Cogn Comput (2009) 1:221–233
DOI 10.1007/s12559-009-9019-6

A Cognitive Computation Fallacy? Cognition, Computations and Panpsychism

John Mark Bishop

Published online: 30 May 2009

Abstract The journal of Cognitive Computation is defined in part by the notion that biologically inspired computational accounts are at the heart of cognitive processes in both natural and artificial systems. Many studies of various important aspects of cognition (memory, observational learning, decision making, reward prediction learning, attention control, etc.) have been made by modelling the various experimental results using ever-more sophisticated computer programs. In this manner progressive inroads have been made into gaining a better understanding of the many components of cognition. Concomitantly in both science and science fiction the hope is periodically re-ignited that a man-made system can be engineered to be fully cognitive and conscious purely in virtue of its execution of an appropriate computer program. However, whilst the usefulness of the computational metaphor in many areas of psychology and neuroscience is clear, it has not gone unchallenged and in this article I will review a group of philosophical arguments that suggest either such unequivocal optimism in computationalism is misplaced—computation is neither necessary nor sufficient for cognition—or panpsychism (the belief that the physical universe is fundamentally composed of elements each of which is conscious) is true. I conclude by highlighting an alternative metaphor for cognitive processes based on communication and interaction.

Keywords Computationalism · Machine consciousness · Panpsychism

J. M. Bishop (✉)
Department of Computing, Goldsmiths, University of London, London, UK
e-mail: m.bishop@gold.ac.uk

Introduction

Over the hundred years since the publication of James' psychology [1] neuroscientists have attempted to define the fundamental features of the brain and its information-processing capabilities in terms of (i) mean firing rates at points in the brain cortex (neurons) and (ii) computations; today the prevailing view in neuroscience is that neurons can be considered fundamentally computational devices. In operation, such computationally defined neurons effectively sum up their input and compute a complex non-linear function on this value; output information being encoded in the mean firing rate of neurons, which in turn exhibit narrow functional specialisation. After Hubel and Wiesel [2] this view of the neuron as a specialised feature detector has become treated as established doctrine. Furthermore, it has been shown that richly interconnected networks of such neurons can 'learn' by suitably adjusting the inter-neuron connection weights according to complex computationally defined processes. In the literature there exist numerous examples of such learning rules and architectures, more or less inspired by varying degrees of biological plausibility; early models include [3–6]. From this followed the functional specialization paradigm, mapping different areas of the brain to specific cognitive functions.

In this article I suggest that this attraction to viewing the neuron merely as a computational device fundamentally stems from (i) the implicit adoption of a computational theory of mind (CTM) [7]; (ii) a concomitant functionalism with respect to the instantiation of cognitive processes [8, 9] and (iii) an implicit non-reductive functionalism with respect to consciousness [10]. Conversely, I suggest that a computational description of brain operations has difficulty in providing a physicalist account of several key features of

human cognitive systems (in particular phenomenal consciousness and 'understanding') and hence that computations are neither necessary nor sufficient for cognition; that any computational description of brain processes is thus best understood merely in a metaphorical sense. I conclude by answering the question *What is cognition if not computation?* by tentatively highlighting an alternative metaphor, defined by physically grounded processes of communication and interaction, which is less vulnerable to the three classical criticisms of computationalism described herein.[1]

The CTM

The CTM occupies one part of the spectrum of representational theories of mind (RTM). Although currently undergoing challenges from dynamic systems, embodied, enactivist and constructivist accounts of cognition (e.g. [13–19]), the RTM remains ubiquitous in contemporary cognitive science and experimental psychology. Contrary to naive or direct realism, indirect realism (or representationalism) postulates the actual existence of mental intermediaries—representations—between the observing subject and the world. The earliest forms of RTM can be traced to Descartes [20] who held that all thought was representational[2] and that it is the very nature of mind (res cogitans) to represent the world (res extensa).

Harnish [7] observes that the RTM entails:

- Cognitive states are relations to mental representations which have content.
- A cognitive state is:
  - A state [of mind] denoting knowledge; understanding; beliefs, etc.
  - *This definition subsequently broadened to include knowledge of raw sensations, colours, pains, etc.*
- Cognitive processes—changes in cognitive states—are mental operations on these representations.

The Emergence of Functionalism

The CTM came to the fore after the development of the stored program digital computer in the mid-20th century when, through machine-state functionalism, Putnam [8, 9] first embedded the RTM in a computational framework. At the time Putnam famously held that:

- Turing machines (TMs) are multiply realisable on different hardware.
- Psychological states are multiply realisable in different organisms.
- Psychological states are functionally specified.

Putnam's 1967 conclusion is that the best explanation of the joint multiple realisability of TMs and psychological states[3] is that TMs specify the relevant functional states and so specify the psychological states of the organism; hence by this observation Putnam makes the move from 'the intelligence of computation to the computational theory of intelligence' [7]. Today variations on CTM structure the most commonly held philosophical scaffolds for cognitive science and psychology (e.g. providing the implicit foundations of evolutionary approaches to psychology and linguistics). Formally stated the CTM entails:

- Cognitive states are computational relations to computational representations which have content.
- A cognitive state is a state [of mind] denoting knowledge; understanding; beliefs, etc.
- Cognitive processes—changes in cognitive states—are computational operations on these computational representations.

The Problem of Consciousness

The term 'consciousness' can imply many things to many different people. In the context of this article I refer specifically to that aspect of consciousness Ned Block terms 'phenomenal consciousness' [21], by which I refer to the first person, subjective phenomenal states—sensory tickles, pains, visual experiences and so on.

Cartesian theories of cognition can be broken down into what Chalmers [10] calls the 'easy' problem of perception—the classification and identification of sense stimuli—and a corresponding 'hard' problem, which is the realization of the associated phenomenal state. The

[1] In two earlier articles (with Nasuto et al. [11, 12]) the author explored theoretical limitations of the computational metaphor from positions grounded in psychology and neuroscience; this article—outlining a third perspective—reviews three philosophical critiques of the computational metaphor with respect to 'hard' questions of cognition related to consciousness and understanding. Its negative conclusion is that computation is neither necessary nor sufficient for cognition; its positive conclusion suggests that the adoption of a new metaphor may be helpful in addressing hard conceptual questions related to consciousness and understanding. Drawing on the conclusions of the two earlier articles, the suggested new metaphor is one grounding cognition in processes of communication and interaction rather than computation. An analogy is with the application of Newtonian physics and Quantum physics—both useful descriptions of the world, but descriptions that are most appropriate in addressing different types of questions.

[2] Controversy remains surrounding Descartes' account of the representational content of non-intellectual thought such as pain.

[3] Although Putnam talks about pain not cognition, it is clear that his argument is intended to be general.

difference between the easy and the hard problems and an apparent lack of the link between theories of the former and an account of the latter has been termed the 'explanatory-gap'.

The idea that the appropriately programmed computer really is a mind, and was eloquently suggested by Chalmers (ibid). Central to Chalmers' non-reductive functionalist theory of mind is the Principle of Organizational Invariance (POI). This asserts that, "given any system that has conscious experiences, then any system that has the same *fine-grained functional organization* will have qualitatively identical experiences".

