From Scheutz, M. (ed.) (2000) *New Computationalism: Conceptus-Studien* **14**.
Academia Verlag: Sankt Augustin, pp 105-121.

TRANSPARENT COMPUTATIONALISM

RONALD CHRISLEY

Summary. A distinction is made between two senses of the claim "cognition is computation". One sense, the *opaque* reading, takes computation to be whatever is described by our current computational theory and claims that cognition is best understood in terms of that theory. The *transparent* reading, which has its primary allegiance to the phenomenon of computation, rather than to any particular theory of it, is the claim that the best account of cognition will be given by *whatever theory turns out to be the best account of the phenomenon of computation*. The distinction is clarified and defended against charges of circularity and changing the subject. Several well-known objections to computationalism are then reviewed, and for each the question of whether the transparent reading of the computationalist claim can provide a response is considered.

Zusammenfassung. Eine Unterscheidung zweier Interpretationen der Behauptung, daß Kognition Berechnung wäre, wird eingeführt. Die *opake* Leseweise versteht Berechnung im Sinne des Verständnisses der gegenwärtig besten Berechnungstheorie und behauptet, daß Kognition genauso zu verstehen sei. Die transparente Leseweise, welche primär dem Phänomen Berechnung, denn einer bestimmten Berechnungstheorie verpflichtet ist, behauptet daß die beste Erklärung der Kognition durch jene Theorie gegeben wird, die sich als beste Klärung des Phänomens Berechnung herausstellt. Diese Unterscheidung wird im folgenden geklärt und gegen Einwände der Zirkularität bzw. Themenwechsel verteidigt. Etliche wohlbekannte Einwände gegen den Computationalismus werden erneut aufgegriffen um festzustellen, ob die transparente Leseweise des Computationalismus eine Antwort liefern kann.

***

Computationalism in the philosophy of mind is the claim that cognition is computation. Although most of the work in cognitive science and artificial intelligence (AI) has been based on this hypothesis, computationalism has always had its opponents, and the criticisms are becoming more frequent and widespread. To the old critiques (e.g., the Gödel/Lucas, phenomenalist, frame problem and Chinese room objections) have been added objections based on dynamics, the dispensability or clumsiness of representation, externalism, universal realisation, the incoherence of internal representation, and the unreality of computational content, to name a few. While some of these objections can be refuted directly, others are more difficult in that even if one believes them to be ill-founded, nevertheless something seems right about them. How can we continue to hold onto computationalism when it does seem that, e.g., digitality is restrictive, formal symbol manipulation isn't sufficiently world-involving, and Turing machines are universally realisable to the point of vacuity?

The confusion stems, I believe, from an ambiguity in the computational claim itself, at least as I have expressed it in my opening sentence: "cognition is computation".[1] A distinction should be made between two senses of the claim. One sense (call it the *opaque reading*) takes computation to be whatever is described by our current computational theory (via the concepts of Turing machines, recursive functions, algorithms, programs, complexity theory, etc.) and claims that cognition is best understood in terms of that theory. The *transparent reading,* by contrast, has its primary allegiance to the phenomenon of computation, rather than to any particular theory of it. It is the claim that the best account of cognition will be given by *whatever theory turns out to be the best account of the phenomenon of computation*. The opaque reading is a claim about specific current theories, while the transparent claim is a claim about the phenomena of computation and cognition themselves.

Making this distinction allows one to eliminate the confusion posed by some of the criticisms of computationalism. One can agree with the critics that there are aspects of, say, algorithms, which make them unsuitable for understanding every aspect of cognition or mentality. In doing this, one must concede that the opaque reading of computationalism is false, given the central role that algorithms play in our current theory of computation. But one can do that and yet simultaneously (and consistently) maintain the truth of computationalism on its transparent reading, by rejecting the assumption that the best account of computation is along current lines (in this case, as a necessarily algorithmic phenomenon). If the notion of an algorithm, while surely itself computational, need not apply to all cases of computation, then although it is a notion *available to* computationalists for the explanation of cognition, it is not *foisted on* them. If cognition is computation, and yet not all computation is algorithmic, then cognition need not be algorithmic. Likewise for other criticisms of computationalism and other aspects of the current theory of computation.

