Consciousness as the Domain of a Computation

James Hodson*

June 2009

Abstract

Modern theories of mind and of cognition tend to revolve around the same basic question: how is it possible–given the assumption that we live in a world made of physical objects–for the brain to give rise to the rich conscious experiences that manifest themselves to us each and every waking or dreaming moment of each and every day? What is it about the physical brain that sets it apart from a piece of wood, or an oriental rug, or the computer that currently sits on my desk? I address this question through a computationalist framework, addressing the stronger objections that have been laid at its door to-date.

Keywords: consciousness, computationalism, thought experiments

*Princeton University, jhodson@princeton.edu (now james.hodson@ijs.si)

1 Introduction

Modern theories of mind and of cognition tend to revolve around the same basic question: how is it possible–given the assumption that we live in a world made of physical objects–for the brain to give rise to the rich conscious experiences that manifest themselves to us each and every waking or dreaming moment of each and every day? What is it about the physical brain that sets it apart from a piece of wood, or an oriental rug, or the computer that currently sits on my desk?

In conjunction with this question, we are led to ask the ever more pertinent-sounding follow-up: if the brain gives rise to conscious experience, then why not also all other physical objects–how do I know that my arm isnt currently having a direct conscious experience of its own? Any theory that addresses the question of consciousness, and its place in a physical universe, must offer a definitive answer to these questions and many more, in order to satisfy all aspects of this seemingly mind-boggling conundrum.

One theory that tries to address these issues is that of computationalism, whereby all that is required in order for any system to qualify as being conscious, is for that system to implement (run) an appropriate set of computations (the appropriate program). As long as we can map states of the system directly to states of the computation, then we can be justified in claiming that the system is in fact experiencing consciousness in exactly the same way as we would experience consciousness (if our brains were implementing the same computations).

This theory has been discussed and developed at length by David Chalmers (1994), and has received much attention from philosophers dissatisfied by the conclusions that may follow from it (Searle, 1980; Maudlin, 1989). Maudlin in particular argues that, because the computationalist is committed to viewing consciousness as essentially a by-product of a system, his theory is inadequate in explaining how a large mass of inactive components (as we assume must occur relatively often in the brain)–some of which may never become causally relevant–could possibly be necessary in order for consciousness to proceed from the system as a whole. If this were necessary, then the computationalist framework would be insufficient for explaining consciousness.

In this paper I will aim to show that Maudlins attack on the computationalist theory neednt cause any trouble to the view as a whole, and is in fact based upon a misunderstanding of the necessary and sufficient conditions for a system to be seen as implementing any given computation. However, Maudlins argument serves as a useful starting point for exploring possible weaknesses of the computationalist view, and therefore allows the development of an even more robust theory.

This paper will be sub-divided into three further sections. Section two will lay the foundations for the computationalist view, and expose each of its underlying assumptions to common criticisms. Firstly, I will clearly define the terms that are used when explicating the theory, and then show that if we did not take the implementation of a particular computation as being the necessary and sufficient condition for consciousness in a physical system, then it is difficult to see how we could proceed whilst maintaining a primary assumption that the universe is purely physical in nature.

The third section will be dedicated to a defence of the computationalist view from the argument that Maudlin gives in Computation and Consciousness (1989). The defence takes the form of a thought experiment designed to show that the computationalist is not bound to Maudlins original implicit assumption, namely that consciousness is derived from an entire physical system that is implementing a computation in a sub-part of itself at any given time. These terms will be explained more fully in what follows.

Finally, the fourth section will conclude that although Maudlins argument is misplaced, the computationalist theory is still not completely safe from potential attack. I will propose other areas that need restating, and possible starting points for such developments. Having said that, we will see that the computationalist approach remains the most hopeful for answering those questions originally posed, and that any modifications that may be required will not be so groundbreaking.

2 Foundations of Computalionalism

In order to understand the computationalist view it is helpful to clarify exactly what is meant by a computation. A computation is simply a high-level description of the steps required in order to transform a given range of inputs of a certain kind to an output of a certain kind.

Usually, computations define States of a physical system, and the kinds of connections between those States (transitions) that are required in order for the system to achieve this goal. In short, a computation is an abstraction of a physical system that describes its workings in a formal manner. In order to implement a computation, then, it is necessary for each of the physical States and State-transitions of a system X to map correspondingly (on a 1:1 basis) to States and State-transitions of the computation Y that it is implementing.