To illustrate the point Chalmers imagines a fine-grained simulation of the operation of the human brain—a massively complex and detailed artificial neural network. If, at a very fine-grained level, each group of simulated neurons was functionally identical to its counterpart in the real brain then, via Dancing Qualia and Fading Qualia arguments, Chalmers (ibid) argues that the computational neural network must have precisely the same qualitative conscious experiences as the real human brain.

Current research into perception and neuro-physiology certainly suggests that physically identical brains will instantiate identical phenomenal states and, although as Maudlin [22] observes this thesis is not analytic, something like it underpins computational theories of mind. For if computational functional structure supervenes on physical structure then physically identical brains must be computationally and functionally identical. Thus Maudlin formulates the *Supervenience Thesis* (ibid) "... two physical systems engaged in precisely the same physical activity through a time will support precisely the same modes of consciousness (if any) through that time".

The Problem of Computation

It is a commonly held view that 'there is a crucial barrier between computer models of minds and real minds: the barrier of consciousness' and thus that 'information-processing' and 'phenomenal (conscious) experiences' are conceptually distinct [23]. But is consciousness a prerequisite for genuine cognition and the realisation of mental states? Certainly Searle believes so, "... the study of the mind is the study of consciousness, in much the same sense that biology is the study of life" [24] and this observation leads him to postulate the *Connection Principle* whereby "... any mental state must be, at least in principle, capable of being brought to conscious awareness" (ibid). Hence, if computational machines are not capable of enjoying consciousness, they are incapable of carrying genuine mental states and computation fails as an adequate metaphor for cognition.

In the following sections I briefly review two well-known arguments targeting computational accounts of cognition from Penrose and Searle, which together suggest computations are neither necessary nor sufficient for mind. I subsequently outline a simple *reductio ad absurdum* argument that suggests there may be equally serious problems in granting phenomenal (conscious) experience to systems purely in virtue of their execution of particular programs; if correct, this argument suggests either strong computational accounts of consciousness must fail or that panpsychism is true.

Computations and Understanding: Gödelian Arguments Against Computationalism

Gödel's first incompleteness theorem states that *"... any effectively generated theory capable of expressing elementary arithmetic cannot be both consistent and complete. In particular, for any consistent, effectively generated formal theory F that proves certain basic arithmetic truths, there is an arithmetical statement that is true, but not provable in the theory."* The resulting true but unprovable statement $G(\check{g})$ is often referred to as 'the Gödel sentence' for the theory (albeit there are infinitely many other statements in the theory that share with the Gödel sentence the property of being true but not provable from the theory).

Arguments based on Gödel's first incompleteness theorem—initially from Lucas [25, 26] were first criticised by Benacerraf [27] and subsequently extended, developed and widely popularised by Penrose [28–31]—typically endeavour to show that for any such formal system F, humans can find the Gödel sentence $G(\check{g})$ whilst the computation/machine (being itself bound by F) cannot. In [29] Penrose develops a subtle reformulation of the vanilla argument that purports to show that "the human mathematician can 'see' that the Gödel Sentence is true for consistent F even though the consistent F cannot prove $G(\check{g})$''.

A detailed discussion of Penrose's formulation of the Gödelian argument is outside the scope of this article (for a critical introduction see [32, 33] and for Penrose's response see [31]); here it is simply important to note that although Gödelian-style arguments purporting to show 'computations are not necessary for cognition' have been extensively[4] and vociferously critiqued in the literature (see [34] for a review), interest in them—both positive and negative—still regularly continues to surface (e.g. [35, 36]).

[4] For example, Lucas maintains a web page http://users.ox.ac.uk/~jrlucas/Godel/referenc.html listing more than 50 such criticisms.

The Chinese Room Argument

One of the most widely known critics of computational theories of mind is John Searle. His best-known work on machine understanding, first presented in the 1980 paper 'Minds, Brains & Programs' [37], has become known as the Chinese Room Argument (CRA). The central claim of the CRA is that computations alone are not sufficient to give rise to cognitive states, and hence that computational theories of mind cannot fully explain human cognition. More formally Searle stated that the CRA was an attempt to prove the truth of the premise:

– Syntax is not sufficient for semantics.

Which, together with the following two axioms:
– (i) Programs are formal (syntactical).
– (ii) Minds have semantics (mental content).

... led Searle to conclude that:

– Programs are not minds.

... and hence that *computationalism*—the idea that the essence of thinking lies in computational processes and that such processes thereby underlie and explain conscious thinking—*is false* [38].

In the CRA Searle emphasises the distinction between syntax and semantics to argue that while computers can act in accordance to formal rules, they cannot be said to know the meaning of the symbols they are manipulating, and hence cannot be credited with genuinely understanding the results of the execution of programs those symbols compose. In short, Searle claims that while *cognitive computations* may *simulate* aspects of cognition, they can never *instantiate* it.

The CRA describes a situation where a monoglot Searle is locked in a room and presented with a large batch of papers covered with Chinese writing that he does not understand. Indeed, Searle does not even recognise the writing as Chinese ideograms, as distinct from say Japanese or simply meaningless patterns. A little later Searle is given a second batch of Chinese symbols together with a set of rules (in English) that describe an effective method (algorithm) for correlating the second batch with the first purely by their form or shape. Finally Searle is given a third batch of Chinese symbols together with another set of rules (in English) to enable him to correlate the third batch with the first two, and these rules instruct him how to return certain sets of shapes (Chinese symbols) in response to certain symbols given in the third batch.

Unknown to Searle, the people outside the room call the first batch of Chinese symbols *the script*; the second batch *the story*; the third *questions about the story* and the symbols he returns they call *answers to the questions about the story*. The set of rules he is obeying they call *the program*. To complicate the matters further, the people outside also give him stories in English and ask him questions about them in English, to which he can reply in English. After a while Searle gets so good at following the instructions and the outsiders get so good at supplying the rules which he has to follow, that the answers he gives to the questions in Chinese symbols become indistinguishable from those a true Chinese man might give.

From the external point of view the answers to the two sets of questions—one in English the other in Chinese—are equally good; Searle-in-the-Chinese-room has passed the Turing test. Yet in the Chinese case Searle behaves like a computer and does not understand either the questions he is given or the answers he returns, whereas in the English case he does. To highlight the difference consider Searle is passed a joke first in Chinese and then English. In the former case Searle-in-the-room might correctly output appropriate Chinese ideograms signifying 'ha ha' whilst remaining phenomenologically unmoved, whilst in the latter, if the joke is funny, he may laugh out loud and *feel the joke within*.

The decades since its inception have witnessed many reactions to the CRA from the computational, cognitive science, philosophical and psychological communities, with perhaps the most widely held being based on what has become known as the 'Systems Reply'. This concedes that, although the person in the room does not understand Chinese, the entire system (of the person, the room and its contents) does.

Searle finds this response entirely unsatisfactory and responds by allowing the person in the room to memorise everything (the rules, the batches of paper, etc.) so that there is nothing in the system not internalised within Searle. Now in response to the questions in Chinese and English there are two subsystems—the native English speaking Searle and the internalised Searle-in-the-Chinese-room—but all the same he [Searle] continues to understand nothing of Chinese, and *a fortiori* neither does the system, because there is nothing in the system that is not just a part of him.

