Can we Prove that there are Computational Correlates of Consciousness in the Brain?*

David Gamez

University of Sussex, United Kingdom
david@davidgamez.eu

Scientific research on consciousness is attempting to gather data about the relationship between consciousness and the physical world. The basic procedure is to measure consciousness through first-person reports, measure the physical world and look for correlations between these sets of measurements. While most of this work has focused on neural correlates of consciousness, it has also been proposed that consciousness is linked to the computations that are being executed by the brain. If this is the case, we would expect there to be a high level of correlation between some of the brain's computations and consciousness. This could be scientifically tested if a plausible method for measuring computations could be found.

This paper investigates whether Chalmers' method for identifying computations could be used to measure computations during an experiment on the correlates of consciousness. A number of arguments are used to show that Chalmers' account of implementation fails for a desktop computer, which makes it unlikely that it could be used to identify computational correlates of consciousness in the brain. While a different account of implementation might be able to rescue computational approaches to consciousness, the problems raised in this paper suggest that it is going to be difficult to develop a method for measuring computations that could be used to test whether there are computational correlates of consciousness in the brain.

*I would like to thank Barry Cooper and the John Templeton Foundation for supporting this work (Project ID 15619: 'Mind, Mechanism and Mathematics: Turing Centenary Research Project'). I would also like to thank Anil Seth and the Sackler Centre for Consciousness Science at the University of Sussex for hosting me as a Research Fellow during this project. I am grateful to Ron Chrisley and the reviewers whose comments have considerably improved this paper.

Journal of Cognitive Science 15: 149-186, 2014

Key words: *Implementation, computation, correlates of consciousness, functionalism, combinatorial state automata, finite state automata, brain*

1. Introduction

Consciousness is a significant research topic in philosophy and a variety of theories and elaborate thought experiments have been put forward. However, although a great deal of effort has been expended, it can be argued that our understanding of consciousness has failed to advance much beyond Descartes. The situation has changed in recent years with the emergence of the scientific study of consciousness. This aims to gather data about correlations between consciousness and the physical world while suspending judgement about the metaphysics and philosophical debates. Scientific data about the correlates of consciousness could help us to improve our philosophical theories about consciousness and it has many practical applications - for example, it could help us to answer questions about the consciousness of brain-damaged patients, infants and animals.

In an experiment on the correlates of consciousness, the general procedure is to measure consciousness, measure the physical world and look for a relationship between these measurements. The measurement of consciousness relies on the working assumption that consciousness is a real phenomenon that can be reported verbally ("I see a red hat", "I taste stale milk") or through other behaviour, such as pushing a button, pulling a lever, short term memory (Koch 2004) or the Glasgow Coma scale (Teasdale and Jennett 1974). The assumption that consciousness can be measured through behaviour enables us to obtain data about it without a precise definition. However, this reliance on external behaviour limits consciousness experiments to systems that are commonly agreed to be conscious, such as a human or a human-like animal. It is not possible to carry out this type of experiment on non-biological systems, such as computers, because a computer's reports are not generally regarded as a measurement of consciousness (Gamez 2012).

The physical world is measured to identify spatiotemporal structures that might be correlated with the measurements of consciousness. In this paper I am focusing on the correlates of consciousness in the human brain,[1] which will be defined in a similar way to Chalmers' (2000) definition of the total correlates of consciousness:

> **D1.** A correlate of a conscious experience, e_1, is a minimal set of one or more spatiotemporal structures in the human brain. This set is present when e_1 is present and absent when e_1 is absent. This will be referred to as a *CC set*.[2]

Correlates defined according to D1 would continue to be associated with consciousness if they were extracted from the brain or implemented in an artificial system. 'Spatiotemporal structures' is a deliberately vague term that captures anything that might be correlated with consciousness, such as activity in brain areas, functions, neural synchronization, computations, information patterns, electromagnetic waves, quantum events, etc.

In an experiment on the correlates of consciousness a contrastive methodology is typically used that compares the state of the brain when it is conscious and unconscious, or compares conscious and unconscious states within a single conscious brain - for example, using binocular rivalry (Logothetis 1998) or through the subliminal presentation of stimuli (Dehaene et al. 2001).[3] Systematic experiments are needed to test all possible combinations of spatiotemporal structures that might form CC sets (see Table 1). When one or more CC sets have been identified they can be used to make predictions about the level and contents of consciousness, which can be compared with first person reports.

While most scientists have focused on the neural correlates of

---

[1] I am focusing on the human brain because we are confident that it is linked to conscious states. Once we have identified the correlates of consciousness in the human brain we can use them to make predictions about the consciousness of other systems (Gamez 2012).

[2] In D1 CC sets are linked with individual conscious experiences. To improve the readability of the text I will often talk about the correlates of consciousness in general, which can be considered to be the complete set of CC sets.

[3] Reviews of some of this experimental work are given by Rees et al. (2002), Tononi and Koch (2008) and Dehaene and Changeux (2011).

Table 1. Illustrative example of correlations that could exist between conscious experiences (e_1 and e_2) and a physical system. It is assumed that e_1 and e_2 can occur at the same time. A, B, C and D are spatiotemporal structures in the physical system, such as dopamine, neural synchronization or 40Hz electromagnetic waves. A, B, C and D are assumed to be the only possible features of the system. '1' indicates that a feature is present; '0' indicates that it is absent. In this example D is not a correlate of consciousness because it does not systematically co-vary with either of the conscious states. {A,B} is a set of spatiotemporal structures that correlates with conscious experience e_1. {C} is a set of spatiotemporal structures that correlates with conscious experience e_2.

Spatiotemporal Structures				Conscious Experiences	
A	B	C	D	e_1	e_2
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	1

consciousness, there are several different types of spatiotemporal structure in the brain that could form CC sets. For example:

- *Physical correlates.* The correlate of consciousness is a spatiotemporal pattern in one specific aspect of the brain, such as neuron activity, electromagnetic waves or quantum states. In this case the CC set consists of the spatiotemporal pattern and the substrate in which the pattern is instantiated - the pattern would not be correlated with consciousness if it was present in a different physical substrate. For example, it has been suggested that some neural synchronization patterns might form CC sets. If this is a physical correlate, then the same synchronization patterns between oscillating electronic circuits would not be correlated with consciousness.

- *Informational correlates*. Tononi (2004; 2008) has suggested that the integration and differentiation of the brain's information states, known as 'information integration', is identical with consciousness. The material in which the information integration is instantiated is not relevant to this theory. If this is correct, there should be a high level of correlation between information integration and consciousness and information integration would form a CC set by itself.[4]
- *Computational correlates*. It has been proposed that the mind can be understood in computational terms or that consciousness is identical with some of the brain's computations (Cleeremans 2005). If this is the case, we would expect there to be a correlation between consciousness and the execution of particular computations, and this should be independent of the substrate or mechanism by which the computations are executing. This putative link between computations and consciousness is the main focus of this paper and it can be stated as the following hypothesis:

H1: There is at least one computation, C_c , that is a sole member of a CC set.[5] The architecture and material of the physical system that C_c is executed on are not part of this CC set and have no effect on whether C_c is correlated with consciousness.

If H1 is correct, it will be possible to create artificial consciousness by running a computer program. We could also create a copy of our

[4] Some of the difficulties with experiments on information integration and consciousness (Gamez 2014) also apply to experiments on the computational correlates of consciousness.

[5] A real world program or computation can be decomposed into collections of subroutines, calls to external libraries and the operating system, etc. In H1 all of these computations are considered to be the single computation, C_c.

consciousness by running a program that simulates our brain at an appropriate level of detail. The consciousness associated with these programs would be the same regardless of whether they are running on a modern desktop computer or on Babbage's Analytical Engine.

This paper examines whether H1 could be scientifically tested using the same methodology as an experiment on the neural correlates of consciousness. This could be done by measuring the computations in the conscious brain, measuring the computations in the unconscious brain and looking for computations that are only present in the conscious brain.[6] The difficulty with this type of experiment is that very few methods for measuring computations in physical systems have been put forward.

Modern digital computers execute programs and convert input strings into output strings, and it might be thought that we could identify computations in other physical systems by looking for executing programs or string processing (Piccinini 2007). The limitation of this approach is that it is unlikely to be applicable to the brain, since it is far from obvious that the brain's operations are based on programs or string processing. If we want to test the hypothesis that the brain contains computational correlates of consciousness, we will have to find a more general way of measuring computations that can be applied to a wide range of physical systems.

