A Critique of Pure Computation: Against Strong AI and Computationalism

Colin J. Causey, *The University of Findlay*

Introduction

Can machines think? This question was posed by Alan Turing in his landmark paper "Computing Machinery and Intelligence," published in 1950. Turing had in mind a particular kind of machine, a Turing machine. Modern electronic digital computers are equivalent to Turing machines, ignoring the constraint of finite memory. For the purposes of this paper, we can define a *computer* as any machine equivalent to a Turing machine. Turing's landmark paper seeded an entire paradigm in the philosophy of mind that holds that the mind is, essentially, a computer. More precisely, the mind can be thought of as a software program running on the hardware of the brain, with mental states being identical to computational states/processes. And, if this is right, then there is in principle no barrier to creating artificial minds (1) by way of merely programming a computer in the appropriate way or (2) by merely bringing about the right sort of computational processes. At least, this is the hope and belief of many computer scientists and philosophers of mind today. Turing himself answered his own question in the affirmative and proposed a test—the Turing Test—for determining whether a computer could genuinely think and possess mentality.

Although popular views today, I want to argue that (1) and (2) are just plain wrong. More precisely, (1) can be stated as the thesis of *Strong AI*.[1] Strong AI can be defined as the following two-part thesis:

> "(*a*) an appropriately programmed computer really would *have* (or be) a mind in the same sense that you or I have,
>
> and
>
> (*b*) its following the program(s) in question would explain its ability to do the psychological things it does" (Preston & Bishop, 2002, pg. 14).

And, more precisely, (2) can be stated as the thesis of *computationalism*. Computationalism is the thesis that mental states are computational processes. As Larry Hauser puts it: "Computationalism says that computation is what thought is *essentially*: (the right) computation is *metaphysically necessary* for thought…and (right) computation

metaphysically suffices [for thought]" (Hauser, 2002, pp. 124). A *computational process* can be defined simply as a process that instantiates or carries out a computation. By *computation*, I mean a calculation carried out solely in accordance with an *effective method* [2], i.e., any calculation that can be performed by a Turing machine (Copeland, 2017). In refuting these two theses, I hope to show that pure computation cannot possibly suffice for mind.[3] The mind, whatever it is, cannot be *purely* computational in nature. Therefore, a computer *qua* computer cannot have or be a mind.

This last point brings us to the discussion of another important definition, that of mind. What *is* a mind? Well, we can consider a simple definition as follows: A *mind* is a substance or an event or process that manifests things like consciousness, qualia, thought, and intentionality. Giving a more satisfactory and rigorous definition of mind is difficult without taking a stance with regard to a particular form of dualism or materialism. Thus, for the purposes of this paper, I will simply stipulate that I take consciousness, intentionality, and qualia to be *essential* attributes of the mind. If something fails to have one or more of these attributes, then it is not a mind. I want to argue that, by way of pure computation alone, these attributes (intentionality in particular) cannot be generated.

My thesis (stated precisely) is as follows: (1) The thesis that the appropriately programmed computer, by virtue of running the right program, would literally have or be a mind in the same sense that you or I have (Strong AI) is false, and (2) The thesis that mental states are identical to computational states/processes (computationalism) is similarly false. Pure computation, therefore, can never be sufficient for mind.

Against Strong AI

In defense of part (1) of my thesis, I will provide a robust defense of the famous Chinese Room Argument. As I will argue, the Chinese Room Argument makes a powerful case against Strong AI. As part of my defense of the argument, I consider four major objections: the Systems Reply, the Robot Reply, the Brain Simulator Reply, and the reply from connectionism. As part of my consideration of the Systems Reply, I offer an original argument as a response.

The Chinese Room Argument

In 1980, in an article titled "Minds, Brains, and Programs," the philosopher John Searle first published his famous argument—dubbed the Chinese Room Argument (CRA)—against both Strong AI and the adequacy of the Turing Test. The CRA consists of a thought experiment along with a species of arguments based on the scenario. The CRA can be summarized as follows:

> Suppose we have a man named Clerk who is locked inside of a room, the Chinese Room. Clerk is a monolingual English speaker. Inside the Room is Clerk, a pencil, many sheets of paper, and a rulebook. The Room is closed off to the outside except for a small slit at the bottom of the door leading into the Room. On the outside of the door is a native Chinese speaker. The native Chinese speaker slips numerous sheets of paper with Chinese symbols on them through the slit at the bottom of the door. What's written on some of these sheets is a story written in Chinese. What's written on the remaining sheets are questions about the story (also written in Chinese). Clerk, on the inside, receives these sheets and consults his rulebook to see what to do with the information on them. The rulebook is written in English and contains instructions for how to correlate one set of Chinese symbols with another set of Chinese symbols. Clerk then follows the rules of the rulebook and (using his pencil) writes down Chinese symbols on the blank sheets of paper he has with him in the Room. Once finished, Clerk slips these sheets of paper (containing Chinese symbols) through the slit at the bottom of the door. Then, on the outside, the native Chinese speaker picks up the sheets of paper and reads them. And he is fully convinced that the man in the Room (Clerk) understands Chinese. In particular, he thinks that Clerk understood the Chinese story and provided perfectly reasonable answers to the questions posed about the story. However, Clerk in fact doesn't understand Chinese at all. To him, the Chinese characters he received as input and produced as output were just meaningless symbols (Preston & Bishop, 2002, pg. 18).

In Searle's own words (as presented by B. Jack Copeland):

> "[Clerk] do[es] not understand a word of the Chinese stories. [Clerk] ha[s] inputs and outputs that are indistinguishable from the native Chinese speaker, and [Clerk] can have any formal program you like, but [Clerk] still understand[s] nothing. [A] computer[4] for the same reasons understands nothing of any stories…[W]hatever purely formal principles you put into the computer will not be sufficient for understanding, since a human will be able to follow the formal principles without understanding…" (Copeland, 2002, pp. 110). Note that

"Clerk" has been inserted into Searle's words. Originally, Searle placed himself in the scenario.

Unfortunately, Searle does not give a precise definition of understanding. Thus, in the interest of rendering the CRA a bit more rigorous, I propose the following definition: *Understanding* is the cognitive faculty by which, or the mental state in which, one comes to know or knows the meaning or meanings of things. From this definition, we can say that x understands y just in case x knows the meaning or meanings of y. Symbolically,

$(\forall x)(\forall y)(Uxy \equiv (\forall z)(Mzy \supset Kxz))$, where U = understands, M = is a meaning of, and K = knows. Regarding knowledge, I will simply adopt the *justified true belief* definition.

To summarize, the CRA illustrates the following: A computer merely implements purely syntactical rules. And syntax, as Searle is fond of saying, is not sufficient for semantics. This is a core principle in Searle's philosophy. Given this, the computer therefore has no way of attaching meanings to the symbols it processes. Thus, as illustrated in the Chinese Room, the computer cannot attach meanings to the symbols making up the Chinese story it receives as input. Thus, under the proposed definition of understanding, the computer does not understand the Chinese story, even though it passes the Turing Test for doing so. And since understanding (and intentional states in general) is a fundamental faculty of the mind (minds like ours at least), it follows that an appropriately programmed computer, by virtue of running a program, cannot have or be a mind. But then tenet (*a*) of the Strong AI thesis is false, entailing the falsehood of Strong AI. This establishes part (1) of my thesis. Furthermore, the Turing Test, as a decisive test of genuine intelligence, is inadequate. This is the basic thrust of the CRA and its various incarnations.