But others are left equally unmoved by Searle's response; for example in [39] Haugland asks why should we unquestioningly accept Searle's conclusion that 'the internalised Chinese room system does not understand Chinese', given that Searle's responses to the questions in Chinese are all correct? Yet, despite this and other trenchant criticism, almost 30 years after its first publication there continues to be lively interest in the CRA (e.g. [40–47]). In a 2002 volume of analysis [48] comment ranged from Selmer Bringsjord who observed the CRA to be "arguably the 20th century's greatest philosophical polariser" [49], to Rey who claims that in his definition of Strong AI Searle "burdens the [Computational

Representational Theory of Thought (Strong AI)] project with extraneous claims which any serious defender of it should reject" [50]. Nevertheless, although opinion on the argument remains divided, most commentators now agree that the CRA helped shift research in artificial intelligence away from classical computationalism (which, pace Newell and Simon [51], viewed intelligence fundamentally in terms of symbol manipulation) first to a *sub-symbolic neural-connectionism* and more recently, moving even further away from symbols and representations, towards *embodied* and *enactive* approaches to cognition. Clearly, whatever the verdict on the soundness of Searle's Chinese room argument, the subsequent historical response offers eloquent testament to his conclusion that '*programs* are not minds'.

Dancing with Pixies

The core argument I wish to present in this article targeting computational accounts of cognition—the Dancing with Pixies (DwP) *reductio*—derives from ideas originally outlined by Putnam [52], Maudlin [22], Searle [53] and subsequently criticised by Chalmers [10], Klein [54] and Chrisley [55, 56] amongst others[5]. In what follows, instead of seeking to justify Putnam's claim that "every open system implements every finite state automaton" (FSA) and hence that "psychological states of the brain cannot be functional states of a computer", I will seek to establish the weaker result that, over a finite time window, every open physical system implements the trace of a FSA Q on fixed, specified input (I). That this result leads to panpsychism is clear as, equating FSA $Q(I)$ to a specific computational system that is claimed to instantiate phenomenal states as it executes, and following Putnam's procedure, identical computational (and *ex hypothesi* phenomenal) states can be found in every open physical system.

Formally DwP is a simple *reductio ad absurdum* argument that endeavours to demonstrate that:

- IF the assumed claim is true: that an appropriately programmed computer really does instantiate genuine phenomenal states
- THEN panpsychism holds
  - *However, against the backdrop of our immense scientific knowledge of the closed physical world, and the corresponding widespread desire to explain everything ultimately in physical terms, panpsychism has come to seem an implausible view…*
- HENCE we should reject the assumed claim.

[5] For early discussion of these themes see 'Minds and Machines', 4: 4, 'What is Computation?', November 1994.

The route-map for this endeavour is as follows: in the next section I introduce discrete state machines (DSMs) and FSAs and show how, with input to them defined, their behaviour can be described by a simple un-branching sequence of state transitions. I subsequently review Putnam's 1988 argument [52] that purports to show how every open physical system implements every input-less FSA. Then I apply Putnam's construction to one execution trace of any FSA with known input, such that if the FSA instantiates genuine phenomenal states as it executes, then so must any open physical system. Finally I apply the procedure to a robotic system that is claimed to instantiate machine consciousness purely in virtue of its execution of an appropriate program. The article is completed by a brief discussion of some objections to the DwP *reductio* and concludes by suggesting, at least with respect to 'hard' problems, that it may be necessary to develop an alternative metaphor for cognition to that of computation.

Discrete State Machines

In his 1950 paper 'Computing Machinery and Intelligence' [57] Turing defined DSMs as "machines that move in sudden jumps or clicks from one quite definite state to another" and explained that modern digital computers fall within the class of them. An example DSM from Turing is a simple machine that cycles through three computational states Q_1, Q_2, Q_3 at discrete clock clicks. Turing demonstrated that such a device, which continually jumps through a linear series of state transitions *like clockwork* may be implemented by a simple *discrete-position-wheel* that revolves through 120° intervals at each clock tick. Basic input can be added to such a machine by the addition of a simple brake mechanism and basic output by the addition of a light that comes on when the machine is in, say, computational state Q_3 (see Fig. 1).

An input-less FSA is specified by a set of states Q and a set of state-transitions $Q \rightarrow Q'$ for each current state Q specifying the next state Q'. Such a device is trivially implemented by Turing's discrete-position-wheel machine and a function that maps each physical wheel position W_n to a logical computational state Q_n as required. For example, considering the simple 3-state input-less FSA described in Table 1, by labelling the three discrete positions of the wheel W_1, W_2, W_3 we can map computational states of the FSA, Q_1, Q_2, Q_3, to the physical discrete positions of the wheel, W_1, W_2, W_3, such that, for example, ($W_1 \rightarrow Q_1$, $W_2 \rightarrow Q_2$, $W_3 \rightarrow Q_3$).

This mapping is observer relative; the physical position W_1 of the wheel could equally map to computational states Q_2 or Q_3 and, with other states appropriately assigned, the machine's state transition sequence (and hence its

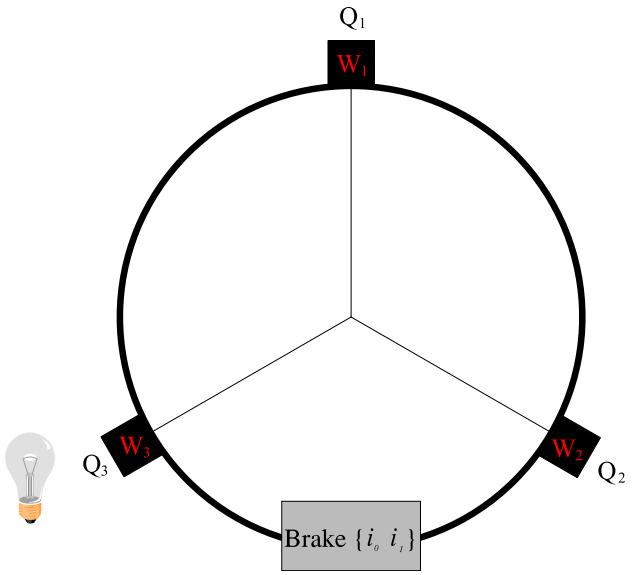

Fig. 1 Turing's discrete 3-state wheel machine

Table 1 Three-state input-less FSA

Previous state	Q_1	Q_2	Q_3
Next state	Q_2	Q_3	Q_1

computational functionality) would remain unchanged. It is central to this argument that all computational states are observer relative in this fashion (i.e. they are not, contra say mass, intrinsic to the physics of the system); computational state determination must always involve an observer-specified function that maps from the physical state of the system onto the computational state of the machine.

Note, after Chalmers, that the discrete position wheel machine described above will only implement a particular execution trace of the FSA and Chalmers remains unfazed at this result because he states that input-less machines are simply an "inappropriate formalism" for a computationalist theory of mind [32].

More generally an FSA with input and output is specified by a set of states, a set of inputs, a set of outputs, and a set of state-transitions $(Q, I) \rightarrow (Q', O)$ for each input/state pair (Q, I), specifying the next state Q' and the output O that will be produced by that state and input (see Table 2). Perhaps surprisingly, over a finite time period we can similarly implement *any* FSA with a given (i.e. specified a priori) set of input/output contingencies.