A second way of measuring computations in a physical system is to map the computation onto a finite state automaton (FSA) and determine whether the sequence of states of the FSA over a particular execution run with a particular input pattern matches the sequence of states of the target system. FSAs are often used to specify programs' state transitions and they can be used to identify computations in a wide range of systems, including the brain. The problem with FSAs is that a disjunctive mapping can be used to interpret a sequence of unique states in *any* physical system as the execution of a program, which leads to an untenable panpsychism (Putnam 1988; Bishop 2002; Bishop 2009). While the FSA mapping issues are disputed (Chrisley 1995; Chalmers 1996), it is likely that this approach is too liberal

[6] To prove that computations are sole members of CC sets it will also be necessary to demonstrate that the substrate or mechanism by which the computation is being executed is irrelevant. A discussion of this issue is given in Gamez (2012).

in its attribution of computations to physical systems. To make this clearer, suppose that we carry out an experiment on the correlates of consciousness and identify a FSA, FSA_1, in the conscious brain. If Putnam (1988) and Bishop (2002; 2009) are right, FSA_1 can be found in *any* sufficiently complex set of state transitions, and so we will be able to find it in the state transitions of the unconscious brain. If FSA_1 is present in the conscious and unconscious brain, it cannot be a correlate of consciousness. Experiments on the computational correlates of consciousness need an implementation method that is less liberal than FSAs.

The most promising method that I am aware of for identifying computations in physical systems is the combinatorial state automata (CSA) approach put forward by Chalmers (1996; 2011), which can be roughly summarized as follows:

1) A *causal topology* is an abstract causal organization, a pattern of causal interactions among a set of elements that is independent of any particular implementation.
2) *Organizationally invariant properties* of a system only depend on its causal topology.
3) When a system implements a particular causal topology it implements the organizationally invariant properties associated with that causal topology.
4) Mental properties are organizationally invariant properties.
5) Instead of FSAs, which have a monadic state, Chalmers uses a combinatorial state automaton (CSA) to specify a causal topology. Each CSA state consists of a vector of substates that can have a finite number of possible values.
6) A CSA is claimed to be an adequate way of specifying a causal topology.
7) To determine whether a physical system implements a particular CSA, the CSA substates are typically mapped onto distinct regions of the physical system. If the system implements the CSA, then the causal interactions between these regions will lead to the state transitions specified in the CSA.
8) Any system that implements a particular CSA has the

organizationally invariant properties that are associated with the causal topology specified by the CSA.

CSAs are related to FSAs, but more restricted in the systems that they apply to. This might enable them to identify computations in the conscious brain that are not present in the unconscious brain. CSAs are also more general than accounts of implementation based on modern digital computers. While many criticisms have been made of Chalmers' account of implementation (Scheutz 2001; Miłkowski 2011; Ritchie 2011; Brown 2012; Egan 2012; Harnad 2012; Klein 2012; Rescorla 2012; Scheutz 2012; Shagrir 2012; Sprevak 2012), it is far from clear what could be used to replace it if it was found to be seriously flawed (see Section 4).

Chalmers' account of implementation enables us to replace the general hypothesis that there are computational correlates of consciousness (H1) with a more concrete hypothesis that could potentially be tested:

H2: There is at least one CSA, CSA_c, that is a sole member of a CC set. The architecture and material of the physical system that CSA_c is executed on are not part of this CC set and have no effect on whether CSA_c is correlated with consciousness.

We can test H2 by measuring consciousness through first person reports, measuring CSAs in the brain (see Section 3.1) and looking for correlations between these two sets of measurements. This type of experiment must meet at least two requirements:

R1: *There must be an objective fact of the matter about which CSAs are being executed by a physical system.* If consciousness is correlated with the execution of a CSA in a physical system, then it must be possible to identify CSAs in a non-ambiguous objective way. Otherwise the results of an experiment on the correlates of consciousness will depend on the subjective interpretation of the experimenter.

R2: *A CSA that is a sole member of a CC set cannot be executed when the system is unconscious*. A CSA that is executed both when a system

> is conscious and unconscious cannot be sufficient for consciousness by itself - other things must be necessary as well.

This paper will examine whether the potential correlation between CSAs and consciousness could be experimentally tested in a way that is consistent with these requirements. To make the issues clearer I will focus on a simple test system, a desktop computer running a robot control program, which is described in Section 2. If Chalmers' method could successfully identify the computations in this test system, then we would have grounds for believing that it could be used to identify computations in the brain. Section 3 raises a number of problems that occur when we try to use Chalmers' method to identify computations in the test system in a way that is consistent with R1 and R2. These suggest that a CSA-based account of implementation cannot be used to identify computational correlates of consciousness in the brain. The discussion in Section 4 considers whether the problems with Chalmers' approach could be addressed by developing a better way of measuring computations in physical systems.

2. Test System

2.1 Introduction

Before Chalmers' CSA-based account of implementation is applied to the brain it is advisable to check that it is a good way of identifying computations in a much simpler physical system. This can be done by applying Chalmers' approach to a basic test system that has known computations. If the result matches the computations that we expect to be executing in the test system and is consistent with R1 and R2, then the same approach could potentially be used to identify computational correlates of consciousness in the brain.

The test system that I have selected for this purpose is a standard desktop computer running a robot control program. Since this is a paradigmatically computational system, we would expect that the CSA corresponding to the running program will be easy to identify using Chalmers' method. The use of a paradigmatically computational system as a test system also avoids contentious discussions about whether arbitrary physical systems, such as walls, rocks or buckets of water, can be said to implement a particular computation (Searle 1990; Copeland 1996). Most people would agree that a desktop computer running a particular program is a computational system and that it is implementing the computations listed in the program. A third reason for choosing this test system is that our theories of mind are often influenced by our beliefs about how computers work. These theories could be clarified and improved by developing a better understanding of the relationship between a physical computer, the programs running on the computer and the computations that the computer is executing.

The test system is presented in detail to support the discussion and to enable other people to suggest how it could be analyzed for computational correlates of consciousness. Although it is not capable of any of the functions that have been proposed to be linked to consciousness (for instance, the implementation of a global workspace (Baars 1988)), the issues raised in this paper are applicable to any program that could run on a computer. No claim is being made about whether or not the test system is actually conscious –it is just being used to illustrate how computations could be identified in a way that is consistent with the requirements of an experiment on the correlates of consciousness (R1 and R2).[7]

2.2 Desktop Computer, P_1 and the Simple Robot

The test system is a standard desktop computer that controls a simple two wheeled robot (see Figure 1). I have chosen a robotic system to address claims that embodiment is necessary for cognition and/or consciousness. The robot has a light sensor on top, and the motors driving the wheels can be independently controlled. This system is detailed enough to enable questions about the physical implementation of computations to be accurately posed, but hopefully simple enough so that the discussion does not get bogged down in details about drivers and internal communications.

The code for the control program, P_1, is as follows:

```
1. lightSensor = 0;
2. oldLightSensor = 0;
```

[7] Also requirement R3, which is set out in Section 3.1.

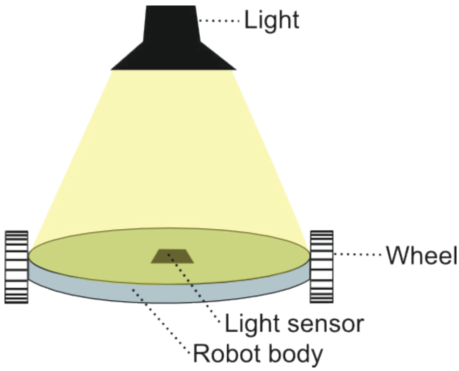

Figure 1. Wheeled robot

```
motivation = 2;
leftWheel = 1;
rightWheel = 1;

while(true){
    lightSensor = getSensorReading();

    if(lightSensor == oldLightSensor){
        if(motivation > 0)
            motivation = motivation - 1;
    }
    else{
        motivation = 2;
        oldLightSensor = lightSensor;
    }

    if(motivation > 0){
        leftWheel = 1;
        rightWheel = 1;
    }
    else{
        leftWheel = 1;
```

```
25.         rightWheel = 0;
26.    }
27.
28.     updateMotors(leftWheel, rightWheel);
29. }
```

P_1 runs as a single thread. The *getSensorReading()* function returns 1 if the light reading of the sensor is above a threshold, or 0 if it is below the threshold. The *oldLightSensor* variable stores the previous state of the light sensor and the *motivation* variable is decreased if the light does not change between cycles of the while loop. The *updateMotors()* function sets the state of the motors, using a device driver to communicate with the robot's hardware. Each motor runs when its variable is set to 1; the motors are off when their variables are set to 0.

Under the control of this program the robot will move forward in a straight line when the light is changing, or circle if the light stays the same. A stroboscope with double the while loop's frequency will be required to drive the robot forward. The robot would exhibit smoother behaviour if the *getSensorReading*() function returned a wider range of values. The robot would then move forward in response to a gradual increase or decrease in light level. I omitted this feature because it would substantially increase the size of the CSA without significantly altering the example.