[1] One ambiguity that is present, but doesn't seem to be the source of the confusion I am addressing, we might call "Clinton's qualifier": that the meaning of the claim "cognition is computation" depends on what the meaning of "is" is. The copula can be taken to indicate various degrees of metaphysical commitment. It might stand for the symmetrical relation "is identical with". Or it could stand for an asymmetrical relation, yielding either 1) cognition reduces to computation, or 2) one can (must?) use computational concepts in order to distinguish cognitive from non-cognitive systems, or 3) cognition is best explained using computational concepts, etc. I hope that the points I make in what follows apply no matter which of these readings one prefers.

On this analysis, the critics have argued against only the opaque reading of computationalism, have only opposed the current, formal notion of computation founded on Turing machines and the like. This is understandable, since the formal view of computation is the de facto orthodoxy, and we are still waiting for a non-formal theoretical alternative. But if it turns out that what makes the artefacts of Silicon Valley tick is not best explained in terms of formal computation, then said critics will have nothing to say against the transparent version of the "cognition is computation" claim.

SEMANTIC GERRYMANDERING?

It must be made clear that adopting the transparent reading of the computationalist claim is not just semantic gerrymandering. Some might suspect that all that is being proposed is a change in the meaning of "computation" (and thus "computationalism") in a post hoc way that saves the "cognition is computation" claim, but only at the cost of either circularity or changing the subject. In this section I want to dispel such suspicions.

To see that the transparent reading is not circular, one can contrast it with a move that *would* be: claiming on the one hand that cognition should be explained using computational concepts, and yet also claiming that computational concepts are whatever concepts give the best account of computational systems, including cognitive systems. If you use cognitive systems to help define what computational concepts are, then the computationalist claim will lose its bite, will be circularly trivial.

But notice that this is not what is being done on the transparent reading. A distinction is made between computers and cognizers: the former are not defined in terms of the latter. Rather, the transparent reading assumes that we have some pre-theoretical, ostensive access to the phenomenon of computation (what PCs do, what iMacs do, etc.); likewise for the phenomenon of cognition (what a person playing chess does, what I do when I try to find the restaurant where we agreed to meet, etc.). The transparent computationalist claim is that whatever concepts give a best account of *this stuff* (gesturing toward the computational phenomena) also give the best account of *that stuff* (gesturing toward the cognitive phenomena).[2]

In order for computationalism to be correct, it doesn't have to be the case that the set of concepts eventually arrived at does justice to *everything* in the initial, pre-theoretic cognitive ostension; nor, for that matter, to everything in the initial, pre-theoretic

[2] Although this distinction between the intuitively computational and theoretical attempts to account for such may now be thought to be radical, it was at the heart of original theoretical thinking in the field. Church's and Turing's theses were that their respective formalisms did justice to a prior, intuitive notion of computation and the computable. It is the fact that these theses bridge the formal/theoretical and the intuitive that makes them unprovable in any formal sense.

computational ostension. The best account (and thus the best theoretically-defined concept) of an ostensively delimited subject matter, it seems to me, is the simplest account which covers as much of the ostension as possible.[3] It might turn out that the best account of computation that we can get rejects as non-computational some things that we pre-theoretically took to be computational (calculators, perhaps), or includes some things that we pre-theoretically took to be non-computational. And it might turn out that some of the things that we pre-theoretically *thought* were cognitive turn out not to be (or things that we *thought* were not cognitive actually are) because the best account of the central cases of cognition implies that they are not (or are).

This idea of letting the "best" account do violence to, or override, our pre-theoretical intuitions might appear to contradict the ethic behind the transparent reading, which I said was to give one's "primary allegiance to the phenomenon of computation, rather than to any particular theory of it". If we discard whatever bits of the pre-theoretic notion of computation (or cognition) don't fit in with our theory, how can we say that our loyalties are with the territory and not the map?