Table 2 Three-state FSA with simple input

	Previous FSA state		
Input	Q_1	Q_2	Q_3
Brake-OFF	Q_2	Q_3	Q_1
Brake-ON	Q_1	Q_2	Q_3

Over a specified time interval Turing's discrete-position-wheel machine can be made to implement an FSA with particular input and output contingencies if there is a mapping from the physical wheel positions onto formal states of the FSA, and from inputs and outputs to the system onto inputs and outputs of the FSA such that: for every formal state-transition $(Q, I) \rightarrow (Q', O)$ in the specification of the FSA, if the physical discrete-position-wheel machine is in a state W_s and receiving input i such that $f(W_s) = Q$ and $f(i) = I$, this causes it to transit into a state q' such that $f(q') = Q'$ and to produce output o such that $f(o) = O$.

Hence, although the operation of an FSA with input is in general described by a series of contingent branching state transitions which map from current state to next state, $(Q, I) \rightarrow Q'$, a priori knowledge of the input to the FSA over a finite time interval entails that such contingent behaviour can be unfolded *like clockwork* to a finite series of linear state transitions. For example, if Turing's 3-state FSA is initially in state Q_1 and the input (brake) is ON for two subsequent clock ticks and then OFF for the next three, its computational behaviour—its execution trace—is fully described by the sequence of state transitions $Q_1 \rightarrow Q_1 \rightarrow Q_1 \rightarrow Q_2 \rightarrow Q_3 \rightarrow Q_1$.

It is trivial to observe that we can fully implement this particular trace of this particular FSA with this particular input (I = BRAKE-ON$_1$, BRAKE-ON$_2$, BRAKE-OFF$_3$, BRAKE-OFF$_4$, BRAKE-OFF$_5$, BRAKE-OFF$_6$) using a six-position discrete state wheel, starting the wheel in position W_1 and using the following function to map from physical wheel positions to computational states:

$$W_1 \vee W_2 \vee W_3 \vee W_6 \rightarrow Q_1$$
$$W_4 \rightarrow Q_2$$
$$W_5 \rightarrow Q_3$$

Thus, with a priori knowledge of input to the FSA over a finite time interval we can trivially implement any FSA with particular input–output contingencies by the use of a simple mapping function to map from each physical wheel position/state W_n to each logical computational state of the FSA Q_n and output (e.g. LAMP ON/OFF) as required.

Open Physical Systems

Discussed in a short appendix to Putnam's 1988 monograph 'Representation and Reality' is a brief argument that endeavours to prove that every open physical system is a realisation of every abstract FSA and hence that functionalism fails to provide an adequate foundation for the study of the mind.

Central to Putnam's argument is the observation that every open physical system, S, is in different states at every discrete instant and hence can be characterised by a discrete series of non-cyclic natural state transitions [s_1, s_2... s_t... s_n]. Putnam argues for this on the basis that every such open system, S, is continually exposed to electromagnetic and gravitational signals from, say, a natural clock. Hence by quantising these natural states appropriately, every open physical system can by considered as a generator of discrete non-repeating state sequences, [s_1, s_2... s_∞].

Thus, reconsidering Turing's 3-state FSA machine over the fixed interval [t_1... t_6], starting in state Q_1, with input (I = BRAKE-ON$_1$, BRAKE-ON$_2$, BRAKE-OFF$_3$, BRAKE-OFF$_4$, BRAKE-OFF$_5$, BRAKE-OFF$_6$) known a priori, it is trivial to observe that, over time interval [t_1... t_6], if we map the FSA state Q_1 to the disjunction of open physical system states, [$s_1 \vee s_2 \vee s_3 \vee s_6$], FSA state Q_2 to open physical system state s_4 and FSA state Q_3 to open physical system state s_5, then the open physical system will fully implement the execution of the FSA (Q, I) as it transits open physical system states [s_1... s_6] over time interval [t_1 ... t_6].

To show that being in state Q_1 at time t_1 caused the open physical system to enter FSA state Q_1 at t_2, we observe that at t_1 the open physical system is in state s_1 (which the mapping function labels FSA state Q_1) and that being in state s_1 at t_1 causes the open physical system to enter state s_2 (which the mapping function also labels FSA state Q_1) at t_2, etc. Hence, given the current state of the open physical system at time t and the mapping function we can easily predict its future state and hence how the states of FSA evolve over the time interval under observation.[6]

Robots, Pixies and Panpsychism

At the heart of the computationalist's putative conscious robot there lies a computational system; typically a microprocessor, memory and memory-mapped peripherals. Such a system is effectively a very sophisticated DSM/FSA. Hence, if the input to the robot is fixed and specified over a finite time interval we can, pace Putnam, map the execution trace of the robot's control program onto the state evolution of any open physical system; thus, if the robot instantiates genuine phenomenal consciousness purely in virtue of its execution of its control program, so must the state evolution of any open physical system; hence the computationalist—*if* his claims are correct—leads us to embrace panpsychism, with phenomenal consciousnesses—*ethereal pixies*—dancing everywhere.

Objections

The Argument from Repeatability

The DwP *reductio* is grounded upon the notion that, for a finite period with all input known, the state transitions of a robot's control program can be mapped onto any [suitably large] discrete-position-wheel such that, if the robot instantiates phenomenal states merely in virtue of its execution of its control program, then so must any [suitably large] digital wheel or, pace Putnam, any open physical system.

Hofstadter, in a critique of the CRA [58], objects that we can only perform this type of mapping a posteriori, i.e. we can only map the robot's computational states onto the physical states of the system after the program has executed and hence know the computational states it generates. Hence Putnam's construct is not a 'real mapping' and this type of argument is simply 'not science'.

However, although Hofstadter is clearly correct to highlight that Putnam style mappings can only be applied a posteriori, this is irrelevant to the force of the DwP *reductio*. Consider a simple experiment (1) where a robot is presented with a specific stimulus—say a bright red square in the centre of its visual field—over a finite time period T_1... T_k, with the robot subsequently reporting that it perceives a vivid red square; the computationalist will assert that over this period T_1... T_k the robot instantiated the phenomenal states associated with experiencing the bright red square precisely in virtue of the execution of its control program operating on this red square input.

Now, imagine a second experiment (2) using exactly the same automaton, over exactly the same length time interval T_1... T_k, with exactly the same input, I_1... I_k. As the experimental setup is precisely the same for experiment (2) as for experiment (1) the computationalist must continue to claim that the robot continues to instantiate appropriate phenomenological states over this period and it is clear that a posteriori knowledge of the system input does not impact this claim.

But, given the vagaries inherent in any real-world experiment, how can we ensure that the input to the robot will be exactly the same in both experiments? One possible solution could be to simply ensure that the appropriate digital values are always stored on the robot's visual sensor-transducer (e.g. by disconnecting genuine optical input to the frame store and simply ensuring that vivid-red pixel values are stored appropriately in its memory map).

[6] In [32], Chalmers critiques Putnam's construction, noting that it fails to ensure that *all* states of the FSA are *reliably* transited; however, he subsequently demonstrates (ibid) that every physical system containing a 'clock' and a 'dial' will implement every input-less FSA.