2.3 CSA_1

The CSA description of the program, CSA_1, is illustrated in Figure 2. A first point to note about the mapping from P_1 to CSA_1 is that a state can make different transitions depending on where it occurs in the program. Suppose that the program variables are represented as the vector [*ls*, *ols*, *m*, *lw*, *rw*] and the call to the *getSensorReading()* function puts the system into state [1,1,1,1,1]. At line 10 of the program this state will lead to [1,1,0,1,1] followed by [1,1,0,1,0]. However, if the system is in state [1,1,1,1,1] at line 19 of the program, then it will stay in this state until *getSensorReading()* is called, which could leave the state as it is or cause it to transition to [0,1,1,1,1]. To take this into account in the mapping of P_1 onto CSA_1 we need an extra variable, not explicitly contained in the program, to fully specify the

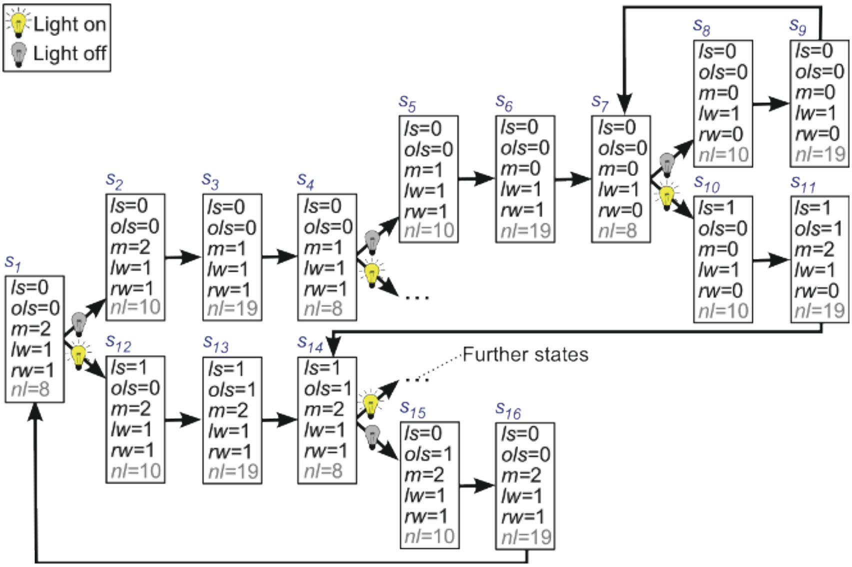

Figure 2. Extract from CSA_1, which corresponds to program P_1 (the full CSA has 144 states). This collapses some of the transitions and omits the communications with the light sensor and motors. The states have been labelled for convenience with s_1, s_2 ... s_{16}. Many transitions only depend on the current state; others are conditional on the state of the light sensor. The abbreviations of the program variables are as follows: *ls=lightSensor*; *ols=oldLightSensor*; *m=motivation*; *lw=leftWheel*; *rw=rightWheel*. *nl* was introduced to distinguish between identical states that make different transitions because they are at different points in the program (see discussion in text).

program logic. For convenience I have used the next program line to be executed, *nl*. I could also have added an extra variable to the program to enable it to be mapped without this ambiguity.

The CSA that corresponds to a program will in most cases be an *incomplete* description of the physical system, with the states and transitions of the program being a small subset of the possible transitions and states that the system is capable of. For example, a computer is typically capable of running many different programs, and so the CSA that corresponds to a particular program will not be a complete description of the states, substates and transitions of the computer. The CSA corresponding to a program also does not specify *how* the transitions between states are causally effected by the physical system. This enables the CSA to be mapped onto many different physical systems, which are considered to implement it in different ways.

3. Can CSA_1 be identified in the Test System a way that is Consistent with R1 and R2?

Since the test system is a paradigmatic example of a computational system, we would expect it to contain a clearly defined set of computations that could easily be identified with Chalmers' method. It also seems reasonable to expect that the CSA corresponding to the executing program could be unambiguously identified in the test system. After discussing the approach that I will use to analyze the test system in Section 3.1, I will highlight some of the problems that occur when we try to use Chalmers' method to analyze the test system for computations in a way that would be consistent with an experiment on the correlates of consciousness.

3.1 Method for Identifying CSAs in the Test System

The programs that are executing on the test system can be identified by launching the Task Manager in Windows or by typing "ps –el" in a Linux shell. If P_1 is on these lists, then P_1 is running on the computer. Some people might think that if P_1 is running on the computer, then CSA_1 is being executed on the computer. However, the running of a program and the implementation of a CSA are different things – one is a list of instructions and data held in memory (combined with a record of the currently executing instruction); the other is a set of causal relationships between substates of a physical system. While the list of programs *might* provide an accurate picture of the CSAs that are executing in a computer, this cannot be assumed at the outset. Furthermore, this simple method for identifying CSAs is inapplicable to the brain, which does not have a convenient interface that displays the running programs. Since the test system is intended to help us understand how CSAs can be identified in the brain, it is pointless to analyze it using a method that is inapplicable to the brain.

Instead, the CSAs in the test system will be identified by mapping what

Chalmers (2011, p. 328) calls the "distinct physical regions" of the system (if such distinct regions can be found) onto CSA substates. By observing how these regions change over time we can map out the state transitions and their conditional structure, which enables us to identify the CSAs that are executing in the system. This method for measuring CSAs is summarized as the following requirement:

> **R3.** *In an experiment to test H2, the executing CSAs will be identified by mapping the distinct regions of the physical system onto CSA substates. Measurements of these distinct regions over time will be used to infer the structure of the CSAs.*

This method can be used to infer the brain's CSAs from measurements of neuron voltages, blood flows, glia activity, and so on. The test system can be analyzed in a similar way by treating it as a physical object made of silicon, copper, plastic, etc., and mapping its distinct physical regions (for example, DRAM storage cells – see Section 3.3) onto CSA substates. While some facts about a computer's operation will be used to illustrate the CSA mapping problems, the analysis should not depend on prior knowledge about the operating system or variable encoding, and the person doing the analysis will not be able to connect up a keyboard and screen to get the computer's own 'view' of its internal states.

3.2 Virtual Memory and Cache

If the test system is implementing CSA_1, there should be distinct regions of the desktop computer that correspond to substates of CSA_1, such that the patterns of causal interaction between these regions lead to the transitions specified in CSA_1. Chalmers suggests that each substate could correspond to an independent element of the physical system, and so we might initially think that *ls*, *ols*, *m*, *lw*, *rw* and *nl* could be neatly mapped onto the areas of dynamic random access memory (DRAM) that hold the program variables *lightSensor*, *oldLightSensor*, *motivation*, *leftWheel* and *rightWheel*. The problem with this mapping is that the physical locations accessed by the CPU as it runs P_1 will vary depending on whether virtual memory is used and whether the data has been copied into the CPU cache. Depending on the cache algorithms and the amount of physical and virtual memory, the transitions between physical states in the desktop computer will have very little correspondence with CSA_1, and they will vary substantially between different runs of the program as different chunks of memory are copied into cache, other programs are run on the computer and virtual memory is swapped on and off the hard drive.

It might be thought that we could solve this problem by mapping substates of CSA_1 onto *sets* of physical states of the desktop computer. For example, we could map the substates onto tables listing the location of each variable at each point in time. However, these tables could only be constructed retrospectively, and they would vary with each run of the program because the use of cache and virtual memory depends on the other processes running on the system (a computer virus could suddenly use up a lot of memory, which would cause other processes to be swapped on and off the hard drive). This use of retrospective tables would completely undermine the mapping of physical states onto CSA substates, and it would also be vulnerable to the panpsychism of Putnam-style disjunctive mappings.

These problems linked to virtual memory and CPU caching are likely to defeat simple methods of interpreting the test system as an implementation of CSA_1. The state transitions of the DRAM that holds the variables will vary with each run of the program and the CSAs corresponding to different programs will overlap in complex ways. In principle it might be possible to untangle the entire operating system and the algorithms of the low level chips in order to track a piece of information through the system. But this untangling process would not preserve the causal and counterfactual relationships between the elements of the physical system that correspond to CSA substates. This does not prove that a desktop computer running P_1 is *not* implementing CSA_1 because CSA_1 might be buried somewhere in its physical states. But it does show that there is little or no connection between the programs that are running on a computer and the CSAs that it is implementing.