For a computationalist, if such a mapping exists, then we are able to say that system X implements computation Y. In this context, the abstraction of a Turing Machine is often seen as helpful in grasping the idea, so this is where we will begin.

A Turing Machine is a useful abstraction to have for this discussion, because it is very simple to understand, with few rules that govern its functioning, and yet provides the same computational power as much more complicated systems. Any finite computable function may be computed using a Turing Machine. It follows then, that for a computationalist, showing that States and State-transitions of a computation can be mapped to States and State-transitions of a given Turing Machine, will provide a good basic model for a theory of implementation.

Essentially, a Turing Machine works as follows. A tape that is infinitely long in both directions, and sub-divided into an infinite number of squares of equal size, contains the data that is to be used as input. Any individual square along the tape can contain either a stroke (denoted by 1) or be empty (denoted by 0). Next, the Turing Machine has a head that can move sequentially back and forth along the tape, and perform one of three actions after reading a particular datum from a square on the tape: do nothing, write a 1 in place of what is currently in the square, or delete what is currently in the square in order to leave it blank (0). The action that the machine performs is determined by its final component, the State table. Each State tells the machine to perform a single action, and provides the next State that the machine should move to.

Explicitly, the State may tell the machine to move one square to the left or right of the square that it is currently on, or read the square that it is currently on and branch (conditionally) to the next instruction based upon what has been read. For example, the compound statement, if (0) move right, else move left, might translate to:

State 1: Read the tape; If the Symbol is a 0, then move right and go to State 2; Otherwise, move left and go to State 3.

This simple instruction set is powerful enough to allow us to implement a wide variety ofcomputations, and can be expanded in various ways to add explanatory power–although its computational power will not increase. That is to say, a Turing Machine can implement the same set of sequential algorithms as any modern computer theoretically could (ignoring time and space constraints).

With this in mind, let us turn now to the central tenet of computationalist thought, the thesis of Computational Sufficiency. Put simply, this states that having the appropriate computational structure, by itself, is enough to qualify as possessing a mind and a wide variety of mental properties. As long as a system is running the correct program (or, as we have been discussing, implementing an appropriate computation), then at any point within that program it will have the same conscious states as any other system that possesses a mind. This means that all of our conscious experience is a product of the computation that (presumably) our brain is instantiating–nothing else is required in order to create the wide variety of experiences we have as human beings.

In addition, a second thesis, that of Computational Explanation, flows directly out of that of Computational Sufficiency. In general, if a computation suffices to give rise to these phenomenae (a mind), then necessarily that which we associate with consciousness–behaviour and cognitive processes–will be derived from the computation being implemented, and nothing more. This thesis is strongly linked to the assumption a computationalist must make, that it doesn't make a difference what kind of matter is involved in the physical implementation of a computation. Neurons, or silicon, as will be later explained, are fully equivalent and produce the same implementation of a computation both qualitatively and quantitatively.

In his paper, A Computational Foundation for the Study of Cognition (1994), Chalmers puts forward a computationalist theory based on a slightly enhanced version of a Turing Machine, and addresses several possible objections to the general computationalist position. I will explicate the theory that Chalmers puts forward, along with insights from another paper of his (Chalmers, 1995), which addresses the reasons for believing that the implementation of a computation is in fact the best way to frame our problem.

Chalmers (1994) attempts to explain why Computation rather than any other discipline is the most valid and fruitful approach to help us understand the mind and cognitive properties. He believes that although a Turing Machine is powerful enough to represent any computation, a more intuitive machine is required for the purpose of bridging the gap between our conception of consciousness and our conception of a computation. The machine that he introduces is the Combinatorial State Automaton (CSA), which essentially acts as a Turing Machine that instead of being in only one state at a time, participates in an over-all State that is defined by a State Vector. Therefore, rather than being in State S1, for example, a given machine will be in State S1,S2,S3.

This, Chalmers believes, allows for a clearer mapping from States of a computation, to States of a physical system[1].

[1]See Brown (2004) for a further discussion of the adequacy/inadequacy of the CSA

Chalmers describes implementation using the terms that we defined earlier, whereby a system can be seen as implementing a computation when there is a direct mapping from physical states of that system, to states of the computation. The next step in piecing together the computationalist theory is addressing how computation relates to cognition–that is, how we get from the implementation of a computation, to any appropriate physical system actually having conscious experience in a similar way to humans.