Table 3 Circuit behaviour

Input-1 (V)	Input-2 (V)	Output (V)
0	0	0
0	5	0
5	0	0
5	5	5

A second possibility is to simply run the entire experiment as a virtual simulation (i.e. to use a virtual robot in a virtual reality). Clearly for the computationalist such a virtual simulation cannot impact the putative phenomenal states of the [now virtual] robot; as long as the input to the robot and its control program remain the same, appropriate contingent state transitions will occur, hence instantiating appropriate 'phenomenal states' in the robot.

Computational States Are not Observer-Relative but Are Intrinsic Properties of any Genuine Computational System

In addressing this objection I will initially consider the most primitive of computational systems—a simple two input/single output logic gate **X**, with physical behaviour fully specified by a table of voltage levels (see Table 3).

It is apparent that under mapping **A** (see Table 4), the gate **X** *computes* the logical **AND** function. Conversely, under mapping **B** (see Table 5), it is apparent that the gate **X** computes the logical **OR** function.

It follows that, at a fundamental level in the physical realisation of any logical system, such observer-relativity must hold: the computational function of the system must be contingent on the observer-determined mapping used.[7].

Furthermore it is clear that, even if the physical-to-computational mapping is known, the precise function of the system-as-a-whole remains fundamentally observer-relative: that is, "different answers grow from the concerns of different individuals."[8] Consider (a) a chess playing computational machine used to control the position of chess pieces in a game against, say, a human opponent and (b) the same program being used to control the illumination of a strip of coloured lights—the two-dimensional chess board being mapped to a one-dimensional strip of lights where the colour of each light is contingent on the value

Table 4 Mapping **A**

5 V	Computational state true
0 V	Computational state false

Table 5 Mapping **B**

0 V	Computational state true
5 V	Computational state false

(king, knight, pawn, etc.) of the piece mapped onto it—in an interactive art exhibition; clearly, to paraphrase Wittgenstein, *the meaning of a computation is contingent on its use*.

The Objection From Randomness

Superficially the DwP *reductio* only targets DSMs; it has nothing to say about the conscious state of suitably engineered Stochastic Automata [60]. In a stochastic automaton the future state of the machine is determined by a probability distribution which determines, given the current state (and any input), the probability of entering any subsequent state. Thus the tuple $\langle Q, I, O, P_{(q',o|q,i)}\rangle$ represents a stochastic automaton where Q denotes the set of states, I the set of input symbols, O the set of output symbols and $P_{(q',o|q,i)}$ the transition probabilities that the stochastic automaton transitions from state $s \in S$ to state $s' \in S$ and outputs symbol $o \in |O$ provided that $i \in I$ was the input symbol.

It is trivial to show that (a) in the theory of computation stochastic automata are no more powerful[9] than deterministic automata [61] and that (b) there exist techniques to decompose a stochastic automaton into a sequential combination of automata such that every stochastic automata can be decomposed into a controlled random source and a deterministic automaton [60]. A controlled random source is a single state stochastic automaton $\langle Q, I, O, P_{o|i}\rangle$ that returns an output symbol $o \in O$ given $i \in I$ (where the output q' becomes the input to the subsequent deterministic automaton).

Clearly pseudo-random symbols generated by computer cannot truly be random; however, what is required to implement a stochastic automaton is merely that a finite segment of the generated symbols must be statistically indistinguishable from a truly random sequence for a suitably long period of time and there exist many well-defined methods to accomplish this (for discussion see [62]). Hence over a finite period the behaviour of two machines, one whose state evolution is determined by a

[7] Although it is true that as the complexity of the logical system increases, the number of consistent computational functions that can be assigned to it diminishes, it remains the case that its computational properties will always be relative to the threshold logic value used; the $\text{physical}_{\text{state}} \rightarrow \text{computational}_{\text{state}}$ mapping will always determine the logical-function that the physical computational system instantiates.

[8] Cf. 'What is a word-processor?', in Winograd and Flores [59].

[9] The set of languages 'acceptable' by a stochastic automaton and a deterministic automaton is the same.

genuine random source (e.g. 'Shot' noise) the other by pseudo-random simulation, can be made indistinguishable. Thus, for the 'objection from randomness' to hold, any putative phenomenal states of the system must be contingent upon epiphenomenal properties of the underlying noise source and not the actual stochastic computational system.

Removing Contingencies

In the study of Bishop [63] I discuss several objections to the DwP *reductio* with perhaps the most potent coming from Chalmers who argues that "Humans do not have a single path of states along which their lives are determined. ... For any given sequence of states that a human goes through, it remains the case that if things in the world had gone slightly differently, they would have functioned in an interestingly different way. Omitting this potentiality leaves out a vital part of the description of human functioning. A wind-up toy or perhaps a videotape of my life could go through the same sequence of states, but it would not be a cognitive system. Cognition requires at least the possibility of functioning in more than one way" [32].

My initial response to this line of argument [64] is analogous to that first outlined by Maudlin [22] who, through application of the supervenience thesis, similarly questions the relevance of counterfactual-involving conditionals to conscious experience. If one blocks the connection that supports some counterfactual conditional in a currently static part of the system, is it plausible that this could change or remove the system's conscious experience? Maudlin argued that it could not.[10]

In the study of Bishop [64], I argue from a physicalist/engineering perspective that the mere deletion of state sequences—that given fixed input to the robot will never be entered—cannot affect the phenomenal states in the robot. In outline, consider what happens if a putative conscious robot R_1 with full counterfactual sensitivity is step-by-step transformed into new robot R_2, such that its resulting behaviour is determined solely by a linear un-branching series of state transitions; substituting each conditional branching state transition sequence in the evolution of R_1 with a linear state transition defined by current state and the (a priori) specified input. It is clear that, over a finite time interval and with identical input, the phenomenal experience of R_1 and R_2 must be the same. Otherwise we have a robot, R_n, ($R_1 < R_n \leq R_2$), whose phenomenal experience is somehow contingent upon the presence or absence in the automaton of a series of potential state transitions that are never entered, contravening the supervenience thesis.

Two responses to this conclusion, from Chrisley and Chalmers, tackle two distinct concerns regarding the above argument. The first, from Chalmers, is Functionalist: from this perspective the moment R_1 and R_2 cease to encapsulate the same *fine-grained functional organisation* they, through the POI, cease to have the same underlying phenomenal states. That is it is not the presence or absence of individual non-entered state transitions sequences which affects the phenomenal states of the robots, but the concomitant constriction of the supervening fine-grained functional organisation [10].

The second, from Chrisley, is Physicalist: from this perspective, as everything supervenes only on the physical, the mere removal of non-entered state transition sequences does not in-itself affect the phenomenal states of the machine. However, at the 'Toward a Science of Consciousness' conference in Tucson 2006 Ron argued that as we morph between R_1 and R_2 with the deletion of each conditional non-entered state sequence substantive physical differences between R_1 and R_2 will emerge.[11] Effectively, with each replacement of the non-entered conditional state sequences, we crucially modify or delete the concomitant [machine/object code]*conditional test and branch instructions* hence R_1 and R_2 gradually become two distinct physical systems—one will always execute each conditional state transition contingent on its (fixed) input before entering Q', the second simply entering Q' directly—and so the DwP *reductio* no longer holds.