This point is well taken. The approach that is being rejected here is one in which the defining theoretical concepts (e.g., that of Turing machines) are fixed in advance, with the domain of empirical interest then being *whatever aspect of the world is best understood in terms of Turing machines.* But to reject this theoretical dogmatism does not require one to take up its opposite, which is empirical or intuitional dogmatism. To understand that theory is the map, and to understand that it is in tension with experience or intuition does not force one to see the latter as the territory. Instead, one can see pre-theoretical intuitions and experience as just more map, albeit of a kind that is in some sort of epistemological opposition with theory. The territory is neither theory nor intuition, but is constructed out of (or revealed through, depending on your anti-realist/realist leanings) a dialectical interplay between the two. Transparent computationalism, then, asks us to let our theoretical notions of computation be driven by our experience, but also recognises that what we experience as computational will and should change in response to our changing theory of computation.

Here's a sample trace of that dialectic in action, using a perhaps over-familiar, but non-computational example. Our pre-theoretical intuitions were that whales are fish, so they were entered into the pool of phenomena against which we tested theories of fish.

[3] A further constraint might be that the concept should apply to as few phenomena outside the original, ostensively delimited set as possible. Also, the idea is that these constraints should be soft – e.g. an account that scores poorly on its coverage of the intuitive extension, or scores poorly on the simplicity criterion (i.e., is rather complex) might nevertheless be the best account because of its high score on the criterion of excluding phenomena not in the intuitive extension.

Inasmuch as we did that, we were not being theoretically dogmatic—we were letting our intuitions have a say (contrast this with the Scholastic or Rationalist who may have tried to derive the classes of animals and their nature from first principles). But once a proper, successful theory of fish was settled on, it determined the extension of interest, excluding whales, since they don't meet the criteria for fish under the adopted theory (specifically, they don't use gills to extract oxygen from water). Furthermore, whales do meet the criteria for being a mammal: they have hair, are endothermic and produce milk.[4] In that we allow whales to be excluded from the class of fish, we are not being empirically or phenomenally dogmatic; we are allowing for virtuous conceptual change to occur, rather than insisting on the ways of carving up the world that the ancients (or children, or our naive selves) had. Eventually, this theoretically-driven extension may become our intuitive, common-sense way of looking at the domain of fish (and mammals). We may even call it "pre-theoretic", despite the fact that it was historically shaped by theory. This new intuitive notion of fish becomes the tribunal for our theories of fish, putting pressure on those theories to do justice to the phenomenon of fish as now understood. Thus the process iterates.

This parable about fish also applies to computation (with pocket calculators perhaps playing the role of whales). Just as our intuitive notions of fish (and gold, and water, and just about everything else) have driven yet been altered by our theories of those natural kinds, so also shall (and should) our intuitive notions of computation constrain and be constrained by our theories of computation.

With all that in place, it is now a relatively easy matter to respond to the other worry stated at the beginning of this section: Is transparent computationalism post hoc and just changing the subject? In a sense, yes. But in a more important sense, no. It isn't in the sense that scientific progress in general isn't. To reiterate with a less fishy example, it's true that we now mean something different by, e.g., gold than the ancients did: they included just about any gold-coloured metal into that category. But the best way to understand scientific progress is not to see them as being right about their notion of gold, but rather as wrong about gold itself, of which we now have a better understanding (Putnam 1975). It is gold itself that unites ancient theorising with current theorising; if we couldn't unite the two, by saying that we and the ancients were striving for an understanding of the same thing, then we would have no grounds on which to say that

[4] This example suggests, therefore, that what we take to be computational not only depends on our theory of computation, but on our other theories as well. We exclude whales from the class of fish because of the explanatory advantages of seeing them as mammals as much as the explanatory drawbacks of seeing them as fish. So also might we deny computational status to an information-processing device if we have a non-computational theory which provides better explanations of it.

our account was an improvement on theirs. We would have to say that we have just changed the subject. But today's chemists *are* theorising about the same stuff that Archimedes was, even though what they are thinking of *necessarily* has atomic number 79 and Archimedes didn't even have the concept of atomic number. Therefore, too, future accounts of computation may indeed be accounts of *computation*, the very same phenomenon that we are trying to understand today with the notions of recursive functions, algorithms and the like, even if it is determined that such notions are not constitutive of *computation*.