Firstly, Chalmers states that a behavioural approach to this explanation would be insufficient. A behavioural approach would be to see how a system behaves when given certain input (stimuli), and concluding that two systems are isomorphic if the behaviour displayed by them is the same in all respects. Chalmers believes such an explanation to be insufficient because we can easily envisage two distinct computations giving exactly the same output. To make this point even clearer, let us imagine two systems that compute the product of three numbers: a, b, and c. The first system multiplies a by b, and then multiplies that total by c; the second system adds a to itself b*c times. Although the two systems invariably produce the same output, they have very different internal processes, different States that they can enter, and a different order in which they are entered.

Another problem with the behavioural approach is that depending on the input we give to two systems, we might end up classifying two systems as equal when in fact they only happened to overlap in behaviour for the input we gave them. So, such a method is insufficient in that it only concentrates on the overall output of a system. What is needed is a stronger link to each of the sub-computations occurring–that is, the causal organisation.

Chalmers asserts that the invariance of support for cognitive function (conscious experience) is maintained by any system that can be seen as the original system with any number of specific transformations applied–moving, distorting/stretching, sufficiently localised re-circuiting, and any other changes that do not alter causal interaction.

Mental processes, that is, cognitive processes, are organisational invariants according to the computationalist view. They depend entirely upon the causal structure of the system–whereas many other properties do not (flying, for instance, as the causal organisation of something that flies is retained even when not flying, whilst the property of flying is not).

But why ought we to think that cognitive processes are organisational invariants rather than physical invariants (i.e. dependant entirely upon the material in which they are instantiated)? Chalmers offers us several reasons

model proposed by Chalmers.

in his paper, Absent Qualia, Fading Qualia, Dancing Qualia. If the causal organisation of a system is all that is required for cognitive properties to be evinced by it, then it is difficult to see why anything more could be required to produce conscious experience (such as the material from which its constituent parts are crafted).

Chalmers attempts to show the empirical improbability of envisaging such dependence. The thought experiments referred to are those of absent qualia, fading qualia, inverted qualia, and dancing qualia.

A case of absent qualia[2] occurs when any introduction of a non-neuronal pathway in the critical conscious sections of a system induce an entire absence of conscious experience in the system–although cognitive processes continue as per usual. That is to say, that we take a fully functioning brain, and replace one of the millions of neurons controlling sight perception with a silicon-based duplicate of that neuron (imagine that we have invented a procedure that allows us to do this without interrupting any of the brains functions). It seems highly unlikely that replacing one neuronal link among billions with a silicon link would cause an end to conscious experience. Bearing in mind that the particular neuron being replaced is perhaps not even directly related to other parts of our conscious life, and that the silicon neuron performs in exactly the same way as the original, such a small change seems rather unlikely to cause a complete loss of our conscious experience.

There would need to be something extremely special about our chemical-biological make-up in order for this to occur, and whats more, this still wouldnt solve the question of what is sufficient for conscious experience to arise in a system (for if it did, then we would be forced to say that anything that is made from neurons would have conscious experience, regardless of the causal connections between those neurons).

One temporary way around this problem is to accept the thesis of fading qualia, which states that the more neurons are replaced, the less vivid our experience becomes. In this way, there is no magic moment at which consciousness ends, but rather, certain conscious experiences will fade out sooner than others, eventually leaving no conscious experience at all when the system has been completely replaced by silicon neurons. Of course, regardless of whether or not qualia are fading from the system, it will in no way affect the cognitive processes occurring in the brain–as these are a product of organisational invariance, not of conscious experience in and of

[2]Qualia are the qualitative experiences associated with our conscious operation–such as what the colour Red looks like to us, or how smoked salmon tastes. These individual experiences are wholly subjective, and are only open to introspection.

itself. For example, even if the faded qualia patient does not experience vivid red when he sees a red car, if asked he will state that the experience he is having of red is very vivid, as vivid as ever. This is due to the fact that neurons send each other signals in order to effectuate actions, a purely causal relationship–regardless of whether or not there is feeling attached to them.

Finally Chalmers considers the case where the replacement of neurons with silicon causes localised qualia inversions–i.e. instead of experiencing the same red when seeing red, we experience what is by all other accounts blue. Chalmers shows us this case because it draws together the idea of the causal layout of the system determining experience, and a refutation of the thought that qualia could change for any other reasons except the causal structure (the implementation of the appropriate computation).