Is Counterfactual Sensitivity Essential to a Computational Account of Cognition?

To respond to Chalmers and Chrisley's concerns I will consider a real robot R_1 operating under tightly controlled experimental conditions and a virtual robot R_2, an exact replicant of R_1, operating in a virtual reality (VR)

[10] "Suppose that a system exists whose activity through a period of time supports a mode of consciousness, e.g. a tickle or a visual sensum. The supervenience thesis tells us that, if we introduce into the vicinity of the system an entirely inert object that has absolutely no causal or physical interaction with the system, then the same activity will support the same mode of consciousness. Or again, if the activity of a system supports no consciousness, the introduction of such an inert and causally unconnected object will not bring any phenomenal state about if an active physical system supports a phenomenal state, how could the presence or absence of a causally disconnected object effect that state?" [22].

[11] In his article, 'Counterfactual computational vehicles of consciousness' [56] Chrisley states, "Bishop's main mistake: claiming that differences in counterfactual behaviour do not constitute physical differences. Presumably, it is by virtue of some physical difference between a state of $R_{1(n)}$ and the corresponding state of $R_{1(n+1)}$ that gives the former a counterfactual property the latter lacks. Note that to delete the nth transition, one would have to physically alter $R_{1(n-1)}$. So despite Bishop's claim, if R_1 and R_2 differ in their counterfactual formal properties, they must differ in their physical properties. Causal properties (even counterfactual ones) supervene on physical properties."

simulation of R_1's experimental environment. Both robots have a complex visual pathway leading from their real/virtual sensors to their computational visual cortex (e.g. a neural network) with which to perceive their environment.

In each of the following experiments the robots are instructed to report the colour of a large red square fixed in the centre of their visual field.

For the real robot operating in the real world this entails obtaining a succession of values from its optical transducer that correspond to the lighting conditions in its visual field; employing image processing algorithms to isolate the square then abstracting the array of values that define its colour. Finally these values are passed to the robot's visual cortex enabling it to report back the colour of the square.

Ex hypothesi the virtual robot, being an exact replication of the real robot in a precisely rendered virtual simulation of the real world, will perform exactly the same computational operations, extracting exactly the same array of colour values to pass to its visual sub-system, before it is also able to report the colour of the square. Hence, when we run the experiment, as both robot's control programs are exactly the same and the robots receive identical data from their environments, both robots will report seeing the vivid red square and both will experience identical phenomenal sensations (if any); if the square had been say deep purple in both the real and virtual worlds, then both robots would have reacted contingently and reported the change.

Next the virtual robot software is re-complied using two slightly different partial evaluation compilers [65] **A** and **B** to produce two new virtual robot systems R_{2a} and R_{2b}. It follows that for the two virtual robots, in virtue of the two partial evaluation compilers *pre-computing all static/known input at compile time*, knowledge of the fixed virtual input—the large red square—will enable the compilers to appropriately improve the efficiency of their object code.

The first partial evaluation compiler, compiler **A**, pre-evaluates the values of all the variables active in the robot's visual pathway and *deletes all unutilised object code including any redundant conditionals*. This is analogous to the situation described in "Removing contingencies".

In the second instance the partial evaluation compiler **B** also pre-evaluates the values of all the variables active in the robot's visual pathway, but this time merely deletes the un-utilised object code on the consequent side of each conditional that forms part of the robot's visual pathway, *leaving all the conditional statements (with their antecedents)* in situ; in this scenario although each conditional continues to be evaluated, because the input to the robot is fixed by the experimental conditions that pertain in the virtual environment—the large red square—the value of each antecedent is fixed and hence only one arm of the conditional consequent ever executed (the non-executed arm being deleted at compile time as before).

In all three cases—the original compiler and partial evaluation compilers **A** and **B**—it follows that all robots respond appropriately by indicating that they perceive the large vivid red square in the centre of their visual field. However, comparing the phenomenal states of R_1 in comparison to R_{2a}, because the compiled (machine) code executed by R_1 and R_{2a} is physically distinct, both Chrisley and Chalmers can claim the DwP *reductio* does not hold.

Conversely, considering the phenomenal states of R_1 in comparison to R_{2b}, because the compiled code that is actually executed by both robots is identical, the two physical systems are engaged in precisely the same physical activity throughout the time period and Maudlin's supervenience thesis applies. Hence, R_1 and R_2 must support precisely the same modes of consciousness (if any) through that time and the DwP *reductio* holds.[12]

Lastly, it has been suggested that the DwP *reductio* directed towards a known conscious system (e.g. human brain states sampled above their Nyquist rate [66]) continues to hold; hence we must conclude that either panpsychism is true or—even more alarmingly—that the human brain is not conscious. However the *reductio* targets computationalism—the formal abstraction and instantiation of consciousness through appropriate DSMs (and/or their stochastic variants); the DwP *reductio* does not target continuous [dynamic] systems or identity theories (where conscious properties of the system are defined to be irreducible from the underlying physical agent–environment system). For the defender of the computational metaphor to simply assume the brain is a DSM is *circulus in probando*.

Are These A Priori Critiques of the Computational Metaphor too Strong?

The three a priori arguments discussed in this article purport to show that computations are neither necessary nor sufficient for cognition; specifically that the execution of mere computations does not instantiate genuine understanding or phenomenal consciousness and hence that there

[12] It is also clear that in this case the 'system as a whole' (i.e. the environment, robot and VR and compiler) remains sensitive to counterfactuals—if we had pre-specified the experimental conditions to be a dull blue square, the partial evaluation compiler **B** would have modified object code accordingly—hence at the level of the system, Chalmers' POI continues to apply. Interestingly, as this form of compile time partial evaluation process cannot be undertaken for the real robot, the DwP *reductio* strictly no longer holds against it; however, this does not help the computationlist as any putative phenomenal states of the real robot have now become tightly bound to properties of the real-world agent/environment interactions and not the mere computations.

are limits to the use of the computational metaphor in cognitive science; but perhaps this conclusion is too strong?

How Do the A Priori Arguments Discussed Herein Accommodate the Important Results Being Obtained Through Computational Neuroscience to Cognition?

There are two responses to this question, one weak and one strong. The first—the weak response—emerges from the Chinese room and DwP *reductio*. It acknowledges the huge value that the computational metaphor plays in current psychology and neuroscience and even concedes that a future computational neuroscience may be able to simulate *any* aspect of neuronal processing and offers insights into all the workings of the brain. However, although such a computational neuroscience will result in deep understanding of cognitive processes there is a fundamental ontological divide between the *simulation of a thing* and *the thing itself*. That is we may simulate the properties of gold using a computer program but such a program does not automatically confer upon us riches (unless of course the simulation becomes duplication; an identity). Hence Searle's famous observation that "… *the idea that computer simulations could be the real thing ought to have seemed suspicious in the first place because the computer is not confined to simulating mental operations, by any means. No one supposes that computer simulations of a five-alarm fire will burn the neighbourhood down or that a computer simulation of a rainstorm will leave us all drenched. Why on earth would anyone suppose that a computer simulation of understanding actually understood anything?*" [37].