A brutally simple argument following from the above reasoning and the Chinese Room scenario—oddly enough dubbed the "Brutally Simple Argument"—can be formulated as follows:

1. Programs are purely formal (syntactical).
2. Minds (human ones, at least) have semantics, mental (i.e. semantic) contents.
3. Syntax by itself is neither the same as, nor sufficient for, semantic content.
4. Therefore, programs by themselves are not constitutive of nor sufficient for minds (Preston & Bishop, 2002, pg. 28).

Searle takes the Chinese Room scenario to illustrate the truth of the third premise. What Clerk is doing in the Chinese Room is, as Searle would say, "purely syntactical." He is merely following a set of formal rules. And though Clerk passes the Turing Test for understanding Chinese, Clerk does not in fact understand Chinese, at least not by virtue of running the purely syntactical program. Therefore, the thought experiment illustrates the aforementioned core principle that syntax is not sufficient for semantics.

A quick objection should be addressed first before getting to the more interesting objections. An objector might very well say something along the following lines: "Wait a minute, Searle. The Chinese Room does not illustrate that syntax is by itself insufficient for semantics. All it shows is that the *particular* syntax of the *particular* program Clerk is running is insufficient for semantics. The reason Clerk doesn't understand Chinese is that he must just be running the wrong program."

What should we make of this objection? By my estimation, not much. Clearly, we could give any Chinese understanding program we like to Clerk (assuming it's Turing-computable) to execute and the results of the thought experiment would be the same. Recall in the above exposition of the scenario, Searle states: "[Clerk] can have *any* formal program you like…" (emphasis added). The objection that Clerk must be running the wrong program, therefore, holds no water.

Systems Reply

Perhaps the chief objection to the CRA is that all it is capable of showing is that Clerk— who is but a part of the Chinese Room—cannot understand Chinese by virtue of running the program. But maybe it is the Room as a whole that understands Chinese. Clerk is really analogous to the CPU, and the Room as a whole is analogous to the computer. And from the fact that the CPU doesn't understand Chinese, it doesn't follow that the computer as a system does not understand Chinese. Remember, the contention of Strong AI is that an appropriately programmed *computer* would have or be a mind. It does not contend that an appropriately programmed *CPU* would have or be a mind. Thus, the CRA fails to refute Strong AI. This objection is defended, for instance, by Ned Block (2002) in his article "Searle's Arguments against Cognitive Science."

Searle's response to this point is to tweak the scenario in the following way: Imagine that Clerk internalizes everything in the Room. He memorizes the rulebook and runs through the logic of the program in his head, keeping track of everything mentally without using a pencil and paper. Clerk, in this scenario, has now become the

Room. There is now nothing in the Room in the original scenario that is not inside him. The rest of the scenario then proceeds as usual. Clerk runs through the program, passes the Turing Test for understanding Chinese, and yet does not understand Chinese. And since Clerk now comprises the entire room, since *he* does not understand Chinese, the *Room* also does not understand Chinese. The Systems Reply, so says Searle, therefore fails.

With Searle's response to the Systems Reply, however, comes another objection, what might be called the Subsystems Reply. While it may now be true that the Room does not understand Chinese, it doesn't follow that therefore no *part* of the Room understands Chinese. In computer terminology, just because the computer as a whole does not understand Chinese, it doesn't follow therefore that no *subsystem* in the computer understands Chinese. The claim is that Searle moves from committing the fallacy of composition to committing the fallacy of division.

B. Jack Copeland is a proponent of this line of attack against the CRA. Regarding the standard (what Copeland calls the vanilla) CRA, Copeland argues that the argument is not logically valid. As Copeland states: "The proposition that the formal symbol manipulation carried out by Clerk does not enable Clerk to understand the Chinese story by no means entails the quite different proposition that the formal symbol manipulation carried out by Clerk does not enable the Room to understand the Chinese story" (Copeland, 2002, pp. 110).

This is a form of the Systems Reply, though Copeland wants to distance the objection from that label. Instead, Copeland calls his objection the "logical reply." Still, it can be considered to be of the same species as the Systems Reply, as the claim is still that the CRA fails to show that the Room as a whole lacks understanding and other intentional states.

Copeland formulates a revised CRA based on Searle's Systems Reply rebuttal as follows:

1. The system is part of Clerk.
2. If Clerk (in general, x) does not understand the Chinese story (in general, does not ϕ), then no part of Clerk (x) understands the Chinese story (ϕs).
3. The formal symbol manipulation carried out by Clerk does not enable Clerk to understand the Chinese story.
4. Therefore, the formal symbol manipulation carried out by Clerk does not enable the system to understand the Chinese story (Copeland, 2002, pp. 111).

In responding to this new argument, Copeland takes Searle to task by pressing his version of the Subsystems Reply, setting his sights especially on the second premise. Copeland writes: "It is all too conceivable that a homunculus or homuncular system in Clerk's head should be able to understand Chinese without Clerk being able to do so" (Copeland, 2002, pp. 112). Edward Feser adds: "But maybe…[Clerk's] conscious understanding of English might be identical to his running a certain program (the program for English competence), while at the same time, by virtue of his following the rules of the rulebook and implementing the program for Chinese understanding, there is a second stream of consciousness that *is* consciously aware of speaking and understanding Chinese, even if the English-speaking program isn't" (Feser, 2005, pg. 124). Ned Block argues similarly (Block, 2002, pp. 74). This second stream of consciousness, per Copeland, can be conceptualized as a homunculus or homuncular system in Clerk's head running a subprogram as part of the overall program Clerk is running. What Searle must do, therefore, is show that this possibility isn't actual. Otherwise, we have no grounds for thinking that premise 2 is true.

Matryoshka Homunculi Argument

In response to Copeland's Subsystems Reply, I propose the following response, what I call the Matryoshka Homunculi Argument (inspired by Daniel Dennett's idea of homuncular decomposition[5]). We can apply homuncular decomposition to Clerk as follows: Let h_i denote a homunculus. Then, there is a set $\{h_1, \ldots, h_n\}$ of homunculi inside Clerk, for some positive integer n. Then, for each homunculus in this set, there is a set of homunculi contained within it. Then, for each homunculus in this set, there is a set of homunculi contained within it, and so on until we reach a basic level of homunculi that cannot be broken down any further. **Figure 1** below provides a visual diagram of homuncular decomposition for three levels of decomposition. The strategy, of course, extends to any arbitrary number of levels.