Since virtual memory and CPU caching are non-essential optimization strategies, they will be set aside for the rest of this paper and the test system will be revised so that P_1 is running on the simple computer shown

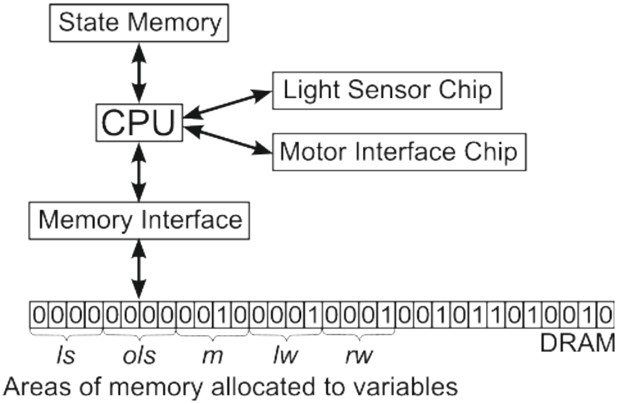

Figure 3. Simple computer. This computer uses dynamic random access memory (DRAM), whose storage cell voltages decay over time and have to be constantly refreshed. The values of 1 or 0 are obtained by applying a threshold to the storage cell voltages (see Section 3.3). The CPU uses device drivers to communicate with the light sensor and motor interface chips.

in Figure 3. This does not have virtual memory or cache and it can be considered to be an implementation of a random access Turing machine.

3.3 Mapping Physical States onto CSA Substates

The most obvious 'distinct physical regions' of the simple computer are the DRAM storage cells. Since the person analyzing the simple computer does not have high level information about the operating system (for example, whether it is 32- or 64-bit)[8] - there are no restrictions on how they should organize these cells into higher level groups. This leads to an extremely large number of equally valid ways of interpreting the simple computer's DRAM storage cells as substates of a CSA.

The voltage in a DRAM storage cell typically ranges from 0 to 1.5 V and changes continuously with time (typically decreasing between refresh

[8] Roughly speaking, a 32-bit operating system uses a maximum of thirty two 1s and 0s to represent a number or a memory address. A 64-bit operating system uses up to sixty four 1s and 0s to represent a number or a memory address.

cycles). To begin with I will consider the standard way of interpreting a DRAM voltage, which is to apply a threshold and interpret a voltage above the threshold as a '1' and a voltage below the threshold as a '0'. To make the discussion clear, I will use a simpler example than the test system: a C++ program, P_2, that calculates the Fibonacci series. At the end of each while loop the *fib* variable contains the next number in the sequence:

```
unsigned char fib = 0;
unsigned char nM1 = 1;
unsigned char nM2 = 0;

while(true){
     fib = nM2 + nM1;
     nM2 = nM1;
     nM1 = fib;
}
```

When P_2 is run on the simple computer we get the sequence of memory states shown in Figure 4a. If we interpret each DRAM storage cell as a CSA substate, we get the CSA shown in Figure 4b.

If we want to locate the CSA corresponding to P_2 within the memory states, we will have to relax the requirement that distinct regions of the physical system are mapped onto CSA substates, and allow *groups* of DRAM cells to be mapped onto CSA substates.[9] If the grouping of DRAM cells in Figure 4c is used, then we get the CSA illustrated in Figure 4d, which corresponds to the Fibonacci program. However, without the distinct regions requirement there are no limits to the ways in which we can interpret groupings of the DRAM storage cells as substates of a CSA. For example, the grouping of DRAM cells in Figure 4e leads to the CSA shown in Figure 4f.

All three CSAs in Figure 4 are equally valid mappings of physical states of the computer onto CSA substates, and the substates within each CSA

[9] This is because a DRAM cell only holds a single 1 or 0, whereas the P_2 variables are encoded using multiple 1s and 0s.

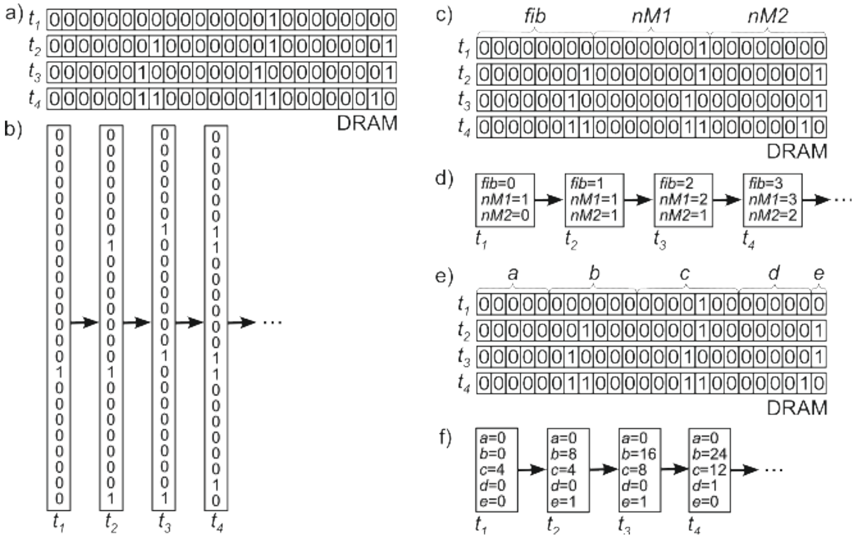

Figure 4. Different mappings of DRAM states onto CSA substates. a) A threshold is typically applied to the DRAM voltages: a voltage above the threshold is interpreted as a '1'; a voltage below the threshold is interpreted as a '0'. When this interpretation is applied to the DRAM of the simple computer as it runs P_2 we get a sequence of states, which is shown for times t_1-t_4. Each time step corresponds to an iteration of the while loop. b) The CSA that results from the mapping of individual DRAM cells onto CSA substates. c) Mapping of variables from the Fibonacci program onto groupings of DRAM storage cells. d) CSA corresponding to this interpretation of the DRAM, which matches the Fibonacci program. e) An alternative interpretation of the DRAM that maps its states onto variables *a*, *b*, *c*, *d* and *e*. f) CSA corresponding to the alternative interpretation of the DRAM.

are causally related to each other to the same extent. This is because the number of bits that are allocated to each number is not a 'distinct region' of the physical system, but a high level piece of information that is typically accessed through a graphical interface. There is no canonical way of mapping groups of DRAM cells onto CSA substates, and so when the simple computer is running P_2 it is simultaneously implementing as many CSAs as there are ways of partitioning its DRAM. The only objective fact of the matter is that at least one interpretation of the DRAM matches the CSA of the program that is running.

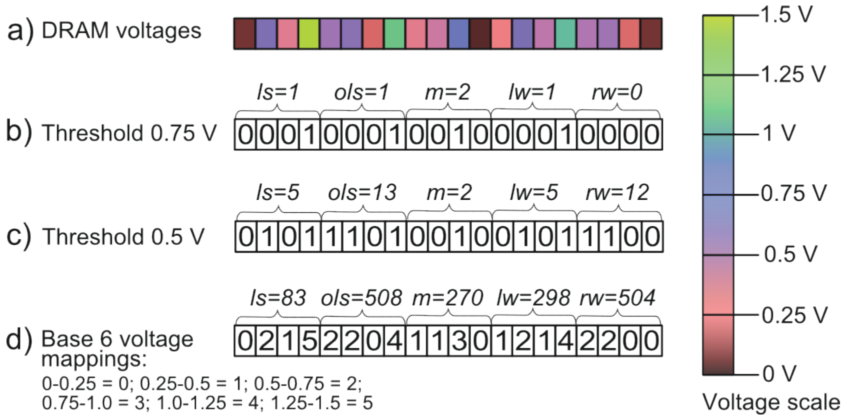

Figure 5. The effect of voltage thresholding on substate values. It is assumed that the DRAM storage cell voltages range from 0-1.5 V. a) Voltages in the DRAM storage cells at a particular point in time; their value is given by colour scale on the right. b) Values of the substates interpreted using a threshold of 0.75 V, so that a voltage >= 0.75 maps to 1 and a voltage < 0.75 maps to 0. c) Values of the substates interpreted using a threshold of 0.5 V, so that a voltage >= 0.5 maps to 1 and a voltage < 0.5 maps to 0. d) Voltage ranges are used to map the voltages onto numbers 0-5, resulting in a base 6 interpretation of the substate values.

So far I have assumed that the voltages are mapped onto 1s and 0s in the standard way. However, an external observer who was attempting to analyze this system for CSAs would not be constrained to apply any particular threshold to this voltage, and the mapping between voltage values and substate values could be carried out in many different ways – for instance, 1.5 V can be interpreted as a substate value of 1, 100 or -0.75. The sampling frequency and the way in which the system is divided into discrete spatial areas also have major effects on the substate values.[10] Some

[10] The application of an arbitrary voltage threshold could be avoided by mapping the DRAM voltages onto continuous substate values. This possibility is discussed by Chalmers (2011, p. 347-349). This would break the link between digital programs and the CSAs implemented by the physical computer and there would still be ambiguities about which parts or aspects of a physical system are measured to extract the analogue values (Gamez 2014).

of these problems are illustrated in Figure 5.