The second—the stronger response—suggests that there may be principled reasons why it may not be possible to adequately simulate all aspects of neuronal processing through a computational system; there are bounds to a computational neuroscience. Amongst others this position has been espoused by: Penrose (see section "Computations and understanding: Gödelian arguments against computationalism"); Copeland claims the belief that "the action of any continuous system can be approximated by a TM to any required degree of fineness ... is false"[13]; and Smith [68] outlines results from 'Chaos theory' which describe how 'Shadowing theorems' fundamentally limit the set of chaotic functions that a TM can model to those that are 'well-behaved'; functions that are not well-behaved cannot be computationally described.

Both of the above responses accommodate results from computational neuroscience, but clearly both also highlight fundamental limitations to the computational metaphor.

So what Is Cognition, If Not Computation?

In this article I have argued from an a priori perspective that it is time for the hegemony of the computational metaphor in Cognitive Science to be challenged; a tentative suggestion for an alternative metaphor for cognitive processes—one based on communication and interaction—has been suggested (see Nasuto et al. [11, 12]). This metaphor, based on communication and interaction, potentially offers the following advantages over the computational metaphor: firstly, from a theory-of-computation perspective, there is evidence that 'Interaction Machines' are computationally more powerful than (TM) algorithms and hence may escape Penrose style criticisms [69]; secondly, a metaphor based on communication and interaction does not explicitly perform discrete computations on discrete representations—symbol manipulations—in the classical cognitivist way and hence is much less vulnerable to CRA/DwP arguments; finally, communication—as a new biological information processing metaphor—could more efficiently and compactly describe complex neuronal operations and provide us with a better intuitive understanding of the meaning of these operations [12]. In contrast to computation, communication is not merely an observer-relative anthropomorphic projection on reality, as even simple organisms (e.g. bacteria) communicate with each other or interact with their environment. Thus the new metaphor—*cognition as communication*—is sympathetic to modern post-symbolic, anti-representationalist, embodied, enactive accounts of cognition such as those from Brooks [70], Varela [19], O'Regan [15], Thompson [71] and Bishop and Nasuto [13].

Conclusion

All matter, from the simplest particles to the most complex living organisms, undergo physical processes which in most sciences are not given any special interpretation. However, when it comes to nervous systems the situation changes abruptly. In neuroscience, and in connectionism, it is assumed that neurons and their systems possess special computational capabilities; this is equivalent to claiming that a spring, when extended by a moderate force, computes its deformation according to Hooke's law. In this

[13] Copeland's argument is detailed, but at heart he follows an extremely simple line of reasoning: consider an idealised analogue computer that can add two reals (a, b) and output one if they are the same, zero otherwise. Clearly either (a) or (b) could be non-computable numbers (in the specific formal sense of non-Turing-computable numbers). Hence clearly, there exists no TM that, for any finite precision (k), can decide the general function $F(a = b)$ (see [67] for detailed discussion of the implications of this result).

article I initially highlighted the ubiquity of this computational metaphor in the Cognitive Sciences before reviewing two well-known arguments that together purport to demonstrate that computation is neither necessary nor sufficient for cognition; I subsequently introduced a less well-known *reductio*—DwP—that endeavours to demonstrate that if computations can instantiate consciousness, then panpsychism is true and consciousness is everywhere. Taken together I conclude these arguments offer serious a priori reason to question the computational hegemony in cognitive science.

Conversely, an alternative metaphor grounded on communication has recently been suggested by Nasuto [11]. In this preliminary study we claim that treating neurons as communicating—rather than computing—with each other more accurately captures their complex, and to us fundamental, capability of modifying their behaviour depending on context. Furthermore, and in contrast to computation, we claim that communication is not merely an observer-relative anthropomorphic projection on reality, as even simple organisms (e.g. bacteria) communicate with each other or interact with their environment. Finally, although it has been robustly demonstrated that populations of simple communicating organisms can solve complex problems in optimisation and search [72, 73]; although the role of communication in human development and in social interactions cannot be overestimated [74]; although communication has advantages over computation as a metaphor for 'hard' problems of cognition—clearly much more research is required (to define, ground and develop the new metaphor) before communication (like consciousness) is 'taken seriously'.

Acknowledgments I would like to thank the reviewers for the many helpful comments I received during the preparation of this article. I would also like to thank Ron Chrisley for his many interesting criticisms regarding the DwP *reductio*. Lastly I would like to thank Slawek Nasuto and Kris de Meyer for their foundational work outlining a new 'swarm' metaphor for cognition based on communication and its subsequent analysis within the framework of stochastic diffusion search.

References

1. James W. Psychology (briefer course). New York: Holt; 1891.
2. Hubel DH, Wiesel TN. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J Physiol. 1962;160:106–54.
3. Hebb DO. Organisation of behaviour. New York: Wiley; 1949.
4. Hodgkin A, Huxley A. A quantitative description of membrane current and its application to conduction and excitation in nerve. J Physiol. 1952;117:500–44.
5. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev. 1958;65(6):386–408.
6. Werbos P. Beyond regression: new tools for prediction and analysis in the behavioural sciences. Ph.D. thesis, Cambridge, MA; 1974.
7. Harnish RM. Minds, brains, computers. Oxford: Blackwell; 2002.
8. Putnam H. Brains and behavior. In: Butler RJ, editor. Analytical philosophy (second series). New York: Barnes & Noble; 1965.
9. Putnam H. The nature of mental states. In: Putnam H, editor. Mind, language and reality: philosophical papers, vol. 2. Cambridge: Cambridge University Press; 1975.
10. Chalmers DJ. The conscious mind: in search of a fundamental theory. Oxford: Oxford University Press; 1996.
11. Nasuto SJ, Bishop JM, De Meyer K. Communicating neurons: a connectionist spiking neuron implementation of stochastic diffusion search. Neurocomputing. 2009;72:704–12.
12. Nasuto SJ, Dautenhahn K, Bishop JM. Communication as an emergent metaphor for neuronal operation. In: Nehaniv C, editor. Computation for metaphors, analogy, and agents, Aizu 1998. Lecture Notes in Artificial Intelligence, 1562. Springer: Berlin; 1999. p. 365–79.
13. Bishop JM, Nasuto SJ. Second order cybernetics and enactive perception. Kybernetes 2005;34(9–10):1309–20.
14. Noe A. Action in perception. Cambridge, MA: MIT Press; 2004.
15. O'Reagan K, Noe A. A sensorimotor account of vision and visual consciousness. Behav Brain Sci. 2001;24(5):973–1031.
16. Port RF, Van Gelder T (eds). Mind as motion: explorations in the dynamics of cognition. Cambridge, MA: MIT Press; 1998.
17. van Gelder T. What might cognition be if not computation? J Philos. 1995;92(7):345–81.
18. Von Glasersfeld E. Radical constructivism: a way of knowing and learning. Studies in mathematics education series, 6. Routledge: Falmer; 1995.
19. Varela F, Thompson E, Rosch E. The embodied mind. Cambridge MA: MIT Press; 1991.
20. Descartes R. Third meditation, II.25, AT VII.36–37; II.29, AT VII.42; 1641.
21. Block N. On a confusion about a function of consciousness. In: Block N, Flanagan O, Guzeldere G, editors. The nature of consciousness. Cambridge, MA: MIT Press; 1997.
22. Maudlin T. Computation and consciousness. J Philos. 1989; 86:407–32.
23. Torrance S. Thin phenomenality and machine consciousness. In: Chrisley R, Clowes R, Torrance S, editors. Proceedings of the 2005 symposium on next generation approaches to machine consciousness: imagination, development, intersubjectivity and embodiment, AISB'05 convention. University of Hertfordshire, Hertfordshire, 2005.
24. Searle J. The rediscovery of the mind. Cambridge, MA: MIT Press; 1992.
25. Lucas JR. Minds, machines & godel. Philosophy. 1961;36:112–27.
26. Lucas JR. Satan stultified: a rejoinder to Paul Benacerraf. Monist. 1968;52:145–58.
27. Benacerraf P. God, the devil & godel. Monist. 1967;51:9–32.
28. Penrose R. The emperor's new mind: concerning computers, minds, and the laws of physics. Oxford: Oxford University Press; 1989.
29. Penrose R. Shadows of the mind: a search for the missing science of consciousness. Oxford: Oxford University Press; 1994.
30. Penrose R. On understanding understanding. Int Stud Philos Sci. 1997;11(1):7–20.
31. Penrose R. Beyond the doubting of a shadow: a reply to commentaries on 'Shadows of the Mind'. Psyche. 1996;2(23).
32. Chalmers DJ. Does a rock implement every finite-state automaton? Synthese. 1996;108:309–33.
33. Chalmers DJ. Minds, machines and mathematics: a review of 'Shadows of the Mind' by Roger Penrose. Psyche. 1995;2(9).