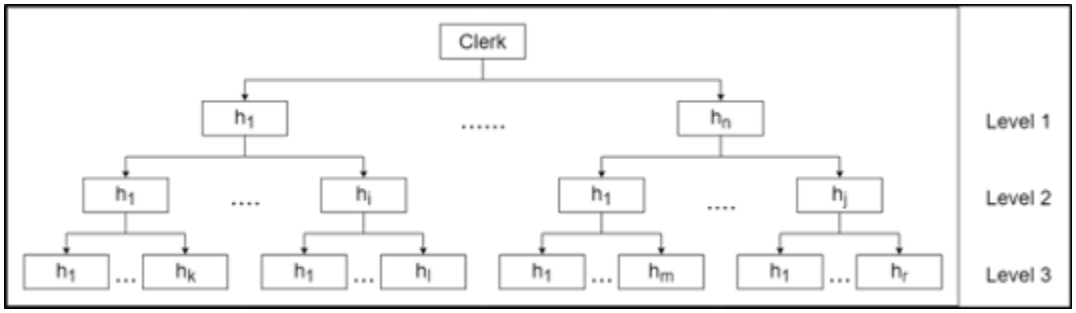

Figure 1: Tree diagram of homuncular decomposition for three levels

Now, we can apply the CRA to this scenario as follows: In response to the Systems Reply, we can, per Searle, suppose that Clerk internalizes the Room, such that he now comprises the Room. And since he does not understand Chinese by virtue of running a computer program, neither does the Room. But, per Copeland, perhaps a homunculus or homuncular system (a subsystem) inside Clerk does understand Chinese by virtue of running a computer program (a subprogram of the overall program Clerk is running). But we can now apply the CRA to this homunculus or homuncular system. Using our diagram as a visual, we can consider the first level of homunculi inside Clerk, Level 1. Level 1 contains a set of homunculi $\{h_1, \ldots, h_n\}$. Now, we know that the set as a whole does not understand Chinese because Clerk comprises this whole system, and we have shown that he does not understand Chinese. We can then show that none of the individual homunculi understand Chinese by applying the CRA to each homunculus in the set. The result is that an individual homunculus, h_i, does not understand Chinese for the same reason Clerk does not in the original Chinese Room.

But, Copeland might say, perhaps there is a subsystem, i.e., another homunculus or homuncular system in one or more of these Level 1 homunculi that understands Chinese by virtue of running a computer program. Now we are at Level 2. And for any homunculus in Level 2, we can apply the CRA to it and show that it cannot understand Chinese by virtue of running a computer program. For instance, we can take all the Level 2 homunculi contained within h_1 from Level 1 and show that none of them understands Chinese. But here one might object with the Systems Reply all over again. Perhaps the *system* of homunculi contained within, for instance, h_1 from Level 1 understands Chinese, even though none of the individual homunculi in the system do. But we have in fact already shown that this is not the case. h_1 from Level 1 comprises this whole system. And we have already shown that it does not understand Chinese. Thus, neither the system (h_1) nor the individual homunculi in the system understand Chinese by virtue of running a computer program.

This same reasoning can be carried out for the rest of the homunculi in Level 2. We can then continue on with this strategy for Level 3, and so on for however many levels Clerk happens to have. And once we reach the last, most basic level of homunculi and show (via this same strategy) that no understanding of Chinese by virtue of running a computer program can be generated in this level, we will have reached a terminus at which Copeland would be unable to appeal to some further homunculus or homuncular system that could possibly understand Chinese.

We can summarize this strategy by conceptualizing the homunculi (or "little Clerks" if we're feeling affectionate) inside Clerk as a collection of multiple sets of Russian matryoshka dolls (depicted in **Figure 2** below). The dolls in Level 1 (the largest dolls) contain the dolls in Level 2, which in turn contain the dolls in Level 3, and so on.

The dolls in the most basic level are the smallest and do not themselves contain any dolls. The CRA is then applied to each of these dolls.

Figure 2: A set of Russian matryoshka dolls

What this argument shows is that Clerk does not understand Chinese (and so, *a fortiori*, neither does the Room), and no subsystem within Clerk understands Chinese, by virtue of running a computer program. Thus, Copeland's objection fails, and the CRA stands.

One objection to this argument would be to say that maybe there is an infinite chain of homunculi. Thus, we never actually reach a terminus and Copeland can continue appealing to further homunculi *ad infinitum*. This, however, is a nonstarter. For if we suppose that Clerk's intentionality is to be explained in terms of h_1's intentionality, and h_1's intentionality is to be explained in terms of h_2's intentionality, and so on *ad infinitum*, then we never really get to any bedrock explanation, leaving Clerk's intentionality ultimately unexplained and ontologically ungrounded.

Robot Reply

Another reply to the CRA is that the reason Clerk lacks understanding is because he lacks contact with the outside world. What's needed is a way to sense things and react to stimuli, creating the right sort of causal relations between the symbols in the program and the things they refer to. The contention is, therefore, that an appropriately programmed robot (with sensors) would have a mind by virtue of running a computer program (Preston & Bishop, 2002, pg. 31). This is the Robot Reply. Searle's response to the Robot Reply is to modify the scenario by putting sensors on the outside of the room. These sensors can then scan the outside world and have causal interaction with it. But

Clerk only has access to the output of the sensors, which consists merely of more symbols for him to manipulate. Thus, Clerk still, by virtue of running the program, has no way of attaching meaning to these symbols. Therefore, Clerk still does not have understanding and other intentional states by running the program. The Robot Reply, so says Searle, thus fails.

However, the Systems Reply can then be pressed against Searle (thus combining the Systems Reply and the Robot Reply). Clerk in this revised scenario is really just a homunculus in the robot's head, implementing but a part of the robot. He isn't implementing the whole system. In particular, he isn't implementing the operation of the sensors. Perhaps the whole robot has understanding and other intentional states, despite the fact that Clerk, who is but a part of the robot, does not (Bringsjord & Noel, 2002, pp. 150-151). But Searle can respond to this objection by deploying a strategy quite similar to the one he deploys against the typical Systems Reply.[6]

As a final note, it should be said that the Robot Reply in fact tacitly concedes the point that computation alone is not enough for mental states (intentionality in particular); rather, what's needed is computation *plus* causal interaction with the environment. Thus, even if the Robot Reply should be found persuasive, it really doesn't cut against my thesis anyway.

Brain Simulator Reply

Yet another reply to the CRA (by now, I suspect you are getting the impression that this is a widely discussed argument) is the Brain Simulator Reply, which can be stated as follows:

> "Suppose…[the program] simulates the actual sequence of neuron firings at the synapses [in] the brain of a native Chinese speaker when he understands stories in Chinese and gives answers to them…[S]urely in such a case we would have to say that the machine understood the stories; and if we refuse to say that, wouldn't we also have to deny that native Chinese speakers understood the stories?" (Winograd, 2002, pp. 87).

Searle responds to this by appealing to the principle that *simulation is not duplication* (Searle, 2002, pp. 52). We can define *simulation* as follows: x simulates y just in case x replicates enough properties of y such that x can be considered to be equivalent to y in a certain context. We can define *duplication* as: x duplicates y just in case x replicates every property of y.[7] Under these definitions, simulation and duplication are not necessarily

mutually exclusive categories; nevertheless (importantly), duplication is not *entailed* by simulation.

The justification for the principle is, in my view, quite strong. In defending the principle, Searle asks us to imagine a computer simulation of digestion, pointing out that it would be absurd to suggest that such a simulation could actually digest things like beer or pizza. Searle writes, "You can simulate the cognitive processes of the human mind as you can simulate rain storms, five alarm fires, digestion, or anything else that you can describe precisely. But it is just as ridiculous to think that a system that had a simulation of consciousness and other mental processes thereby had the mental processes, as it would be to think that the simulation of digestion on a computer could thereby actually digest beer and pizza" (Searle, 2002, pp. 52). Given such powerful counterexamples to its falsehood, Searle's principle is highly plausible, providing a powerful reason to reject the Brain Simulator Reply. The fact that we can *simulate* neural processes with computation does not entail that we can *duplicate* neural processes with computation.