However, this only establishes that transparent computationalism is possible; it does not guarantee that just *any* notion constrained by some future theory *T* can be considered a successor to the current notion of computation. It might be instead that the notions in *T* eliminate the notion of computation; or *T* may just be a different, unrelated theory. What must be true of *T* in order for it to be about the same ostensively individuated phenomena as current theories of computation? That is, what must be true of *T* in order for it to be an attempt at a theoretical account of what we pre-theoretically take to be computation? And what must be true of the notions *T* yields for them to be refinements to rather than usurpers of the notion of computation? For example, a typical tactic when taking the transparent computationalist line is to say something like "Yes, much of current computation is essentially digital. But there is some computation, such as what goes on in connectionist networks, which is not digital. So criticisms of computationalism that assume digitality are misplaced". But this assumes that an account of connectionism together with digital computation should be considered an account of computation, rather than an account of some category which includes computation and other phenomena besides. What justifies this assumption (or indeed, its negation)?

To some extent, it can't be justified — at least not on any traditonal theoretical/conceptual/logical/analytic/rational grounds. The determinants of the successor relation between theories are extra-theoretical. Because of the non-conceptualised nature of its starting point, any transition from a pre-theoretic view of a domain to a more theoretical perspective is, to some extent, extra-rational. But the transition is not *entirely* extra-rational, and it is certainly normative: there are good ways of conceptualising one's non-conceptually mediated experience of the world, and there are bad ways of doing so. And since conceptualisations of an empirical domain are always incomplete (because we are resisting the dogmatism which defines the domain to be that which is exhausted by our conceptualisation of it) the same goes for later transitions, from one partial conceptualisation of a domain to another.

There are several (non-logical) reasons why we should include a new phenomenon, ostensively individuated, into a previously existing class. For example, we might pre-theoretically call some new device (the "Watt machine") a computer, despite the fact that it is analogue and non-algorithmic, because it was produced by Intel (perhaps even by the same scientists and engineers who produced the Pentium III), requires many of the same materials and production procedures as does the Pentium III, can be used to perform tasks that we take (pre-theoretically or theoretically) to be in the same class as the tasks we use computers for, etc. But again, we must not replace theoretical dogmatism with what we might call empirical (or intuitional or pragmatic) dogmatism. If it turns out that there is no simple unified theory which accounts for both Watt machines and computers[5], then we would have to deem the Watt machine non-computational, despite the non-theoretical similarities to PCs. But if there is such a comprehensive theory *T*, the non-theoretical connections between the PCs and Watt machines would be sufficient to establish *T* as a refinement of our previous theories of computation. In such a case, the Watt machine would be confirmed as a computer.

This recognition of non-theoretical constraints on the theory/data dialectic also allows us to answer some other questions. Go back to the parable of the whales and fishes; at the end of that tale it was suggested that the theory refinement/intuition refinement cycle iterates indefinitely. But why should it? It seems that one cycle is enough: start out with an intuitive notion, come up with a theory that attempts to do justice to it, and use the best theory to go back and trim off the bits that don't fit well with the theory. How could the theory-tailored intuitions in turn demand a change in the theory which tailored them? The answer, as we have seen, is that theories aren't the only factors shaping our intuitions — the non-theoretical constraints provide perturbations that require theory to be ever ready to respond. Of course, it is an empirical issue whether a stabilised intuition/theory relationship can be found relatively quickly. The answer depends on such factors as the importance of the notion to the power structures in society, its relevance to current technological innovations, the inherent complexity of the theory involved, etc. It is my suspicion that the importance of computation in contemporary society, the fast rate of change in the technology, and the complexity of the artefacts involved make a quick quiescence unlikely.