While Chalmers acknowledges that a system can implement more than one computation, he claims that this is not a problem as long as it does not implement every computation. However, to prove that a CSA is correlated with consciousness we have to show that it is executing when the system is conscious and *not* executing when it is unconscious (R2). This means that we will have to search through the effectively infinite number of ways of mapping DRAM states onto CSA substates to prove that a candidate CSA is not present when the system is unconscious. This vast search space is purely the result of how we define the mapping rules linking states of the simple computer to substates of the CSA - it has nothing to do with complex mapping functions that change arbitrarily over time or Putnam-style disjunctive mappings. With an infinity of parallel CSAs it is going to be impossible to prove H2 because we will not be able to demonstrate that a particular CSA is absent from the unconscious brain.

3.4 The Causal Structure of a Computer

Suppose that we ignore the problems raised in the previous section and map the substates of CSA_1 onto the areas of the simple computer's DRAM that actually hold the program variables, using standard bit allocation methods and voltage thresholding. Although the substates and state transitions of the simple computer will now follow the substates and state transitions of CSA_1, the causal topology specified in CSA_1 will not match the causal topology of the mapped system because the mapped DRAM areas are causally isolated from each other. Left to themselves these are incapable of producing a single state transition - at most the storage cell voltages will decay over time without the constant refresh required to maintain their states (see Figure 6a).

The state transitions in the simple computer depend on complex causal interactions between the DRAM voltages and a large number of substates in the CPU and the rest of the system (Figure 6b).[11] These electromagnetic

[11] The operations and instructions that are carried out by a CPU (AND, OR, etc.) are high level descriptions of the physical circuits, whose electromagnetic interactions cause the state transitions.

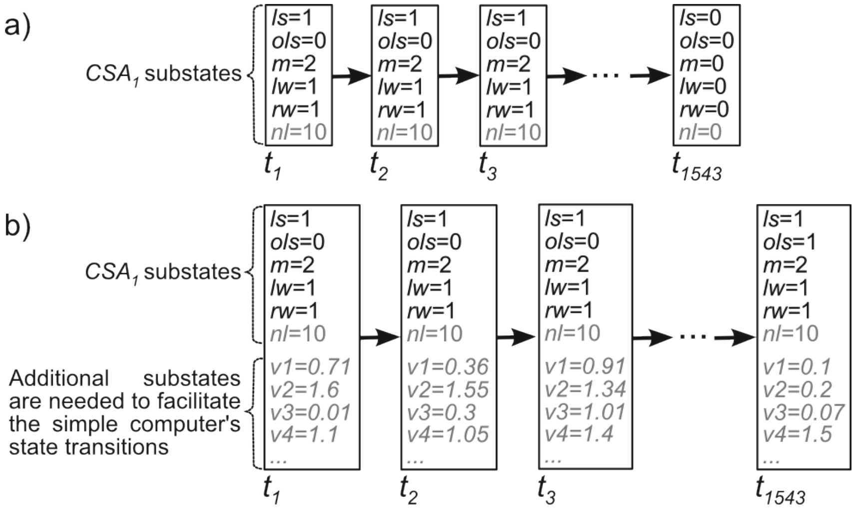

Figure 6. a) The simple computer's DRAM is mapped onto CSA_1 substates. These memory storage cell areas have no causal interactions with each other, and so they cannot bring about the CSA_1 state transitions by themselves. Without the voltage refresh they will rapidly decay to zero. b) In the simple computer a large number of other substates (mostly chip voltages) are required to causally facilitate the state transitions. These substates are not included in CSA_1, and they will vary widely with the architecture of the computer.

interactions between voltages in different parts of the chips are governed by complex laws, and the number, type and behaviour of the additional substates varies with the architecture of the computer. Since these additional substates are not specified in CSA_1, the DRAM areas holding the P_1 program variables will not be implementing CSA_1, but a completely different causal topology, which is required to make the program actually work.

This breaks the link between the program that is running on a computer and the CSAs that are present in the physical states of a computer. While it is possible that CSA_1 can be found somewhere inside the simple computer when it is running P_1, it is equally likely that CSA_1 will be executed when the simple computer is running a completely different program. The fact that a computer is running P_1 is not evidence for the claim that it is implementing CSA_1.

3.5 Counterfactuals

A real physical system is subject to influences from its environment that cause state transitions. While great care is taken to isolate computers' electronic components, they are still subject to external interference, which can cause them to make state transitions that are not specified in the program. A computer executing P_1 will therefore have a much richer set of connections between its states than is specified by CSA_1, which are mediated by unlikely but possible events. For example, *if* there is a nuclear test within a certain distance of the simple computer with a particular electromagnetic pulse size and intensity etc., *then* the simple computer will transition from s_1 to s_3. Some examples of these state transitions are given in Figure 7b.

The causal topology of a physical system at a particular point in time includes all of the possible state transitions that *could* be caused internally or in response to external events. While most events in the universe will not affect the state transitions of the simple computer, it seems reasonable to assume that there are enough possible effects of the environment on the fragile DRAM voltages that could link each state with *every* other state of the system. This complete connectivity between the states of the simple computer running P_1 is illustrated in Figure 7c.

When a computer's state transitions do not follow the program or operating system, we typically say that the program or computer has crashed or failed. However, the analysis described in this paper is solely based on an examination of a system's states (R3), and in this context there is no crashing or failing – just state transitions that are or are not conditional on external events. All of a system's possible state transitions count for its implementation of a CSA – any implications that these might have for the user's interaction with the computer are irrelevant.

This problem with a CSA-based approach to implementation has been considered by Chalmers (2012), who suggests that some form of normal background conditions might have to be included: "... the definition might require that there be a mapping M *and* conditions C [that currently obtain] such that for every formal state-transition rule $S_1 \rightarrow S_2$, if conditions C

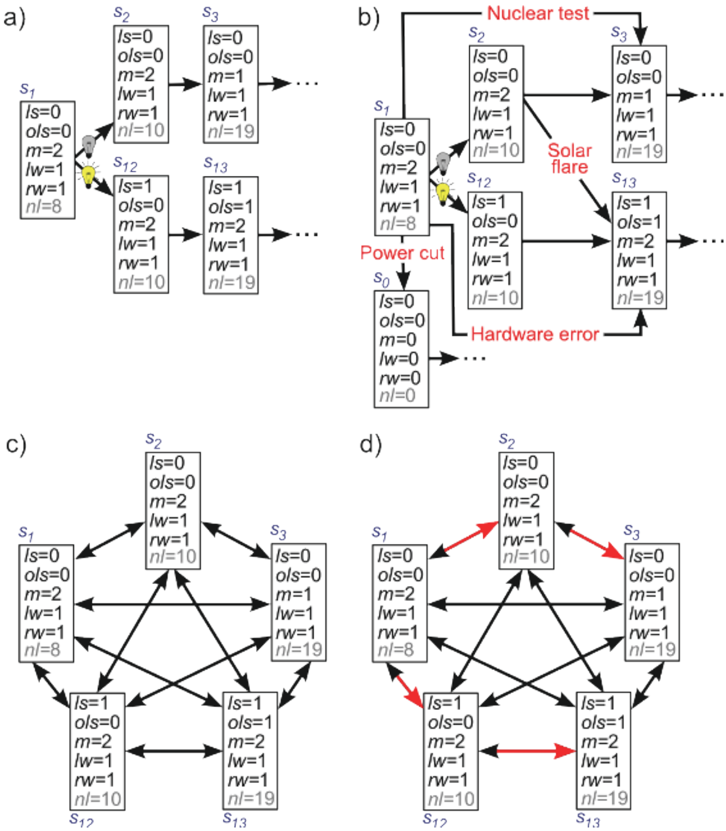

Figure 7. a) Extract from CSA_1. b) Some of the counterfactual state transitions that are implemented by the simple computer when it is running P_1. c) CSA describing the actual connections between s_1, s_2, s_3, s_{12} and s_{13}, which are conditional on unlikely but possible events, such as power failure or electromagnetic interference. d) The extract from CSA_1 (part 'a' of this figure, with the connections shown in red) can be identified within the fully connected CSA (part 'c' of this figure).

obtain and the system is in a total physical state that M maps to S_1, it transits to a total physical state that M maps to S_2." (p. 235). For example, although the simple computer running P_1 is in fact counterfactually and reliably sensitive to solar flares with particular parameters, the connection from s_2 to s_{13} can be ruled out because this is a low or zero probability event under normal conditions. One difficulty with this response is that normal background conditions would presumably have to be specified using an arbitrary probability threshold, which excludes abnormal events. This would violate R1 because the threshold would be set by the person carrying out the experiment, and different experimenters could use different thresholds. A second problem is that the normal conditions requirement makes the computations that are executing in a particular system dependent on events elsewhere in the universe that alter the probability of events, but might never interact with the system. At one point in time the probability of solar flares might be below the arbitrary threshold, there would be no connection between s_2 and s_{13} and the simple computer running P_1 would be judged to be implementing CSA_1. At a later time, events inside the sun might make solar flares more likely and the system will cease to implement CSA_1, even if no solar flares with the appropriate parameters have in fact occurred. If normal background conditions have to be included in an account of implementation, the probability of every event in the universe that could possibly affect a system will have to be taken into account to establish whether it is conscious or not.