Connectionist Objections and the Chinese Gym

A final (and formidable) objection I will consider comes from proponents of *connectionism*, the idea that the mind is a large neural network with many nodes simultaneously interacting with each other as a parallel system. The objection is that the CRA targets a serial computer. Parallel computation performed by a collection of computers (forming computational neural networks), therefore, avoids the CRA. Searle has two responses to this. The first is to argue that parallel computations can be simulated serially. More technically, any finite collection of Turing machines can be simulated by a single universal Turing machine. And, therefore, the original CRA can be pressed against connectionism with the proviso that Clerk can simulate parallel computation by running a parallel program serially in the appropriate way.

Our friend Copeland, however, argues that this response to connectionism fails. To do so, Copeland uses Searle's core principle at work against the Brain Simulator Reply (as stated above): *simulation is not duplication*. To think that simulation is duplication would be to commit what Copeland calls the *simulation fallacy*. He states it formally as follows: "x is a simulation of y; y has property ϕ, therefore x has property ϕ" (Copeland, 2002, pp. 115). Thus, from the fact that Clerk is *simulating* parallel computations, it does not follow that he is therefore *duplicating* parallel computations. The CRA is therefore powerless against connectionism. And (I would add) if Searle, upon hearing this

objection, decides to abandon his own principle, then the Brain Simulator Reply can be pressed against him. We can formulate this reasoning in the form of a constructive dilemma as follows:

1. Either the principle that simulation is not duplication is true or it is false.
2. If the principle that simulation is not duplication is true, then the CRA fails (defeated by connectionism).
3. If the principle that simulation is not duplication is false, then the CRA fails (defeated by the Brain Simulator Reply).
4. Therefore, the CRA fails.

In my view, this objection is very plausible (under the assumption that the brain is sufficient for mind). At the very least, the objection shows that the CRA is inconclusive with regard to connectionism.

So, is connectionism victorious? Not quite. Recall that I mentioned Searle has two responses. The second response is to construct a new scenario, the Chinese Gym. Searle describes the Chinese Gym scenario as follows:

> "Imagine that instead of a Chinese room, I have a Chinese gym: a hall containing many monolingual English-speaking men. These men would carry out the same operations as the nodes and synapses in a connectionist architecture…and the outcome would be the same as having one man manipulate symbols according to a rule book. No one in the gym speaks a word of Chinese…Yet with appropriate adjustments, the system could give the correct answers to Chinese questions" (Copeland, 2002, pp. 116).

Copeland's reply to this is that another kind of Systems Reply can be mounted against the Chinese Gym. Just because none of the individuals in the Gym understand Chinese, it doesn't follow that therefore the Gym as a whole does not understand Chinese. Copeland writes, "The fallacy involved in moving from part to whole is even more glaring than in the original version of the Chinese Room Argument" (Copeland, 2002, pp. 116). In response to this objection, Searle could deploy a strategy similar to the one deployed in his response to the Systems Reply to the original Chinese Room. We can suppose that Clerk internalizes the Gym such that he now comprises the entire connectionist network. Now, questions from a Chinese speaker are posed to Clerk who then…what? If we say that Clerk simply "submits" the input to the Gym inside him and then receives the output, it is certainly true that Clerk does not understand Chinese. But to say that this shows that the Gym doesn't understand Chinese would be to beg the question. In fact, this reply is really just a facade. All we've done, essentially, is to take

the connectionist network, throw it inside a box, argue that the box doesn't understand Chinese, and then conclude that the connectionist network therefore doesn't understand Chinese.

If instead we say that Clerk is actively running through the parallel program that would have otherwise been instantiated in the connectionist network, then what Clerk is really doing (given that he is acting as a serial computer) is merely *simulating* the connectionist network, thus opening the way for our constructive dilemma. Thus, this line of argument won't work. Although I'm unwilling to say that the connectionist is ultimately victorious, I must tentatively conclude that the *CRA* is inconclusive against connectionism. However, there are other (arguably) more fundamental arguments that will be explored in the section on computationalism that strike forcefully against connectionism. And it is to that section that I shall now turn.

Against Computationalism

In defense of part (2) of my thesis, I will turn my attention to a refutation of computationalism. In many ways, the arguments presented here cut a wider swath than the CRA is capable of on its own. We've seen, for instance, that the CRA is (arguably) inconclusive against connectionism. First, it is worth pointing out how Strong AI and computationalism differ. It is, for instance, possible to adhere to one and not the other. For example, saying that an appropriately programmed computer would have or be a mind does not commit one to also saying that mental states are identical to computational processes. Strong AI does entail that computation is *sufficient* for mental states, but it is not committed to saying that mental states are *identical* to computational states. Computationalism also sets out to be a more full-bodied position than Strong AI (as defined in this paper). For instance, computationalism insists that the right causal relationships between computational states and the right (computational) processes must be generated in order to guarantee mental content. As we'll see, this is a key point made by computationalists. Programs alone are not enough. Processes are what is essential. In this way, computationalism tries to take causal powers more seriously than Strong AI, *tries* being the operative word.

Processes over Programs

In response to the Chinese Room, computationalists object and argue that the problem with the dialectic is that it conceptualizes the mind as a black box with only

inputs and outputs being relevant. But the mind, according to computationalists, is not a black box. Rather, what's essential to the mind is what goes on in between inputs and outputs—the internal processes. For instance, Georges Rey argues that the Chinese Room attacks a behavioristic strawman. In contrast to the defunct position of behaviorism, Rey argues that computationalism (he refers to this as a version of Strong AI in his article) is a species of functionalism, the dominant position in the philosophy of mind today, and that Searle misunderstands the functionalist project. Computationalism, so says Rey, takes processes and not merely behavior seriously, unlike the CRA. He further adds that the computationalist needn't be at all committed to the Turing Test (Rey, 2002, pp. 201-206). The charge is that the CRA therefore fails to refute computationalism.

A similar computationalist objection to the CRA (specifically, the Brutally Simple Argument) comes from the aforementioned Larry Hauser. Hauser challenges the first premise on the grounds that "although programs are purely syntactic, the *processes* in which they run are not" (Preston & Bishop, 2002, pg. 36). Running programs have causal and dynamic properties that "static" programs do not. Hauser concludes that the Brutally Simple Argument therefore misses the point, at least as far as computationalism is concerned. So what if *static* programs are purely syntactical? *Dynamic* programs are not (Hauser, 2002, pp. 126). The premise is, therefore, ambiguous. Does it refer to static programs or dynamic programs? If it refers to static programs, then (Hauser would concede) the premise is true, but it is irrelevant to the claims of computationalism. If instead it refers to dynamic (running) programs, then (says Hauser) the premise is false. Either way, the argument fails to refute computationalism.

I have two responses to this objection. The first response is to defend the Brutally Simple Argument against Hauser. While it may be true that the processes in which programs run are not purely syntactical, nevertheless the aspect of those processes by virtue of which they are instantiating a program (by virtue of which they are *computational* processes) is purely syntactical.[8] The processes instantiating the program do so by following the rules of the program. And those rules are purely syntactical. If Hauser wants to say that it is really the physical processes and not the formal rule-following *qua* rule-following that is constitutive of mind, then the program itself would seem to be superfluous, for it is all too conceivable that the same physical processes could be present without any program present that the processes would otherwise be instantiating.[9]

And if it is the physical processes that are supposed to be sufficient for semantics and for mind, then the program and computations themselves would seem to be

causally inefficacious, so why bother positing them as being significant to the mind in the first place? By my lights, the theory is explanatorily bankrupt and in need of a shave from Occam's razor. But then, part (*b*) of the Strong AI thesis is false, in addition to part (*a*). The appropriately programmed computer (*qua* programmed computer) cannot really have or be a mind, and its following the program in question would not explain its ability to do the psychological things it does. We thus have additional support for part (1) of my thesis. Furthermore, since the argument shows also that computation *qua* computation is not sufficient for mind, computationalism is similarly shown to be false, thus proving part (2) of my thesis.