In the foregoing I have focussed on the case of extending the concept of computation to new or fringe cases. But of course a change in the concept of computation might

[5] One might think that the fact that physics aspires to giving an account, in some sense, of all phenomena, or at least of both computers and Watt machines, trivialises this constraint of non-empirical dogmatism. This suggests that the additional constraint on best accounts mentioned in footnote 3 (the constrain of covering as little extra-domain phenomena as possible) might be a necessary one; its adoption would allow one to rule out physics as the best accoun of computers and Watt machines since it accounts for much else besides.

occur because it is realised that that change would do better justice to paradigmatic examples of computational systems. This provides an even more effective way of using the transparent reading of computationalism as a way to reply to its critics.

OBJECTIONS TO COMPUTATIONALISM

In the remainder of the paper I'll briefly review some objections to computationalism, looking at whether transparent computationalism can help rebut the objection. In light of the results of the preceding discussion, I will try to give a reason why we might expect the concept of computation to change in a way that disables the objection, if possible.

DYNAMICISM I

Recently, the non-dynamical nature of computational systems has come under attack. Specifically, van Gelder (1998) has argued that what is essential to computation is the notion of an effective procedure, and essential to that is the notion of discrete steps in an algorithm. He then claims that this discreteness, in both its temporal and non-temporal aspects, prevents computation from explaining many aspects of cognition, which he claims to be a fundamentally dynamical phenomenon.

The transparent reading of computationalism can be invoked here: if, in order to explain actual computers, it turns out that we need a more general notion of effective procedure, one which encompasses non-algorithmic, non-discrete, essentially temporal systems, then explaining cognition dynamically will be a possibility for the computationalist.

But do we have reason to include some dynamical systems into the pre-theoretic class of computational systems, as phenomena which computational theory should account for? I think so, and one need not appeal to some hypothetical Watt machine in order to make this point. The fact is, current computers do much of the work they do by virtue of their (currently) non-formal, temporal properties. In any real-time computational system, correct performance depends on the computer getting the timing just right. Consider two computational systems intended to perform the complex task of landing a plane. The two systems could have identical algorithmic or Turing machine characterisations — from the perspective of current computational theory, or at least the theory that critics of computationalism use as their. And yet one of these systems could have perfect timing, and be ideal for the computational task, while the other could be useless, having little or no correlation between the timing of its steps and the timing required to successfully land the plane. The difference between computational success and computational failure is completely beyond the non-dynamic version of computational theory. Since we need a time-involving theory in order to explain extant computational systems anyway, it is no real objection to computationalism to point out that we need the same to understand cognition.

DYNAMICISM II (THE MIND IS A WOT, GUV'NOR?)

Earlier, however, van Gelder made a different challenge to computationalism (van Gelder 1995). He argued that the conceptual anchor for understanding cognition shouldn't be the Turing machine, but rather the Watt governor, a simple device that, through its dynamical properties, regulates the speed of a steam engine. The Watt governor performs its function without any use of algorithms, representations, etc.; any such paraphernalia would merely impede its elegant means of co-ordination. Perhaps, then, we should also see minds as systems in coupled correspondence with their environments, rather than as discrete, representational, symbol manipulating systems.

This is an example of an objection to computationalism that is not handled well by taking the transparent approach. That is, it seems very unlikely that systems such as the Watt governor will be best understood in computational terms. If computation is essentially intentional (*about* something), and if Smith is right in claiming that intentionality requires both connection with and disconnection from one's subject matter (Smith 1996), then the close coupling between the Watt governor and its engine, which van Gelder extols, is exactly the reason for relegating it to the ranks of the sub-intentional and for thus deeming it to be less than computational. Simple connection and simple disconnection come for free in the physical world; negotiating between the two to maintain some abstract correspondence is a more sophisticated, and (at least sometimes) computational, achievement.