The experiment is simple: we replace a portion of the brain large enough to have a significant experiential difference on a specific variable (such as the colour red), with a silicon system that has the same causal structure[3]. We will keep the portion as small as possible to avoid too much conscious experience changing (that we might infer that we had given rise to a different person if in fact the causal structure were not the only thing at play). We could imagine placing these two systems (neural and silicon) in parallel, with a switch between them. Each time we flick the switch the experience changes from red to blue, and back again–and yet, because of the organisational invariant, the person will not display any different behaviour to normal, or manifest any peculiar behaviour–their cognitive processing will continue as normal even though their experience of red continues to switch back and forth from red to blue (and they would not even express this difference if asked!).

If this were possible, Chalmers argues, then psychology and phenomenology would be radically out of step–appearing to have very little influence on each other. Furthermore, this could lead to a situation in which qualia are continuously shifting regardless of whether or not neurons are replaced by silicon chips–simply from the constant molecular-level changes occurring in the brain–and we would never be aware of it!

These hypotheses seem to Chalmers to be empirically improbable–as the idea of unconscious qualia inversions is absurd (as qualia are conscious manifestations). The most plausible conclusion would be that no change in

[3]It is important for the systems to have the same causal structure, and not just the same input/output–because the high-level implementation details are important, as noted in the product of three numbers experiment.

qualia would result from any of the experiments. Thus, it seems that the most rational explanation of conscious experience is as an organisational invariant.

Chalmers (1994) provides responses to possible objections to the computationalist view that we have outlined. One of the most famous of these objections is a thought-experiment proposed by John Searle (1980). Searle proposes an experiment involving a man inside a room that has an input slot, and an output slot. The man is given a set of instructions regarding how to manipulate formal symbols that he receives through the in-slot, in order to provide a set of different symbols through the out-slot. The man inside the box has no idea what the symbols he is manipulating mean, he just follows the set of rules in order to produce output.

Unbeknownst to him, people outside the room use the in-slot and out-slot in order to post questions in Chinese, and receive coherent Chinese answers from the room. As time passes, the man inside the room becomes so good at the symbol manipulations, that to everybody outside, the answers are indistinguishable from those of a native speaker of Chinese. Yet, Searle says, the man cannot be said to understand Chinese–after all, he is simply using the given instructions in order to produce output that is based on the input; the symbols have no meaning for him. Furthermore, the system cannot be said to understand Chinese either, because we could imagine the man not being inside a room, and internalising all of the rules so that they come naturally to him–the output would be the same as a native speaker, but he would not know a word of Chinese!

Searles argument is supposed to show that the room is exactly like a computer implementing a program that understands Chinese in the behavioural sense, and yet it is inaccurate to say that the room understands Chinese in the same way that the man understands English as his native tongue. This is because, although the formal structure might be the same, the system lacks the semantics required in order for it to understand the content it is processing. Therefore, the implementation of a computation alone cannot suffice for the instantiation of a mind.

Chalmers response is to say that although the man does not understand Chinese per se, the system as a whole inevitably does. If the computation being implemented by the man in the room is equivalent to the computation being implemented by a native speaker of Chinese, then we are quite justified in saying that the system understands Chinese in exactly the same way as a native speaker understands Chinese.

Searle has a response to this, which is to imagine that the man internalises all of the instructions to the point that the room and the instruction set are no longer required. Then we are left with a man who knows all there is to know about the implementation of a system, and yet does not understand Chinese–despite appearing to everyone else as if he does. This again invites the response of if the computations are equivalent, then the understanding derived would also be equivalent.

Instead, let us try to look beyond this latter response, as to where the divergence lies between these two views, and see if it cannot be reconciled. Searles objection is that the semantics provide an obstruction between simply following rules, and actually understanding Chinese. As Searle states that machines do not have semantics, but brains do, then this at first appears to be an insurmountable barrier. However, if we were to properly analyse the computation occurring in the brain of a native Chinese speaker, we would be able to see that the knowledge of each symbol is grounded in a representation of that symbols meaning in the world. For example, the symbol for a table is linked to whatever idea is evoked in the native Chinese speakers mind when he/she hears the word table uttered. In this way, if the computation that is being implemented by the man in the room did not somehow incorporate the elements of understanding that link the word table to the idea of a table in the mind of a native Chinese speaker, then of course the man in the room would not understand Chinese. However, if the computation were to incorporate these aspects, it seems harder to object that the man-in-the-room system wouldnt understand Chinese.