Another response is to accept the basic thrust of Hauser's objection and abandon the Brutally Simple Argument in favor of another argument based on the Chinese Room scenario. The Chinese Room scenario is in fact a description of a *running* program, not merely a *static* program. Clerk is instantiating the program, giving it the causal and dynamic properties that Hauser has in mind in his objection. And still, Clerk does not understand Chinese by virtue of these causal and dynamic properties exhibited by the processes in which the program is executed. But then, the Chinese Room scenario shows that running programs are not sufficient for semantic content any more than static programs are. The argument can be formalized as follows:

1. Minds have understanding and other intentional states.
2. Instantiating a computer program (computational processing) is never by itself a sufficient condition for understanding and other intentional states.
3. Therefore, instantiating a computer program (computational processing) is never by itself a sufficient condition for minds.

The Chinese Room scenario, in this case, can be used to support premise 2. Hauser notes that the argument no longer makes reference to the syntax-semantics distinction, but so what? The soundness of the argument doesn't have anything to do with whether it makes reference to the syntax-semantics distinction.

Hauser, however, has another arrow in his quiver. Premise 2 of this new argument is far less obvious than premise 3 of the Brutally Simple Argument. Nevertheless, our claim is that the Chinese Room thought experiment provides support for premise 2 of this new argument just as it provides support for premise 3 of the Brutally Simple Argument. However, premise 3 of the Brutally Simple Argument (Syntax by itself is neither the same as, nor sufficient for, semantic content) is a conceptual, logical truth, the kind of truth that can be established by a thought experiment like the Chinese Room. Premise 2 of the new argument, however, is not a mere logical truth (not obviously so, anyway).

Thus, the Chinese Room experiment, says Hauser, "must bear some *empirical* weight; and here, it's *the experiment* that doesn't suffice" (Hauser, 2002, pp. 127). Hauser argues that the Chinese Room is but a single experiment and its results are contrary to those of other "real" experiments. Hauser argues that it is evidently the case, for instance, that computers are capable of doing such things as *seeking*, *comparing*, and *deciding*. And these activities are indicative of mentality and thought. Moreover, computers are evidently capable of following rules and carrying out instructions, a further indication that computers are capable of mental activities. Computational processing, Hauser concludes, therefore does seem to suffice for mind (Hauser, 2002, pp. 129, 141).

How a Computer "Computes"

In response to Hauser, I want to argue that the idea that computers literally follow rules and carry out instructions is mistaken. At the very least, the suggestion carries little force once we take into account how an electronic digital computer really works at the hardware level. The primary component of a computer, of course, is the central processing unit (CPU). Within the CPU are two major components: the control unit and the arithmetic logic unit (ALU). To simplify the discussion, I'll focus on the ALU. The ALU is responsible for performing arithmetic and logic operations such as addition, logical AND, and so forth. Let's consider a very simple 1-bit ALU as depicted below:

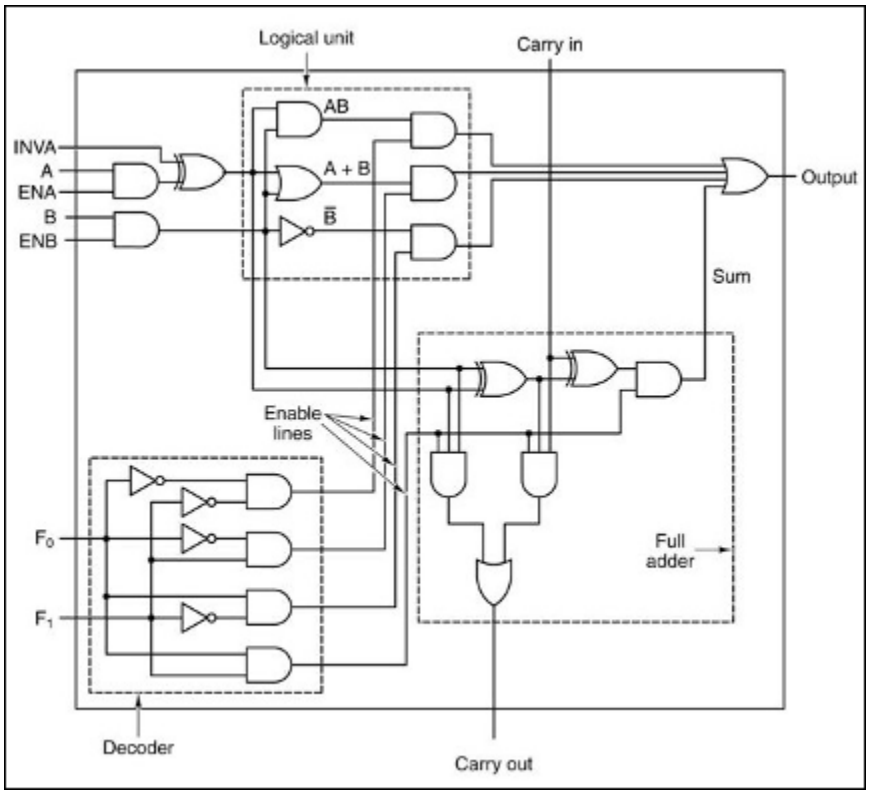

Figure 3: A 1-bit ALU, taken from (Tanenbaum & Austin, 2013, pg. 167)

This particular ALU performs four different operations: addition, logical AND, logical OR, and logical NOT. For any particular processing cycle, one operation is "selected" and "carried out" or "executed" on the "inputs" and the resulting "output" is transmitted outside of the ALU. I put these terms in scare quotes because I want to argue that the ALU does not *literally* or *intrinsically* do these things. Physically, the inputs, instructions, and outputs are nothing more than electrical signals being sent through the circuitry of the ALU. The inputs in the diagram are A and B. The instructions are encoded by F_0 and F_1. To simplify, we'll ignore INVA, ENA, and ENB, as their consideration is irrelevant to the argument. Now, the electrical signals running through the circuitry are either high voltage or low voltage (the specific numerical measure isn't important). Bits in a computer are represented by these signals and their voltages. For instance, it is common for a high voltage to correspond to a 1 and a low voltage to correspond to a 0. Thus, every bit has two possible states (hence, the name *binary* or *digital* computer).

The operation the ALU is to perform in any given cycle depends on the states of F_0 and F_1. Given that we have two variables, with each having two possible states, there

are a total of $2^2 = 4$ possible combinations of states (00, 01, 10, and 11). These four combinations correspond to the four operations the ALU can perform. For example, (F_0, F_1) = (0, 0) corresponds to logical AND, (F_0, F_1) = (0, 1) corresponds to logical OR, and so forth.