Since it seems unlikely that our notions of computation can be expanded to include the Watt governor, a different kind of reply is needed. However, one can use the way in which transparency failed to work as an indication of the basis of a proper response: if the Watt governor is too simple to be computational, perhaps it is too simple to be a good conceptual anchor for cognition (Clark and Toribio 1994). However, there is a slightly different Watt governor, or a description of the Watt governor that is different from the usual characterisation, which might have the complexity necessary for being both a conceptual anchor for cognition and for warranting an extension of the notion of computation. Consider a Watt governor that can become temporarily disengaged from the engine it is regulating (such a governor might be useful in the same way that a clutch is useful in a car). If the Watt governor becomes momentarily disengaged, it doesn't instantly stop spinning. If it did, it would adopt a configuration which it should only be in when the engine speed needs a maximal increase, which would be inappropriate in the case we are considering. The Watt governor needs some way of distinguishing temporary disengagement from the (relatively rare) case of the engine needing maximal acceleration. This is provided naturally in the dynamics of the governor: even when it is disengaged, its inertia and momentum mean that it will only slowly, and gradually, reduce its speed and thus the angle of its arms. Thus, when re-engagement occurs, it is likely that there won't be an enormous mismatch between the speed of the governor and the speed it should have, given the engine speed at the time of re-engagement. The suggestion here is that the inertia of the governor maintains a correspondence between the speeds of the engine and the governor even when they are disconnected. Is this a kind of negotiation between phases of connection with and disconnection from the subject matter, which Smith claims is at the heart of intentionality?[6] If so, or if one can imagine modifications to the governor that allow notions of intentionality to apply without destroying the Watt governor's purely analogue, dynamical nature[7], then one would have grounds for extending the notion of computation and taking the transparent computationalist line as a way of responding to van Gelder.

UNIVERSAL REALISATION

Another criticism of the computational approach is that its formality renders it universally realisable — Putnam (1988) and Searle (1992) argue that any physical system can be interpreted as realising any formal automaton. This has the consequence that an account of cognition cannot be in terms of formal computation, since any particular formal structure, the realisation of which is claimed to be sufficient for cognition, can be realised by any physical system, including those that are obviously non-cognitive.

The previous two subsections were examples of two different situations: cases in which the notion of computation could be extended to cover the phenomenon in question, and cases in which it could not, respectively. This subsection is an example of a third situation: cases in which the notion of computation need not be extended to new phenomena, but rather re-conceived concerning its application to central, undeniably computational phenomena. Specifically, the solution that has been proposed by Chalmers (1994), Chrisley (1994), et al. has been to acknowledge an aspect of traditional automata theory that has lain dormant until these objections were raised: the essentially

[6] Compare the inertial Watt governor with the super sunflower (Smith 1996, p 202 ff.).

[7] For example, perhaps one could arrange the dynamics of the governor such that there was a not only a first order correspondence between governor and engine speed during disconnection, but a second order correspondence as well. I.e., consider a Watt governor that, if it was detached from the engine during a period of engine speed increase, would increase its own speed during the disconnection, at an appropriate rate.

causal nature of computational state transitions. It is only this causal requirement that explains our pre-theoretic view that although a PC may be a realisation of a Turing machine *T*, the same is not true of the set of display screen states graphically depicting *T*, even though both PC and screen go through an isomorphic pattern of states. Once this is acknowledged, then it is seen that computation is not universally realisable, since (*pace* Putnam) the requisite counterfactual-supporting causal transitions necessary for the realisation of a particular automaton will not be found in an arbitrary physical system.

With respect to transparent computationalism, this kind of reply seems to be a borderline case. Does making an implicit aspect of the traditional notion of computation explicit count as changing the concept of computation? Inasmuch as the answer is yes, this line of response to the universal realisability objection to computationalism adopts the transparent reading of computationalism. Inasmuch as such explicitation does *not* count as conceptual change, then it is a refutation of a more direct sort — the transparent approach is not required.

EXTERNALISM

Mental states are relationally individuated (Putnam 1975); computational states are not (Fodor 1981), therefore computation cannot explain mentality (Putnam 1988), (Fodor 1994). That's the externalist objection to computationalism, in a subtlety-ravaging nutshell. The transparent approach is to question the second premise. Peacocke has done just this by arguing that even conventional computational explanations are essentially relational and world-involving (Peacocke 1994).

DIAGONALISATION

Consider the family of questions: Does the *k*th Turing machine halt on input *n*?