If a plausible account of normal background conditions cannot be developed, then the only CSAs that are actually implemented by human-scale physical systems, such as brains and computers, are ones that have complete connectivity between their states. Human-scale physical systems cannot be completely isolated from low probability events, such as electrical interference or solar flares, that counterfactually link each state of a CSA to every other state. As systems get larger, the external factors that could cause state transitions will decrease, and the universe as a whole does not have a completely connected state structure because there is no possibility of outside interference.

It might still be claimed that a system is implementing CSA_1 because CSA_1 is *part* of the completely connected CSA (see Figure 7d). This is correct, but in this case CSA_1 cannot be a correlate of consciousness because the brain will have a completely connected CSA regardless of whether it is conscious or unconscious. Any CSA that can be found in the completely connected states of the conscious brain will also be present in the completely connected states of the unconscious brain.

3.6 Overlapping CSAs

Consider a simplified 'brain' (SB) that consists of four biological neurons, each of which can be in two states, firing or not firing. The complete state space has 16 different states, $s_1 - s_{16}$, with s_1 corresponding to state [0,0,0,0], s_2 corresponding to state [0,0,0,1], and so on (the exact mapping of substates

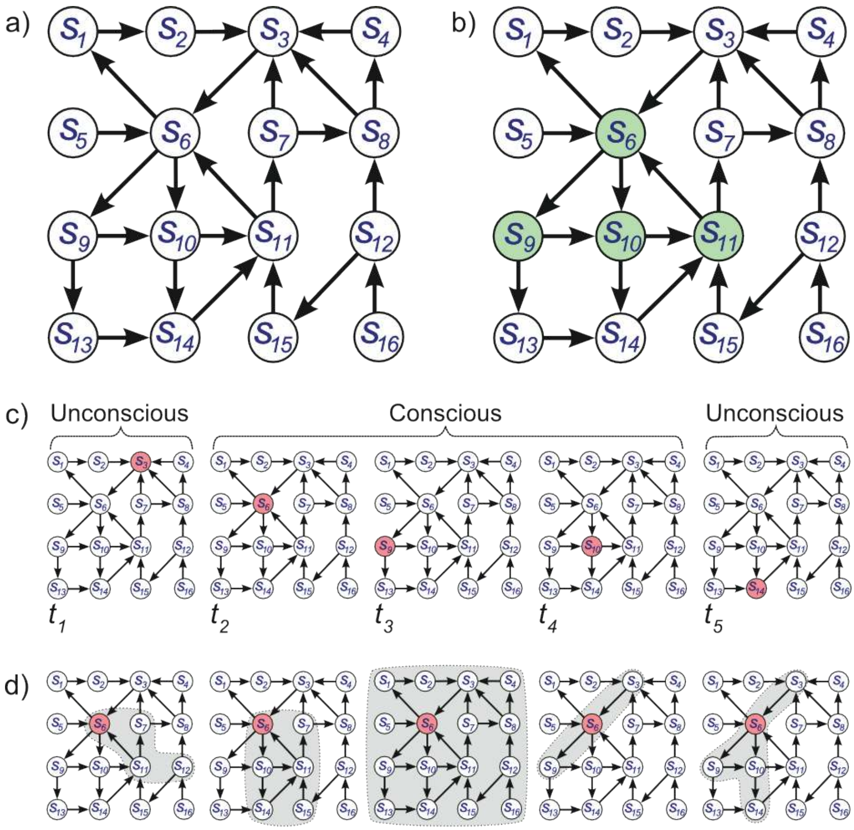

Figure 8. a) CSA describing complete state space of SB's four neurons. Although each state only has a single label, this is a CSA, not a FSA, because the states are constituted by the substates of four neurons. b) States that are active whenever SB is conscious. c) Over the time period t_1-t_5 SB moves through states s_3, s_6, s_9, s_{10} and s_{14}. It is conscious from t_2-t_4. d) The grey areas are examples of the large number of overlapping CSAs that can be considered to be executing when the system enters state s_6.

to labels is not important). Figure 8a shows an illustrative CSA for the SB in which the conditional branches can be thought to depend on a combination of input and internal dynamics. Let us suppose that states s_6, s_9, s_{10} and s_{11} are found to occur when SB is conscious, and that these states never occur when SB is unconscious (Figure 8b). During a particular execution run SB enters states s_3, s_6, s_9, s_{10} and s_{14}. It becomes conscious when it enters s_6 and ceases to be conscious when it enters state s_{14} (Figure 8c).

Suppose that SB enters state s_6. If we are looking for correlations between consciousness and CSAs, then we need to identify the CSAs that are executing at this point in time. Presumably the system is executing its complete CSA when it enters s_6, and it can also be considered to be executing any and all of the possible subsets of the state space (sub-CSAs) that include s_6 (examples are given in Figure 8d). It would violate R1 if we arbitrarily selected one of these overlapping CSAs and claimed that it was the *only* CSA that was executing. SB must therefore be considered to be executing a *set* of CSAs when it enters s_6, which suggests that we can only hope to identify correlations between sets of CSAs and consciousness.

It might be objected that in this example states s_6, s_9, s_{10} and s_{11} are correlated with consciousness, and so these states must form the sub-CSA, shown in Figure 8b, whose execution is correlated with consciousness. While there might be no fact of the matter about which sub-CSA is being executed when the system enters a particular state, Ockham's razor could be invoked to support the assumption that the sub-CSA consisting of states s_6, s_9, s_{10} and s_{11} is correlated with consciousness, even if only some of its states are entered during an execution run. According to this hypothesis, during the execution run shown in Figure 8c, the system would start off executing a sub-CSA that was not correlated with consciousness (for instance, one involving s_1, s_2, s_3 and s_4) or there would be no fact of the matter about which sub-CSA out of the overlapping sub-CSAs was being executed. At time t_2 it would start executing the sub-CSA associated with consciousness until at t_5 it would start executing a different sub-CSA not associated with consciousness, or there would cease to be a fact of the matter about which sub-CSA was being executed.

The problem with this claim is that a person who was monitoring the physical system would have no reason to believe that a particular sub-

CSA was being executed or that there was a transition between active sub-CSAs. Nothing in the physical system corresponds to this change - someone who only knew the physical facts could not identify or predict it. It is only because there is a change in the consciousness associated with the system that we are inclined to say that one sub-CSA is being executed instead of another. This suggests that a change in the executing sub-CSA is not a measureable property of the physical system, but is being attributed to the system to support a particular theory – namely that executing CSAs are linked to conscious states. A correlate of consciousness is a property of the physical system that is correlated with the presence of conscious states – we measure the physical system, measure consciousness and look for correlations between the two. Consciousness cannot be used to identify features of the physical system that are supposed to be correlated with consciousness – an independent measure of a physical property is required. The state and the set of CSAs that overlap a state might be considered to be objective facts about a system (if one could address the other problems with CSAs that have been raised in this paper). It is not a fact that a single CSA is being executed to the exclusion of all the other CSAs that overlap the current state.

In systems with more than a few elements an extremely large number of CSAs will overlap each state. This number increases factorially with the size of the system, which will rapidly make it impractical to record the CSAs that are correlated with consciousness. A second issue is that any small change to the system's state space will alter some of the CSAs that overlap a particular state. So even if was possible to exhaustively list the CSAs that were correlated with consciousness in one system, this list would become obsolete once the system changed its state space, which happens all the time in the brain. It would also be difficult or impossible to use the sets of CSAs that are correlated with consciousness in one system to make predictions about consciousness in other systems.

Given these problems it is more plausible and experimentally tractable to focus on correlations between particular states of a system and consciousness. In SB it is far simpler to enumerate the states that are correlated with consciousness (s_6, s_9, s_{10} and s_{11}) than to list the large number of CSAs that overlap these states. System states are more robust correlates that can be preserved across modifications of the state space, and it is more likely that two conscious organisms will have common states, rather than common sets of CSAs.