With the conceptual machinery in place, let's run through an operation performed by this ALU. Suppose we have a high voltage transmitted down the A line and a low voltage transmitted down the B line. Furthermore, let's suppose we have a high voltage running down the F_0 line and a high voltage running down the F_1 line. Then, the circuitry (whose details we have ignored for simplification purposes) causes the electrical signals being transmitted to behave in such a way that the voltage on the Output line is high. Now, recall that we've decided to treat a high voltage as being representative of a 1 and a low voltage as a 0. Our F_0 and F_1, therefore, both correspond to 1, resulting in the "selection" of the addition operation. A then corresponds to 1, and B corresponds to 0. The Output, then, being a high voltage, corresponds to 1, which corresponds to the sum of 1 + 0.

Thus, our ALU has behaved *as-if* it had performed addition. But, literally and intrinsically, all that physically happened was the transmission of electrical signals through circuitry. That these electrical processes running through silicon performed addition is true only so far as we have taken them to have done so. That is to say, it is *we* who interpret these signals as having mathematical significance. The argument, then, is that a computer has at best *as-if* intentionality (and other mental states) and not *intrinsic* intentionality (and other mental states). It does not, contrary to Hauser, literally *seek*, *compare*, and *decide* things nor does it literally follow rules and carry out instructions. Rather, it merely behaves in accordance with rules that we describe. It is rather analogous to the way in which a falling rock "follows" the law of gravity. Hauser's *empirical* weight, therefore, is as light as a feather.

An objection that might now be raised is as follows: All right, so the computer does not literally follow rules. But the processes are, as you admit, behaving in accordance with the rules of the program. And if something is behaving in accordance with the rules of a program, then it is in fact instantiating that program. That's just what it is to instantiate a program.

My reply is that this suggestion is actually very damaging to the computationalist cause. If anything that behaves in accordance with the rules of a program is instantiating that program, then the whole notion of instantiating a program becomes utterly trivial and explanatorily inept with regard to the mind. For this would

entail that nearly (and possibly) everything is instantiating an infinity of programs. Indeed, it can be argued that nearly everything instantiates every program and thus, *a fortiori*, instantiates mind programs, leaving us with an absurd panpsychism (assuming we are computationalists). This trivialization point carries with it, I think, a tremendous amount of force against both Strong AI and computationalism and goes even deeper than the Chinese Room Argument, and it is what I shall now turn to.

Trivialization Argument

In addition to his Chinese Room Argument, Searle has also given an argument that what counts as a computer is up to us, and that something only counts as a computer in the first place if we say so. Computation is, therefore, a mind-dependent, observer-relative phenomenon (Searle, 2002, pp. 67). This point is exemplified by my above argument involving the ALU. The ALU "performs" addition only relative to our saying so, relative to us interpreting its physical states as having mathematical significance. Searle then uses this idea to draw the conclusion that the concept of a computer is entirely trivial. Almost anything could count as a computer running a program. This follows because we can interpret physical states as computational states. This is in fact what we do with electronic digital computers, as explained above with the ALU. We interpret electrical voltages as having mathematical significance, as being constitutive of bits. But there isn't anything special about electricity here. Computation can be realized in other things as well. It is, as functionalists like to say, *multiply realizable*. What I want to argue is that computation is not merely multiply realizable, but *universally realizable*.

If this trivialization point is correct, then computationalism would seem to crumble. As John Preston writes, "If almost any process can count as almost any computation, then the computationalist view of cognition, instead of being the interesting (and empirical) hypothesis its advocates intend, is vacuous" (Preston & Bishop, 2002, pg. 43). Hilary Putnam and Mark Bishop provide, I think, a compelling case for this very proposition.

In the appendix of his book *Representation and Reality*, Putnam lays out an argument that every open physical system is every finite state automaton (FSA). In Putnam's own words, "Every ordinary open system is a realization of every abstract finite automaton" (Putnam, 1988, pg. 121). Bishop, in his article "Dancing with Pixies: Strong Artificial Intelligence and Panpsychism," provides a robust defense of a more modest version of this argument. Bishop argues that "over a finite time window, every open system implements the trace of a particular FSA **Q**, as it executes program (**p**) on

input (**x**)" (Bishop, 2002, pp. 361). And if the computational states **Q** goes through are sufficient for mental states and phenomenal experience, this entails, since every open physical system implements a trace of **Q**, every open physical system has mental states and phenomenal experience. Thus, computationalism, if it is true, entails panpsychism, an absurd result. We thus have an informal *reductio ad absurdum* argument against computationalism.

The argument goes like this: The operations of a Turing machine, taken over a finite time interval, can be replicated by an FSA. This is true because, over a finite time interval, a Turing machine transits a finite number of computational states. Thus, any given trace of a program executed by a Turing machine over a finite time interval can be implemented with an FSA even though, in general, Turing machines are computationally more powerful than FSAs. Thus, an FSA can be said to be equivalent to a Turing machine over a finite period of time in the sense that an FSA can replicate the same computational states as a Turing machine over the finite time interval in question. Therefore, if we have a Turing machine executing a program and instantiating computations that are hypothesized to be sufficient for mind over a finite time interval, an FSA, by instantiating those same computations over the interval in question, would also be sufficient for mind.

With all this in mind, Bishop asks us to consider an FSA **Q** with states [**A**] and [**B**] that, over a finite time interval [$t_1 \ldots t_6$], goes through the sequence of states <**A B A B A B**>. Next, we can consider any open physical system **S**. Over a finite time interval, say [$t_1 \ldots t_6$], **S** will go through physical states [$s_1 \ldots s_6$]. We can then map **Q**'s computational states onto **S**'s physical states such that [**A**] corresponds to the disjunction of [s_1 v s_3 v s_5] and [**B**] corresponds to the disjunction of [s_2 v s_4 v s_6]. With this mapping in place, as **S** transits through its physical states over the time interval in question, it will completely implement **Q** and its state transitions.[10] And since it is clear that this same procedure could be carried out for any FSA and open physical system having any (finite) number of states and going through any particular state transitions over a finite time interval, we may conclude that every open physical system implements the trace of any finite state automaton executing a program with a defined input over a finite time interval.

And since an FSA is capable of instantiating a trace of any Turing-computable program over a finite time interval, we can conclude that every open physical system implements a trace of any Turing-computable program over a finite time interval. Now, if we assume that Strong AI and computationalism are true, then, by running the right program and instantiating the right computation, a mind emerges. And since every open physical system implements a trace of any Turing-computable program over a

finite time interval, *a fortiori*, every open physical system implements a trace of any Turing-computable Strong AI program. Hence, we are left with the absurd conclusion that every open physical system is suffused with mind. Given this absurdity, it must be that our original assumption is false. That is, it must be that Strong AI and computationalism are false. Reductio argument concluded.

Perhaps the chief objection to this argument is that **S** does not properly implement **Q** because it lacks the ability to support what we might call *counterfactuals of computation*. These are counterfactuals about what computation a computer would perform were it in a particular machine state and given a particular input. Ned Block is a proponent of this objection. Although Block refers specifically to a wall, his objection can be generalized to any open physical system. It should also be noted that, in his example, Block is referring to a simple 1-bit addition program. Block writes, "In order for a wall to be [a] computer [performing addition], it isn't enough for it to have states that correspond to '*0*' and '*1*' followed by a state that corresponds to '*1*'. It must also be such that *had* the '*1*' input been replaced by a '*0*' input, the '*1*' output *would have been* replaced by the '*0*' output. In other words, it has to have symbolic states that satisfy not only the *actual* computation, but also the *possible* computations that the computer could have performed. And this is non-trivial" (Block, 2002, pp. 77).