So, in short, the reason why Searles thought-experiment seems to show that the system wouldnt understand Chinese, is because it purposefully does not take into account the fact that the native Chinese speakers computation is tied to the non-linguistic representation of the symbols (as an interwoven part of the computation). Instead, the man in the room has no instructions relating the symbols to their representations, and is therefore not implementing the appropriate computation for the understanding of language. Now that we have laid down the foundation for considering issues relating to the computationalists framework, we can move forward to consider an objection that was not dealt with at all in Chalmers papers–that of Maudlin.

3 Defence and Thought Experiment

Maudlin (1989) argues that the mere implementation of a computation cannot suffice for consciousness. This intuition is based on the fact that we can imagine a complex system to which a computationalist would assign mentality (consciousness), then consider only a small task carried out by a tiny subsection of this system (the active part).

If we isolate this subsystem then the computationalist is supposed to assert that this new subsystem would have no conscious experience (as the conscious experience is a product of the entire system). But then, Maudlin argues, how could the existence of consciousness depend upon the presence or absence of a huge mass of inert matter being present and connected to the subsystem? Surely, if there is no conscious experience in the subsystem, having all of this excess paraphernalia will not help one bit to create it.

The intuition here could be described in a different way. A computationalist would perhaps argue that the system as a whole is conscious, rather than say that the sub-part of the system that handles vision is conscious. Then, if we were to isolate the system that handles vision, a computationalist would certainly deny the idea that such a subsystem could be conscious. The reason for this is that a computationalist sees consciousness related not to localised events in the system, but rather to the way those local subsystems interact with the parts of the system that serve consciousness (that help us to understand the goings on in the other areas, in this case the visual arena). That is to say, if all of the system that is required for consciousness is present in the subsystem at any time, then that subsystem will be conscious, otherwise it will not be. It is also important to note that the part of the system responsible for consciousness would be (one would think) always active.

In order to illustrate these points more clearly I propose a thought experiment as follows: Let us imagine a human brain, made of billions of neurons. Scientists have recently developed an extremely novel procedure which allows neurons to be connected and disconnected without affecting their individual health–therefore, when a neuron is not firing, or being fired at, it can be removed without losing its potential. We could imagine that such a technique could be used for maintenance purposes in order to extend the lives of future generations. The way this procedure works is that it removes the neuron when it is playing no causal role in the system[4], fixes it, and then puts it back in just before it is needed again.

Under Chalmers view of computation, the causal structure of the system has not been affected when we remove and replace these neurons. Now,

[4]This is a rather clever procedure, which is able to keep track of the current firings of all Neurons in the brain, and of the shortest time period possible before relevant neurons could fire, and there could be a need for the neurons out for maintenance. Counterfactuals of the form: "if the stimulus were suddenly changed...", cannot be applied here, as 1) this would place the behaviour of the procedure outside of its causal structure, 2) our clever procedure would keep track of such a change and modify its actions accordingly.

further imagine that this procedure is highly efficient, and is capable of removing many neurons from the brain at once–only those that are playing no causal role (which according to Maudlins example is the majority of the brain). Causally, the brain works exactly as it does at any other time, and certainly no computationalist with Chalmers view of implementation need admit that at any moment this system has conscious experience that differs from the full system when not being maintained.

Finally, we can imagine this same machine being used to perform Maudlins experiment, removing so many neurons at a single time that only those related to a tiny subsection remain in place. As we have just shown, this does not affect the causal structure of the system, and therefore need not be regarded as a threat to the computationalist view of consciousness as being directly linked to the implementation of the correct computation.

One of the side effects of Maudlins claim is that, if true, conscious systems could pop into and out of existence all of the time, simply by momentarily having the right state transitions.

The computationalist response to this claim is that, yes, if an appropriately structured system were to momentarily have the correct state transitions, then it would momentarily be conscious. However, it is important to note here that it is highly unlikely that such a large number of inactive neurons would ever exist–as any system implementing a conscious being would most probably be continuously active across large portions asymmetrically. So, the chances of such a complex system fleeting into and out of existence in nature is rather low–though certainly not impossible.