3.7 Patterns of Causation

Chalmers' account of implementation relies on causation to avoid problems with panpsychism and trivial implementations:

> While programs themselves are syntactic objects, implementations are not: they are real physical systems with complex causal organization, with real physical causation going on inside. In an electronic computer, for instance, circuits and voltages push each other around in a manner analogous to that in which neurons and activations push each other around. It is precisely in virtue of this causation that implementations may have cognitive and therefore semantic properties. (Chalmers 2011, p. 344)

If the target system matches the *causal* topology specified by the CSA, then it is supposed to instantiate the organizationally invariant properties that are associated with this causal topology, such as the property of having a mind and potentially consciousness. However, a major problem with Chalmers' use of causation is that it is far too weak and vague. This makes it easy to claim that one system has the same causal topology as another and is therefore implementing the same computation. If Chalmers' account of implementation is going to become a scientific hypothesis, the nature of a causal relationship has to be spelled out in enough detail to enable it to be accurately measured in a physical system.

Chalmers' vague handling of causation is linked to his use of CSAs to specify causal topologies: "...the CSA formalism provides a perfect formalization of the notion of causal topology. A CSA description specifies a division of a system into parts, a space of states for each part, and a pattern of interaction between these states This is precisely what is constitutive of causal topology." (Chalmers 2011, p. 341). While a CSA does specify a division of a system into parts and a space of states for each part, it does not *specify* the pattern of interaction between the parts that lead to the state transitions – it merely records the brute fact that the parts or substates of the system have one set of values at one time and at an arbitrary amount of time later the substates have a different set of values. This makes it highly questionable whether the CSA formalism provides any account of *causal* topology at all. A specification of the amount and pattern of causal relationships between the parts of a system would need to be added to the CSA formalism to enable it to describe a causal topology.

Within the literature on causation, there is a useful distinction between a *conceptual* analysis of causation, which elucidates how we understand and use causal concepts in our everyday speech, and an *empirical* account of causation, which attempts to explain how causation actually operates in the physical world (Dowe 2000). Predominantly conceptual accounts of causation include Lewis' counterfactual analysis (Lewis 1973) and Mackie's INUS conditions (Mackie 1993). The most developed empirical account of causation has been set out by Dowe (2000), who defines a causal interaction as follows:

- A *conserved quantity* is a quantity governed by a conservation law, such as mass-energy, momentum or charge.
- A *causal process* is a world line of an object that possesses a conserved quantity.
- A *causal interaction* is an intersection of world lines that involves the exchange of a conserved quantity.

While conceptual accounts of causation are unlikely to lead to an agreed fact of the matter about the causal topology that is implemented by a particular physical system (R1), empirical accounts do enable causal topologies to be precisely specified. They also support the identification of supervenient causal relationships at different levels of a system - for example, an empirical account of causation can easily deal with the fact that causal relationships between neuron voltages supervene on ions moving across a cell membrane. This suggests that Dowe's approach could be used to provide a detailed specification of the amount and type of causal interactions in a system - defining substates as causal processes and linking state transitions to the exchange of conserved quantities, such as mass-energy or momentum. While this would go a long way towards tightening up Chalmers' account of implementation, it would also virtually eliminate the idea that one system can implement the causal topology of another. Systems with different architectures and materials have very different patterns of exchange of conserved quantities. A modern digital computer, the ENIAC and Babbage's Analytical Engine that are running the same program will not be implementing the same computations, if this more causally accurate theory of implementation is used to identify the computations. All of the generality of Chalmers' account will have been lost.

4. Discussion and Conclusions

This paper has asked whether the execution of a computation could be a sole member of a CC set (H1). In other words, could the execution of a computation be correlated with consciousness independently of the architecture of the system or the material in which it is executing? I have suggested that this question could be addressed experimentally by identifying the computations that are executing in the conscious and unconscious brain. This requires a way of measuring computations in the brain and Chalmers' CSA approach was selected as the best method that is currently available for this purpose. H1 was then rephrased as the question about whether a CSA could be a sole member of a CC set (H2). To answer this question I investigated whether CSAs could be unambiguously identified in a computer in a way that would be consistent with an experiment on the correlates of consciousness (R1-R3). The following problems were identified:

1. Virtual memory and cache (Section 3.2) and the causal structure of a computer (Section 3.4) break the link between the programs running in a computer and the CSAs that are present. There is little or no relationship between the running programs and the CSAs that are being executed.
2. There are an effectively infinite number of different ways in which a physical system can be divided into parts that are mapped

onto CSA substates (Section 3.3). This will make it impossible to experimentally prove that a CSA which is present in the conscious brain is not present in the unconscious brain (R2).

3. Low probability counterfactuals link all states of a human-scale physical system, leading to complete connectivity of the CSA describing a system's states (Section 3.5). This suggests that CSAs cannot be correlated with consciousness because there will be identical CSAs in the conscious and unconscious brain.
4. When a system enters a particular state it is executing all of the CSAs that include this state (Section 3.6) - it is not possible to pick out a single CSA and claim that a system is executing this CSA to the exclusion of all others. This means that we can only look for correlations between sets of CSAs and consciousness. Such correlations will be impossible to specify on larger systems and they will not generalize easily.
5. When Chalmers' vague notion of causation is cashed out in a physically plausible way using an empirically grounded theory, it becomes much less likely that different computers running the same program will have the same causal topology (Section 3.7).

These problems have been illustrated on some simple systems and they will be many orders of magnitude harder on the brain, which has vague 'distinct physical regions' and an extremely large state space that changes all the time. The issues can be separated into theoretical problems (points 3 and 5) and pragmatic difficulties (points 2 and 4). The theoretical problems suggest that CSAs cannot be used to identify computational correlates of consciousness in the brain, even in principle. If the theoretical problems could somehow be addressed, the pragmatic difficulties are likely to prevent us from ever using CSAs to test H1.

The most obvious way of addressing these problems would be to develop a better account of implementation. Chalmers' theory was not designed for experimental work on the correlates of consciousness, and so it is not surprising that it is does not perform well in this context. Some of the difficulties with the CSA approach are linked to its abstractness and generality, and so it might be possible to fix it by specifying causal topologies in more detail. However, if the CSA account is supplemented with more details about the patterns of exchange of physically conserved quantities, then it will become a description of a physical correlate of consciousness, because it is unlikely that a detailed pattern of exchanges could implemented in different physical systems. It might be thought that the problems of complete connectivity highlighted in Section 3.5 could be fixed by specifying probabilities for each state transition. However, this is unlikely to work in the brain where the state transitions change all the time as the brain learns and shifts between tasks. There does not appear to be an easy way of adding detail to the CSA account that could address the problems that have been identified.

A second way of tackling these problems would be to define computations in terms of their inputs and outputs. For example, an adding function is defined by the fact that it returns the sum of a set of numbers – the details of its implementation do not matter and it would be superfluous to specify it using a CSA. The problem with this approach is that we often exhibit very little external behaviour when we are conscious – for instance, when I sit quietly in an armchair with my eyes closed there is no obvious external behaviour that could be used to identify computations that could be correlated with my consciousness. We might divide the brain up into modules and attempt to specify the computations that are carried out by each module by using their input-output relationships. But many of the computations linked to consciousness are likely to be impossible to partition in this way. Some functions that might be correlated with consciousness are implemented by highly modular brain areas – for example, V5 is linked to our visual experience of motion. But others, such as a global workspace (Baars 1988), are likely to be dynamically implemented through coordinated interactions between many areas of the brain. Since there are an infinite number of ways of dividing up the brain into modules, it is going be very difficult to develop a workable method for defining computations based on the external behaviour of brain areas.

It might be claimed that CSA_1 does not fully capture the computations in P_1 because it does not explicitly include the logical operations that are specified at a high level in the program using 'if', 'else', 'while', etc., which are mapped down to more basic computational operations by a compiler.

To address this concern, a much larger CSA could be constructed that specifies the causal topology that actually occurs in the CPU and memory of the simple computer as it runs P_1, which would include the causal structure of the implementation of the low level computational operations. Let us suppose that we do this and produce CSA_2, which describes the causal topology of the simple computer's chips at the nanometre scale as it runs P_1. In this case, it could more plausibly be claimed that the simple computer is implementing CSA_2 when it runs P_1, and that any other system that implements CSA_2 will implement the same computations as the simple computer running P_1. The problem with this response is that CSA_2 is specific to one particular type of computer. It will not be implemented by computers that use different chips to implement their logical operations or by programmable computers with completely different architectures, such as Babbage's Analytical Engine. In the very best case, this approach could show that identical computers running identical programs are implementing the same computations. However, it would remove one of the key attractions of CSAs, which were supposed to be general enough to identify the same computations in different systems. Furthermore, if CSAs have to be specified at this level of detail, then there will be little or no resemblance between the CSAs that are implemented by brains and computers. This would suggest that brains are not executing computations, or it could be interpreted to show that CSAs cannot be used to identify computational correlates of consciousness in the brain.