In order to provide more clarity to both the trivialization argument and Block's objection to it, we can consider the following diagram of an FSA:

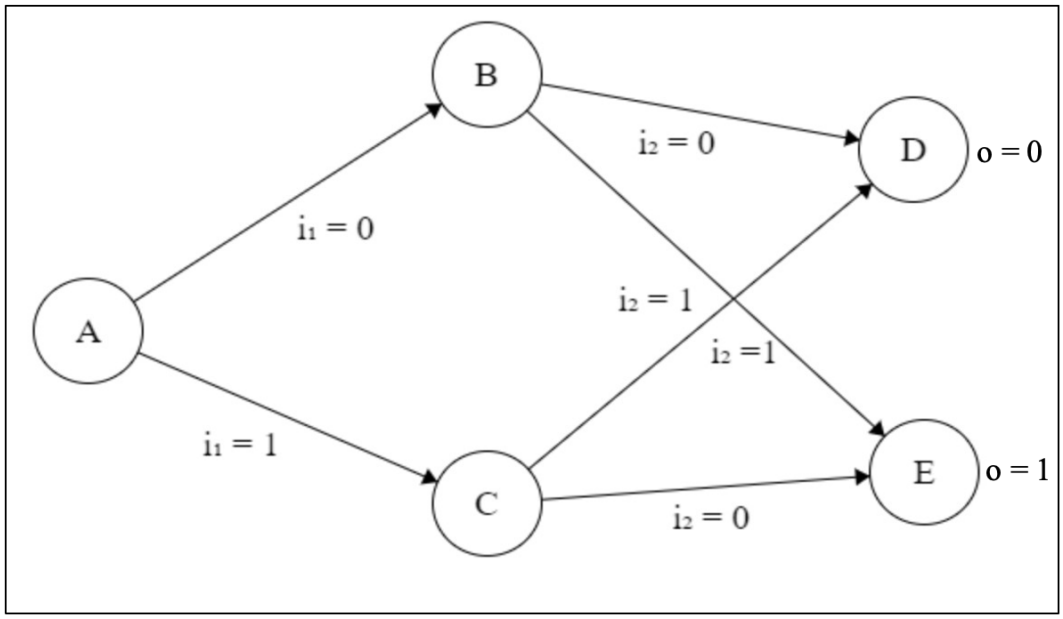

Figure 4: State transition diagram of a finite state automaton, based on (Block, 2002, pp. 77)

This is a state transition diagram of a finite state automaton that performs 1-bit addition without carry. Its behavior, like every finite state automaton, is fully determined by its current state and input. In this case, the initial state is [**A**] and the final state is either [**D**] or [**E**]. The set of inputs **i** is the set $\{i_1, i_2\}$. The set of outputs **o** is the set {o}. If, for example, we have **i** = {1, 0}, then the FSA goes through the sequence of states <**A C E**> and the output is **o** = {1}. By transiting through these states, our FSA has thereby calculated 1 + 0. Now, what has been shown in the preceding argument is that every open physical system implements a trace of every finite state automaton. For instance, the sequence of states <**A C E**> is such a trace. Thus, if we know the inputs for a given trace—in this case **i** = {1, 0}—the "combinatorial structure" of our FSA collapses and we can represent this trace of the FSA with a simple inputless FSA consisting of a simple linear path of states. Thus, if we know that **i** = {1, 0}, our FSA collapses to the following form:

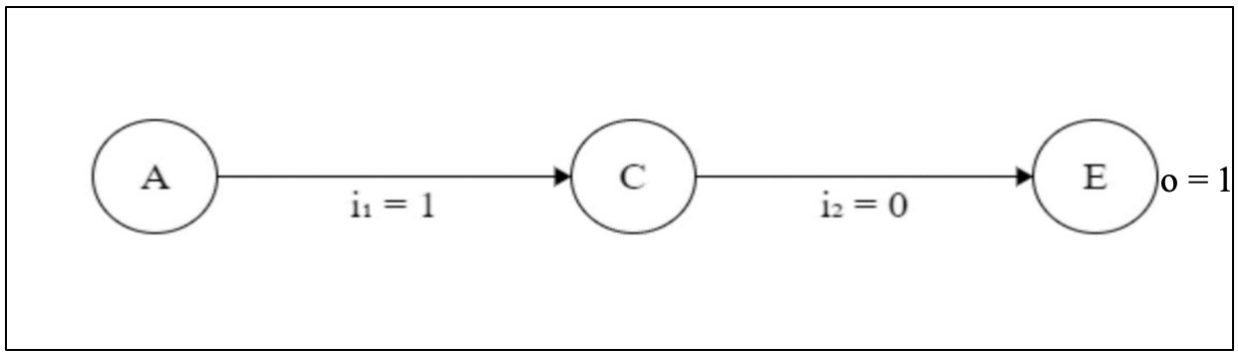

Figure 5: State transition diagram of same finite state automaton with input defined as **i =** {1, 0}

It is a particular trace such as this that is implemented in every open physical system, rather than the complete original FSA without inputs defined ahead of time. And it is this fact that Block finds objectionable. It isn't enough that every open physical system can implement something like **Figure 5**; rather, for the trivialization argument to go through, every open physical system must be able to implement something like **Figure 4**. So says Block.

But as Bishop points out, saying that counterfactuals matter commits one to saying that non-entered machine states have causal powers, a tough pill to swallow. It's difficult to see, for instance, how the mere possibility that we could have transited from state [**C**] to state [**D**] (though we did not in fact do so) could have any causal impact on anything. To drive this point home, we can imagine (as Bishop invites us to do) that we have a computing machine **Q** running a program (**p**) with known input (**x**) over some finite time interval $\{t_1 \ldots t_k\}$. Further, let's suppose that (**p**) is a Strong AI program such that running it is sufficient for mental states. Now, let's suppose we turn **Q** on and let it run over the time interval in question. Then, over this time interval, **Q**(**p**, **x**) will generate mental states. Once the time interval is up, we switch **Q** off. Now, over the time interval in question, **Q** transits a finite number of states. These state transitions can be fully replicated with an inputless finite state automaton. Thus, if **Q**(**p**, **x**) is sufficient for mental states over the finite time interval in question, then so is this inputless finite state automaton. The possibility that **Q** could have entered different states had the input (**x**) been different clearly has no effect on whether **Q** has mental states given the input (**x**) under consideration. Perhaps **Q** wouldn't be very interesting if it could only accept a single set of inputs, but that would not change the fact that **Q** would have mental states over the finite time interval in question, assuming the truth of Strong AI and computationalism. Block's objection, therefore, carries little force.

Thus, just as we can map computational states to electrical states in the case of an electronic digital computer, we can map computational states to the physical states of any open physical system. What counts as computation is, therefore, entirely trivial. Computationalism thus crumbles, establishing part (2) of my thesis.

Conclusion

What I hope to have shown is that the purely computational approach to the mind is inadequate. What this paper has established is that (1) The thesis that the appropriately programmed computer, by virtue of running the right program, would literally have or be a mind in the same sense that you or I have (Strong AI) is false, and (2) The thesis that mental states are identical to computational states/processes (computationalism) is similarly false. This is all to say that pure computation can never be sufficient for mind.