A familiar diagonal argument shows that there is no Turing machine which can answer all *n*, *k* instantiations of this question.[8] Supposedly we humans can.[9] Thus, following Gödel and Lucas, it is still argued (e.g., in (Penrose 1989) and (Penrose 1994)) that there are things which we can do which no Turing machine can, so computationalism is false.

[8] Why not? Because any Turing machine which supposedly could do this would appear as some k=K in the list of Turing machines. When it got around to considering the case of *K*, it either halts or it does not. If it does not halt, then in order to compute the function it would have to halt, a contradiction. If it does halt, then it is claiming that it does not halt, and so gives the incorrect answer. See Penrose (1994) for a clear and thorough explanation.

[9] The main reason for believing this is that we humans can follow the reasoning in the previous footnote.

There are *many* ways to respond to this argument, but the transparent computationalist approach seems novel: reject the last step as a non sequitur. Even if Turing machines can't do everything humans can, that does not imply that *computers* can't. Humans can do more than Turing machines? Big deal: *computers* can do more than Turing machines can as well. For example, computers can take up space, consume energy, exert gravitational force on their neighbours, keep time, etc. Now it is true that traditional theory would have it that these properties are irrelevant to the computational properties of a system. But that is an odd position, since it is by virtue of such properties that a computer has the computational properties it has and gets any computational work done at all. And as computers become more entrenched and embedded in the real world, those "non-computational" properties become more and more relevant to *computational* success or failure. The case of timing was discussed before, and this alone would be enough to allow one to conclude that computers can do more than Turing machines. Suppose the *mass* of the computer on the Pathfinder robot stabilises its movement to such an extent that the visual navigation algorithms implemented in the computer can actually work. Is the mass of the computer then irrelevant to the computational success of the processor? If these non-formal properties are relevant to designing and understanding computational systems, theoretical results about the limitations of disembodied automata will be seen as less and less relevant.

Despite the foregoing, I doubt that the resolution of the diagonalisation objection really has much to do with the fact that humans and computers are embodied while Turing machines are not. Rather, I think diagonalisation is something *no* entity can escape, be it abstract or concrete — including humans. Consider the person-halting problem: suppose we enumerate all persons, and all questions of English. Then, is there any person who can answer the following question correctly for all values of *n* and *k*?:

Is the kth person's answer to the nth question "no"?

No, and for the same diagonalising reasons as for the traditional halting problem. But notice that there is no non-question-begging argument against the possibility of a single Turing machine being able to answer all those questions. Since Turing machines are not mentioned in those questions, there is no way for them to trip up a Turing machine on a case of contradictory self-reference. In short, a transparent response, although possible, is probably not the best solution here.

THE CHINESE ROOM

The same goes for the Chinese room argument (Searle 1980). This argument aims to show that strong AI is false: all mental states cannot be had simply by virtue of implementing the right program, since in particular the mental state of understanding Chinese cannot not be had simply by implementing the right program. Searle's argument for this was that there is no program such that if Searle implements it, he will thereby come to understand Chinese—all he will be doing is meaningless symbol manipulation. So if one could get a computer to understand, it would have to be by virtue of something other than the fact that it was implementing the right program *P*. In computational terms, Searle implementing *P* and any computer implementing *P* are identical. So given Searle's non-understanding while implementing P, if some computer does actually understand Chinese while implementing P, it must be at least partially in virtue of a non-computational fact.

The transparent reading of computationalism allows one to resist this conclusion. One could agree that perhaps according to current theories of computation, Searle and any other system that is implementing *P* must be in the same computational states. But it might be that according to a better theory of computation, there are *computational* differences that are mere implementation detail according to current theory. So mental properties might supervene on computational properties in the transparent sense even if they do not supervene on algorithmic properties. Specifically, one could claim that program *P* when implemented by a non-understander is sufficient for understanding Chinese, but program *P* when implemented by an understander (such as Searle) is not. If one could also motivate the claim that the distinction "implemented by an understander vs. not" is a computational one, then one would have a means of resisting Searle's conclusions.[10]

A similar move can be used to counter another Chinese problem for AI, Block's claim that although the entire population of China could be linked up to realise some program supposedly sufficient for understanding, it seems absurd to say that thereby the entire nation of China was having a conversation about the Mets or whatever. If realisations that involve understanders are computationally distinct from those that do not, then one is not committed to saying that a program which is sufficient for understanding English when realised by a conventional computer is also sufficient for such understanding when implemented by the population of China.

