whether an attribution is warranted or not, then it is clear that sometimes we will be warranted in deeming a system a (particular kind of) computer, sometimes not. The question "is this physical system a digital computer running program P?" will be meaningful, and resolved, at least to some degree, empirically.

8. Formal Computation: Meaningful, but Inadequate

To be fair, characterizing computation in terms of actual inputs and outputs, and in such a way that the actual causal properties of the underlying physical system matter, ventures far beyond the explicit nature of current computational theory, as expressed in, for example, Turing Machines and recursive function theory.[12]

In fact, some may ask: why defend these formal models of computation, when there are many reasons to believe that more embedded, embodied and semantic accounts are required to understand real world computational systems? I agree

that a theory of computation founded solely upon formal notions such as Turing Machines and finite state automata would be an impoverished one. Nevertheless, I think that it would be premature to assume that the success of a mature theory of computation is independent of the status of these purely formal theories.

Accordingly, both Putnam and Searle have done cognitive science a service, by drawing attention to the fact that its uses of the notion of computation may only make sense when accompanied by some implicit assumptions. These assumptions should be made explicit, so that they may be developed and refined. Both Searle and Putnam are in one sense right: a completely formal, non-causal notion of computation is inappropriate for cognitive science. Fortunately, our current understanding, at least implicitly, is more concrete: it is not empty and incoherent (as they claim). Nevertheless, those of us who wish to understand computation, especially those who wish to understand how it relates to cognition, have a substantial and exciting task ahead: that of discovering and articulating these non-formal elements of computation, whether they are, like causation and embeddedness, implicit in our current understanding, or as yet unknown.

Acknowledgements

An early (i.e., inferior) version of this paper, titled "The Ontological Status of Computational States", was read to the University of Sussex Philosophy Society on November 4th, 1990, and will appear in issue 1 of *The European Review of Philosophy* (CSLI Publications, Stanford) in 1994. A later version was presented at G. H. von Wright's Philosophy Research Seminar at the University of Helsinki on April 20th, 1993. This work was made possible by support from the Center for the Study of Language and Information at Stanford University, and by grants from The San Francisco Branch of the English-Speaking Union, The Chancellors of the UK Universities, and the Oxford Overseas Student Support Scheme. Special thanks to John Batali, Matthew Elton, and John Searle for helpful discussions; to Kathy Wilkes for detailed comments on a draft; and to 6 anonymous reviewers for many helpful suggestions.

Notes

[1] Presumably, even those wishing to establish the universal realizability of computation would agree that ambiguous (one-to-many) interpretation functions could not provide an adequate notion of computation. Otherwise, their claim is trivially established: any system realizes any finite automaton because every physical state can be mapped to every computational state, even under the same interpretation. At any rate, those wishing to define computation as non-vacuous merely have to stipulate that computational properties supervene (at least) on physical ones (i.e., if you change the computational state, you must change the physical state somehow) in order to reject this extreme form of universality.

[2] One anonymous reviewer agreed that the screen states are not causally related, but suggested that neither are the bits in screen memory, bits in RAM, or voltages. That is, yes, the screen states are mere depictions of Turing Machine states, but it is depictions *all the way down*. I disagree. There is

some complicated set of CPU, memory, wires, voltages, etc. which causally realize the various Turing Machine states. Otherwise, given, *in advance*, a particular scheme of interpreting physical states as computational states, it would be a miracle, a fluke, that we could reliably get this stuff to simulate a particular Turing Machine.

[3] It is odd that Putnam emphasizes that the possible worlds notion of causation considers "situations in which the boundary conditions are quite other than they actually are." For the mathematical physics notion, too, must vary at least some of the boundary conditions. Otherwise, the only systems that would have different "possible trajectories" would be non-deterministic ones, yet Putnam has stated that he is focusing on the classical (hence, presumably, deterministic) case.

[4] However, those who wish to naturalize intentionality with computation should take heed of a difficulty that Brian Smith has suggested to me in personal discussions. If our account of computation does depend on a notion of similarity of possible worlds, and if the proper account of similarity of possible worlds is itself an intentional one, then it appears that an account of *all* intentionality in computational terms would have to be circular. Perhaps computation can only help naturalize some subset of intentional phenomena?

[5] Therefore, strictly speaking, Putnam is not claiming that computation is *universally* realizable, since there may be some systems that are shielded from every clock. But that alone is not enough to give the computationalist any solace, for reasons similar to those discussed in Section 6, below. For example, anyone who wishes to claim that mental states are computational states would have to admit that not only does a stone have mental states, but it has *all* possible mental states.

[6] My thanks to a participant (Ilkka Kieseppä, I believe) at the G.H. von Wright Research Seminar reading of this paper, who pointed this out to me.

[7] In thinking about the issues raised in the above passage, I benefited from a discussion with Matthew Elton.

[8] But even then one will be in the unsatisfying position of being unable to differentiate inputs from outputs, since they are both defined to be the same boundary state.

[9] Notice that the Cryptographer's Constraint, though useful in other contexts (viz. syntax to semantics, rather than physics to syntax considerations), doesn't help here. The Cryptographer's Constraint (which has been mentioned in related contexts by McCarthy, Dennett, and Harnad) is the observation that as, say, the length of a string of characters increases, the chances that there is more than one meaningful interpretation for that message decreases drastically. The reason why we cannot apply this constraint here (even assuming that we find some syntactic norm to replace the one of "meaningful") is that Putnam is not allowing us (via his continuity assumption) to take as fixed in advance the primitives ("characters") over which the interpretation is being conducted. Consider: if a cryptographer doesn't even known what counts as the characters of a coded message (the prima facie characters? Their orthographical components? The tertiary structure of the molecules of ink?), then the Cryptographer's Constraint does not apply.

[10] To be fair, it should be pointed out, again, that Putnam's main goal in his text was to undermine any computational understanding of mind, and not necessarily anything more. Nevertheless, I sense that Putnam's general scepticism concerning the "reality" of computation is shared by an alarming number of people, many of whom apply it to a broader range of issues. Therefore, the further discussion here is relevant.

[11] In fact, there is a strong current in modern philosophy of science that claims that many, if not all or our explanatory sciences, even (or especially) those as fundamental quantum physics, are based as much on human interests as they are on some ontologically independent reality.

[12] However: (1) some theorists are trying to correct this, as Searle points out (Searle 1992, p. 209; see, e.g., Smith 1991); (2) although embedded, causal computation might be at odds with our current theoretical understanding of computation, it doesn't seem to be that alien to our everyday notion of computation as manifested in computational *practice*.

References

Hilary Putnam (1988), *Representation and Reality*, MIT Press.
John Searle (1990), 'Is the brain a digital computer?', *Proceedings and Addresses of the American*

Philosophical Association 64(3), November 1990. This paper was a Presidential Address delivered at the Annual Pacific Division Meeting of the APA in Los Angeles on March 30th, 1990, and was also delivered at the 5th Annual Computers and Philosophy Conference at Stanford University on August 8th, 1990. A revised version of this paper appeared as Chapter 9 of *The Rediscovery of the Mind*.

John Searle (1992), *The Rediscovery of the Mind*, MIT Press.

Brian Cantwell Smith (1991), 'The owl and the electric encyclopedia', in D. Kirsh, ed., *Foundation of Artificial Intelligence*, MIT Press.