4 Future-proofing Computalionalism

Now that we have shown Maudlins argument to not be such a dangerous threat to the computationalist view after all, we can move on to consider whether our formulation is strong enough to withstand other possible attacks. These attacks are aimed not at whether or not implementation is sufficient for consciousness, but rather, at whether the theory is capable of expressing the relationship between implementation and consciousness adequately. If our computationalist theory were found to be lacking in explanatory power, then it would need to be restated, which may be just as detrimental as an attack to the central tenets of the theory.

I will not offer a restatement of the theory here, but I will provide insights into possible solutions to the problems about to be posed. The first of these explanatory gaps in the computationalist theory is a product of the computational model used pervasively–that of the Turing Machine.

As described at the beginning of Section I of this paper, the Turing Machine is capable of computing any computable function, and as such is capable of carrying out any computation that models the conscious mind. However, being powerful enough to compute a given function does not suffice in and of itself. Implementation requires not only the power to compute, but also a direct mapping of States of the computation, to states of the physical system involved.

Such a direct mapping appears at first to be a simple matter of spatial and causal correspondence, but in fact, a third correspondence is most certainly also relevant–temporal. In order for a system to count as implementing a computation, we ought to require of it that State transitions throughout the system occur in parallel with State transitions in the computation. A Turing machine can be in only one state at a time, whilst a mind is most probably in more than one state at any given time.

Chalmers offers us the Combinatorial State Automaton as a replacement for the Turing Machine–it is capable of being in several States at once, by virtue of a State vector representing the current overall State of the machine. The problem, nonetheless, is not that of not being able to represent enough States, but rather, that of only being able to represent the synchronous execution of a multi-State computation–whereas a mind does not have synchronous limitations.

Conscious experience is the conjunction of all of the sub-computations occurring within a conscious system. If these sub-computations do not overlap appropriately causally and temporally, then the system would have very disjoint experiences at best. Therefore, computationalism needs a better model of computation if it is to offer us a way of examining the claims fully.

A further, and final note is meant to frame future objections to computationalism in a manner that does not ignore the practical requirements of a system that implements consciousness. Namely, the fact that such a system would be in constant flux over the majority of its physical domain. Actions do not happen spontaneously, the build-up is a constant process, and consciousness/understanding cannot be understood independently of the whole.

For example, Searles Chinese room argument offers convincing anecdotal evidence that the system involved does not understand Chinese, but not in any interesting sense. If the room were equipped with the necessary apparatus for understanding Chinese, then it would understand Chinese.

When objecting to computationalism, it is not possible to break the problem of consciousness down into smaller chunks–it must be attacked as a concerted whole. Seen in this way, the theory is perhaps stronger than first perceived.

5 Conclusion

In conclusion, although the Turing Machine by itself may not be a powerful enough model of computation, there are several ways of envisaging a system that runs asynchronously and is capable of supporting the localised and variable development of sub-computations. This may even be achieved through an array of slightly modified Turing Machines, but I will not go into the details of such an arrangement.

When taken in the way described in this paper, the computationalist stance offers firm responses to all criticisms laid at its door to date. The task now is to expand our understanding of the conditions of a physical system capable of implementing a conscious being, and ensure that the framework developed permits these.

6 Bibliography

[1] Boolos, George. 1968. Minds, Machines, and Gdel: God, the Devil, and Gdel. The Journal of Symbolic Logic, Vol. 33, No. 4, 613-615.

[2] Brown, Curtis. 2004. Implementation and Indeterminacy. Canberra Conferences in Research and Practice in Information Technology, Vol. 37.

[3] Chalmers, David. 1994. A Computational Foundation for the Study of Cognition.

[4] Chalmers, David. 1995. Absent Qualia, Fading Qualia, Dancing Qualia. Conscious Experience: Imprint Academic.

[5] Copeland, Jack. B. 1996. What is Computation? Synthese, Vol. 108, No. 3.

[6] Fallis, Don. 2000. The Reliability of Randomized Algorithms. British Journal of the Philosophy of Science 51.

[7] Maudlin, Tim. 1989. Computation and Consciousness. Journal of Philosophy 86(August):407-32.

[8] Peacock, Christopher. 1994. Content, Computation, and Externalism. Mind & Language, Vol. 9, No. 3

[9] Putnam, Hillary. 1960. Minds and Machines. Dimensions of Mind: New York University Press.

[10] Searle, John. 1980a. ”Minds, Brains, and Programs.” Behavioural and Brain Sciences 3, 417-424.

[11] Siegelmann & Sontag. 1993. Analog Computation via Neural Networks. Theoretical Computer Science 131.