The problem with trying to make the transparency move against either Searle's or Block's objection is that we have no independent grounds to suppose that "implemented by an understander vs. not" will be an interesting, generalization-supporting, explanation-providing distinction in millennial computer science. Thus appeal to it is merely post hoc, or a version of the Many Mansions reply (Searle 1980). As with the diagonalisation arguments, it is probably best not to reply to Searle and Block using the

[10] I believe this kind of transparent computationalist move was originally proposed by Putnam.

transparency reading of computationalism, at least not in this way — some other response is required. Fortunately, many are at hand.

But perhaps transparent computationalism isn't finished here. Harnad (1990), convinced by Searle's argument and not impressed with the many attempts to rebut it, has conceded that more than symbol processing is required for cognition—symbols need to be grounded in non-symbolic interactions with the world. If Harnad is right, there might be trouble ahead for opaque computationalists, who are typically thought to take computational properties to be independent of computer—world relations. But a transparent computationalist need not be worried, under one condition—that non-symbolic interactions with the world are seen to be crucial to understanding uncontroversially computational systems as well.

SUMMARY

If one is committed to computationalism, one must ask: why? Is it because of a fondness for the particular formalisms with which we currently understand computation, and the belief that these will provide an account of cognition? Or is it because of an intuition that there is something special about computers, when compared to other human artefacts, and that whatever explains that specialness will also explain the specialness of human cognition? Those who opt for the latter have a means of responding to objections to computationalism, a means that those who opt for the former lack. But these transparent computationalists thereby incur the responsibility of identifying phenomena that call for new theories of computation, and of producing such theories.

REFERENCES

Chrisley, R. (1994). "Why everything doesn't realize every computation". *Minds and Machines*, **4(4),** 403—420.

Clark, A. and Toribio, J. (1994). "Doing without representing?". *Synthese,* **101,** 401—431.

Fodor, J. (1981). "Methodological solipsism considered as a research strategy in cognitive science". In J. Haugeland, editor, *Mind Design.* MIT Press: Cambridge, MA.

Fodor, J. (1994). *The Elm and the Expert.* MIT Press: Cambridge, MA.

Harnad, S. (1990). "The symbol grounding problem". *Physica D,* **42**, 335—346.

Peacocke, C. (1994). "Content, computation and externalism". *Mind and Language*, **9(3)**, 303—335.

Penrose, R. (1989). *The Emperor's New Mind.* Oxford University Press: Oxford.

Penrose, R. (1994). *The Shadows of the Mind.* Oxford University Press: Oxford

Putnam, H. (1975). "The meaning of meaning". In: *Mind, Language and Reality: Philosophical Papers, Volume 2.* Cambridge University Press: Cambridge, MA.

Putnam, H. (1988). *Representation and Reality.* MIT Press: Cambridge, MA.

Searle, J. (1980). "Minds, brains and programs". *Behavioral and Brain Sciences* **3**:3, pp. 417-458.

Searle, J. (1992). *The Rediscovery of the Mind*. MIT Press: Cambridge, MA.

Smith, B. C. (1996). *On the Origin of Objects.* MIT Press: Cambridge, MA.

Van Gelder, T. (1995). "What might cognition be, if not computation?". *Journal of Philosophy* **92**, pp. 345—381.

Van Gelder, T. (1998). "The dynamical hypothesis in cognitive science". *Behavioral and Brain Sciences*, **21**, 615-665.

Ronald L. Chrisley
`ronc@cogs.susx.ac.Uk`

School of Cognitive & Computing Sciences
University of Sussex
Falmer BN1 9QH, UNITED KINGDOM