The disconnect between a computer's running programs and the CSAs that it is implementing has implications for the suggestion that we might eventually be able to upload our brain onto a computer or replace part of our brain with a functionally equivalent chip (Moor 1988; Chalmers 1995; Kurzweil 1999; Chalmers 2010). Suppose we manage to identify a CSA in the brain that is correlated with consciousness, write a program that implements this CSA, and run it on a computer. The arguments in sections 3.2 and 3.4 suggest that this system is unlikely to have the same consciousness as the original brain because the CSAs that are present in the computer are unlikely to correspond to the CSAs of the running programs. A computer running a program that simulates my brain is unlikely to be implementing the CSAs that are present in my brain. A chip that replicates the input-output functionality of part of my brain (for example, a silicon hippocampus) is unlikely to be implementing the CSAs of the original brain area. It might be possible to design a different type of 'computer' to implement CSAs, but this would require a very different architecture from a standard computer. For example, neuromorphic chips (Indiveri et al. 2011) that use the flow of electrons in silicon circuits to model the flow of ions in neurons might be capable of implementing some of the CSAs that are present in biological neurons.

Many of the issues with experiments on the computational correlates of consciousness will be encountered by experiments on the functional correlates of consciousness. To prove that there are functional correlates of consciousness we need to measure the functions that are executing in the brain, so we can show that certain functions are only present when the brain is conscious. The problems identified in this paper suggest that CSAs will not be a workable method for specifying and identifying functions, and it is not obvious how else one could measure functions during this type of experiment.

The theoretical and pragmatic problems raised in Section 3 suggest that H2 cannot be experimentally tested. While H2 might be a valid philosophical or metaphysical theory about the world, if it cannot be falsified, it is not a scientific hypothesis (Popper 2002). H1 cannot be tested until a plausible method for measuring computations has been found. Many of the problems with Chalmers' approach are likely to be encountered by other methods of measuring computations, but it will not necessarily be impossible to develop a method that can circumvent these difficulties and meet the requirements of an experiment on the correlates of consciousness (R1-R3). Until such a method has been found, we are likely to make more progress by focusing on patterns in particular physical structures that could be correlated with consciousness.[12]

[12] It might be possible to use CSAs to describe the behaviour of one part or aspect of a physical system. This would not be a computational correlate of consciousness in the sense of H1 because claims about consciousness based on this physical correlate could not be extended to other systems that exhibited the same CSA in a different substrate. There was not space in this paper to explore this application of CSAs in more detail.

References

Baars, B. J. (1988). *A Cognitive Theory of Consciousness.* Cambridge; New York, Cambridge University Press.

Bishop, J. M. (2002). Counterfactuals Cannot Count: A Rejoinder to David Chalmers. *Consciousness and Cognition* 11(4): 642-652.

Bishop, J. M. (2009). A Cognitive Computation Fallacy? Cognition, Computations and Panpsychism. *Cognitive Computation* 1: 221-233.

Brown, C. (2012). Combinatorial-State Automata and Models of Computation. *Journal of Cognitive Science* 13(1): 51-73.

Chalmers, D. (1995). Absent Qualia, Fading Qualia, Dancing Qualia. In *Conscious Experience*, edited by T. Metzinger. Thorverton, Imprint Academic: 309-328.

Chalmers, D. (1996). Does a Rock Implement Every Finite-State Automaton? *Synthese* 108: 309-333.

Chalmers, D. (2000). What Is a Neural Correlate of Consciousness? In *Neural Correlates of Consciousness*, edited by T. Metzinger. Cambridge, Massachusetts, MIT Press: 17-39.

Chalmers, D. (2010). The Singularity: A Philosophical Analysis. *Journal of Consciousness Studies* 17: 7-65.

Chalmers, D. (2011). A Computational Foundation for the Study of Cognition. *Journal of Cognitive Science* 12(4): 323-357.

Chalmers, D. (2012). The Varieties of Computation: A Reply. *Journal of Cognitive Science* 13(3): 211-248.

Chrisley, R. (1995). Why Everything Doesn't Realize Every Computation. *Minds and Machines* 4: 403-420.

Cleeremans, A. (2005). Computational Correlates of Consciousness. *Prog Brain Res* 150: 81-98.

Copeland, B. J. (1996). What Is Computation? *Synthese* 108: 335-359.

Dehaene, S. and Changeux, J. P. (2011). Experimental and Theoretical Approaches to Conscious Processing. *Neuron* 70(2): 200-227.

Dehaene, S., Naccache, L., Cohen, L., Bihan, D. L., Mangin, J. F., Poline, J. B. and Riviere, D. (2001). Cerebral Mechanisms of Word Masking and Unconscious Repetition Priming. *Nat Neurosci* 4(7): 752-758.

Dowe, P. (2000). *Physical Causation.* Cambridge, Cambridge University Press.

Egan, F. (2012). Metaphysics and Computational Cognitive Science: Let's Not Let the Tail Wag the Dog. *Journal of Cognitive Science* 13(1): 39-49.

Gamez, D. (2012). Empirically Grounded Claims About Consciousness in Computers. *International Journal of Machine Consciousness* 4(2): 421-438.

Gamez, D. (2014). Are Information or Data Patterns Correlated with Consciousness? *Topoi*. 10.1007/s11245-014-9246-7.

Harnad, S. (2012). The Causal Topography of Cognition. *Journal of Cognitive Science* 13(2): 181-196.

Indiveri, G., Linares-Barranco, B., Hamilton, T. J., van Schaik, A., Etienne-Cummings, R., Delbruck, T., Liu, S. C., Dudek, P., Hafliger, P., Renaud, S., Schemmel, J., Cauwenberghs, G., Arthur, J., Hynna, K., Folowosele, F., Saighi, S., Serrano-Gotarredona, T., Wijekoon, J., Wang, Y. and Boahen, K. (2011). Neuromorphic Silicon Neuron Circuits. *Front Neurosci* 5: 73.

Klein, C. (2012). Two Paradigms for Individuating Implementations. *Journal of Cognitive Science* 13(2): 167-179.

Koch, C. (2004). *The Quest for Consciousness: A Neurobiological Approach.* Englewood, Roberts & Company.

Kurzweil, R. (1999). *The Age of Spiritual Machines.* New York, Viking.

Lewis, D. (1973). Causation. *Journal of Philosophy* 70(17): 556-567.

Logothetis, N. K. (1998). Single Units and Conscious Vision. *Philos Trans R Soc Lond B Biol Sci* 353(1377): 1801-1818.

Mackie, J., L. (1993). Causes and Conditions. In *Causation*, edited by E. Sosa and M. Tooley. Oxford, Oxford University Press: 33-55.

Miłkowski, M. (2011). Beyond Formal Structure: A Mechanistic Perspective on Computation and Implementation. *Journal of Cognitive Science* 12(4): 359-379.

Moor, J. H. (1988). Testing Robots for Qualia. In *Perspectives on Mind*, edited by H. R. Otto and J. A. Tuedio. Dordrecht/Boston/Lancaster/Tokyo, D. Reidel Publishing Company.

Piccinini, G. (2007). Computing Mechanisms. *Philosophy of Science* 74: 501-526.

Popper, K. R. (2002). *The Logic of Scientific Discovery.* London, Routledge.

Putnam, H. (1988). *Representation and Reality.* Cambridge, Mass.; London, MIT Press.

Rees, G., Kreiman, G. and Koch, C. (2002). Neural Correlates of Consciousness in Humans. *Nature Reviews Neuroscience* 3(4): 261-270.

Rescorla, M. (2012). How to Integrate Representation into Computational Modeling, and Why We Should. *Journal of Cognitive Science* 13(1): 1-38.

Ritchie, B. (2011). Chalmers on Implementation and Computational Sufficiency. *Journal of Cognitive Science* 12(4): 401-417.

Scheutz, M. (2001). Computational Versus Causal Complexity. *Minds and Machines* 11(4): 543-566.

Scheutz, M. (2012). What It Is Not to Implement a Computation: A Critical Analysis of Chalmers' Notion of Implemention. *Journal of Cognitive Science* 13(1): 75-106.

Searle, J. R. (1990). Is the Brain a Digital Computer? *Proceedings and Addresses of the American Philosophical Association* 64: 21-37.

Shagrir, O. (2012). Can a Brain Possess Two Minds? *Journal of Cognitive Science* 13(2): 145-165.

Sprevak, M. (2012). Three Challenges to Chalmers on Computational Implementation. *Journal of Cognitive Science* 13(2): 107-143.

Teasdale, G. and Jennett, B. (1974). Assessment of Coma and Impaired Consciousness. A Practical Scale. *Lancet* 2(7872): 81-84.

Tononi, G. (2004). An Information Integration Theory of Consciousness. *BMC Neurosci* 5: 42.

Tononi, G. (2008). Consciousness as Integrated Information: A Provisional Manifesto. *Biological Bulletin* 215(3): 216-242.

Tononi, G. and Koch, C. (2008). The Neural Correlates of Consciousness: An Update. *Annals of the New York Academy of Sciences* 1124: 239-261.