34. Psyche, Symposium on Roger Penrose's shadows of the mind. Psyche. 1995. 2. http://psyche.cs.monash.edu.au/psyche-index-v2.html.
35. Bringsjord S, Xiao H. A refutation of Penrose's Gödelian case against artificial intelligence. J Exp Theor AI 2000;12:307–29.
36. Tassinari RP, D'Ottaviano IML. Cogito ergo sum non machina! About Gödel's first incompleteness theorem and turing machines, CLE e-Prints. 2007;7(3).
37. Searle J. Minds, brains and programs. Behav Brain Sci. 1980;3(3):417–57.
38. Searle J. The mystery of consciousness. London: Granta Books; 1994.
39. Haugland J. Syntax, semantics, physics. In: Preston J, Bishop JM, editors. Views into the Chinese room. Oxford University Press: Oxford; 2002; p. 360–79.
40. Freeman A. Output still not really convinced. The Times Higher April 11. London, UK; 2003.
41. Freeman A. The Chinese room comes of age: a review of Preston & Bishop. J Conscious Stud. 2004;11(5–6):156–8.
42. Garvey J. A room with a view? Philos Mag. 2003;3:61.
43. Overill J. Views into the Chinese room: new essays on Searle and artificial intelligence. J Logic Comput. 2004;14(2):325–6.
44. Rapaport WJ. Review of Preston J & Bishop M, editors. Views into the Chinese room: new essays on Searle and artificial intelligence. Aust J Philos. 2006;94(1):129–45.
45. Richeimer J. Review of Preston J and Bishop M, editors. Views into the Chinese room: new essays on Searle and artificial intelligence. Philos Books. 2004;45(2):162–7.
46. Sprevak MD. The Chinese carnival. Stud Hist Philos Sci. 2005;36:203–9.
47. Waskan JA. Review of Preston J and Bishop M, editors. Views into the Chinese room: new essays on Searle and artificial intelligence. Philos Rev. 2005;114(2):277–82.
48. Preston J, Bishop JM (eds). Views into the Chinese room. Oxford University Press; Oxford; 2002.
49. Bringsjord S, Noel R. Real robots and the missing thought-experiment. In: Preston J, Bishop JM, editors. Views into the Chinese room. Oxford: Oxford University Press; 2002. p. 360–79.
50. Rey G. Searle's misunderstanding of functionalism and strong AI. In: Preston J, Bishop JM, editors. Views into the Chinese room. Oxford: Oxford University Press, 2002. p. 360–79.
51. Newell A, Simon HA. Computer science as empirical inquiry: symbols and search. Commun ACM. 1976;19(3):113–26.
52. Putnam H. Representation and reality. Cambridge, MA: Bradford Books; 1988.
53. Searle J. Is the brain a digital computer? Proc Am Philos Assoc. 1990;64:21–37.
54. Klein C. Maudlin on computation (working paper) 2004.
55. Chrisley R. Why everything doesn't realize every computation. Minds Mach. 1995; 4:403–20.
56. Chrisley R. Counterfactual computational vehicles of consciousness. Toward a science of consciousness. April 4–8 2006. Tucson Convention Center, Tucson, AZ, USA; 2006.
57. Turing AM. Computing machinery and intelligence. Mind. 1950;49:433–60.
58. Hofstadter D. Reflections. In: Hofstadter D, Dennett DC, editors. The mind's I: fantasies and reflections on self and soul. Penguin: London; 1981.
59. Winograd T, Flores F. Understanding computers and cognition. New York: Addison-Wesley; 1986.
60. Paz A. Introduction to probabilistic automata. New York: Academic Press; 1971.
61. Sipser M. An introduction to the theory of computation. Course Technology Inc.; 1997.
62. Brent RP. Random number generation and simulation on vector and parallel computers. Lect Notes Comput Sci. 1998;1470:1–20.
63. Bishop JM. Dancing with pixies. In: Preston, J, Bishop JM, editors. Views into the Chinese room. Oxford: Oxford University Press; 2002. p. 360–79.
64. Bishop JM. Counterfactuals cannot count: a rejoinder to David Chalmers. Conscious Cogn. 2002;11(4):642–52.
65. Futamura Y. Partial evaluation of computation process—an approach to compiler-compiler. Syst Comput Controls. 1971;2:45–50.
66. Nyquist H. Certain topics in telegraph transmission theory. Trans AIEE. 1928;47:617–44.
67. Copeland BJ. The broad conception of computation. Am Behav Sci. 1997;40:690–716.
68. Smith P. Explaining chaos. Cambridge, UK: Cambridge University Press; 1998.
69. Wegner P. Why interaction is more powerful than algorithms. Commun ACM. 1997;40(5):80–91.
70. Brooks R. Intelligence without representation. Artif Intell. 1981;47:139–59.
71. Thompson E. Mind in life. Cambridge, MA: Harvard University Press; 2007.
72. Bishop JM. Stochastic searching networks. In: Proceedings of the 1st IEE International Conference on Artificial Neural Networks. London: IEE Press; 1989. p. 329–31.
73. De Meyer K, Nasuto SJ, Bishop JM. Stochastic diffusion optimisation: the application of partial function evaluation and stochastic recruitment in Swarm Intelligence optimisation. In: Abraham A, Grosam C, Ramos V, editors. Studies in computational intelligence (31): stigmergic optimization. Berlin: Springer; 2006. p. 185–207.
74. Brown R. Social psychology. New York: Free Press; 1965.