What are the implications of this result? Well, if I am right, then we are much further away from sentient machines than many enthusiasts and futurists believe. For instance, Amir Husain, the founder and CEO of SparkCognition as well as an author

and popularizer of AI research, has recently written a book, *The Sentient Machine*, in which he argues that sentient AI is seemingly right around the corner. Husain writes regarding AGI (more on this term in a moment), "Whether in twenty, seventy, or two hundred years, many in the community agree that AGI is on the horizon" (Husain, 2017, pg. 37).

Regarding the term *AGI*, Husain distinguishes between *artificial narrow intelligence* (ANI) and *artificial general intelligence* (AGI). ANI includes things like chess-playing programs and self-driving cars. ANI is capable of doing specialized tasks that would normally be done by a human. AGI, on the other hand, would be capable of a level of general intelligence and cognitive ability on par or better than that of humans. As Husain writes, "In order to be considered AGI, an AI system would…need to understand meaning and context, be able to synthesize new knowledge, have intentionality, and—in all likelihood—be self-aware, so that it could understand what it means to have agency in the world" (Husain, 2017, pg. 34). The realization of AGI, therefore, would entail the truth of Strong AI. But, given the powerful arguments against Strong AI, we have warrant for thinking that AGI cannot and will not be realized, despite the sensational prognostications of many in the AI community.

With all this being said, what I have most certainly *not* shown in this paper is that sentient machines of any kind are impossible. Perhaps it is possible to construct artificial minds. I merely maintain that we won't get there by way of simply programming a computer, by way of pure computation.

Notes

1. This is opposed to *Weak AI*. Weak AI is simply the claim that we can simulate mental processes with a computer and that computers can provide valuable insights into the mind and how it might work (pg. 14, 226).

2. The Stanford Encyclopedia of Philosophy defines an effective method as follows: "A method, or procedure, *M*, for achieving some desired result is called 'effective' (or 'systematic' or 'mechanical') just in case:

1. *M* is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
2. *M* will, if carried out without error, produce the desired result in a finite number of steps;
3. *M* can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;

4. *M* demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method" (Copeland, 2017).

3. I'll grant, however, that computation may be *necessary* for mind.

4. For clarification, it is obvious that the Chinese Room is analogous to a computer. In fact, it really is a computer (that is, a Turing machine), albeit a quirky one. To cast the elements of the Chinese Room in Turing's own computer terminology, Clerk is the executive unit and the control, and the sheets of paper and rulebook compose the store. The sheets of paper are where calculations are carried out, and the rulebook is the table of instructions or the program (Turing, 1950).

5. Edward Feser summarizes Dennett's idea of homuncular decomposition as follows: "We can usefully regard our minds as comprised of a number of subsystems that perform various mental functions: visual processing, linguistic competence, and so on. Each subsystem can itself be metaphorically understood as a 'homunculus'—a 'little man' who performs some particular task. But the functions performed by each of these homunculi can, like our own minds, be thought of as comprised of yet more basic functions performed by smaller subsystems; in other words, each of the homunculi comprising our own minds can be thought of as comprising smaller homunculi of its own" (Feser, 2005, pg. 151).

6. For instance, we can suppose that Clerk internalizes the Room (in the same way as was done in response to the standard Systems Reply). He memorizes the rulebook and runs through the program in his head. Furthermore, he implements the sensors by using his own eyes (and possibly ears as well). And still, merely by virtue of running the program, Clerk does not understand Chinese. There is certainly more that can be said for the Robot Reply (for both sides), but for considerations of length for this paper, I must direct the reader to Selmer Bringsjord and Ron Noel's article "Real Robots and the Missing Thought-Experiment in the Chinese Room Dialectic" for further discussion.

7. For example, suppose that x is a rubber tube and y is an esophagus. And suppose the rubber tube is such that pouring water down it causes it to behave similarly or identically to the esophagus with water pouring down it. In this case, the rubber tube *simulates* the esophagus in this particular context because it replicates the property of behaving in such-and-such a way when water is pouring down it. The rubber tube might not, however, *duplicate* the esophagus because it might fail to replicate other properties that the esophagus has such as having the property of making solid food inside it behave in a certain way.

8. Searle explains this more fully as follows: "Computation is defined purely formally or abstractly in terms of the implementation of a computer algorithm, and not in terms of energy transfer. Let me repeat this point: computation as standardly defined does not name a machine process… Computation is the name of an abstract mathematical process that can be implemented with machines that engage in energy transfer, but the energy transfer is not part of the definition of computation. To state the point with a little more precision: the notion 'same-implemented program' defines an equivalence class that is specified not in terms of physical or chemical processes, but in terms of abstract mathematical processes" (Searle, 2002, pp. 57).

9. Hauser could reply to this line of argument by insisting that whenever physical processes are isomorphic to computational processes, the physical processes are instantiating the computational processes. I shall have more to say about this shortly.

10. In other words, $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow s_5 \rightarrow s_6$ is isomorphic to **[A]**→**[B]**→**[A]**→**[B]**→**[A]**→**[B]**.

References

Bishop, M. (2002). Dancing with Pixies: Strong Artificial Intelligence and Panpsychism. In Preston, J., & Bishop, M., *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence* (pp. 360-378). New York, NY: Oxford University Press.

Block, N. (2002). Searle's Arguments against Cognitive Science. In Preston, J., & Bishop, M., *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence* (pp. 7079). New York, NY: Oxford University Press.

Bringsjord, S., & Noel, R. (2002). Real Robots and the Missing Thought-Experiment in the Chinese Room Dialectic. In Preston, J., & Bishop, M., *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence* (pp. 144-166). New York, NY: Oxford University Press.

Copeland, B. J. (2017, November 10). The Church-Turing Thesis. Retrieved November 15, 2018, from https://plato.stanford.edu/entries/church-turing/

Copeland, B.J. (2002). The Chinese Room from a Logical Point of View. In Preston, J., & Bishop, M., *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence* (pp. 109-122). New York, NY: Oxford University Press.

Feser, E. (2005). *Philosophy of Mind: A Short Introduction*. Oxford: Oneworld.

Hauser, L. (2002). Nixin' Goes to China. In Preston, J., & Bishop, M., *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence* (pp. 123-143). New York, NY: Oxford University Press.

Husain, A. (2018). *The Sentient Machine: The Coming Age of Artificial Intelligence*. New York, NY: Scribner.

Preston, J., & Bishop, M. (2002). *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence*. New York, US: Oxford University Press.

Putnam, H. (1988). *Representation and Reality*. Cambridge, Mass: MIT Press. Retrieved from https://login.ezproxy.findlay.edu/login?url=https://search.ebscohost.com/login.aspx?direc t=true&db=nlebk&AN=1763&site=eds-live&scope=site

Rey, G. (2002). Searle's Misunderstandings of Functionalism and Strong AI. In Preston, J., & Bishop, M., *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence* (pp. 201-225). New York, NY: Oxford University Press.

Searle, J.R. (2002). Twenty-One Years in the Chinese Room. In Preston, J., & Bishop, M., *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence* (pp. 51-69). New York, NY: Oxford University Press.

Tanenbaum, A. S., & Austin, T. (2013). *Structured Computer Organization*. US: Pearson Education.

Turing, A.M. (1950). Computing Machinery and Intelligence. *Mind*, 236: 433-60. Retrieved from https://login.ezproxy.findlay.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edsjsr&AN=edsjsr.2251299&site=eds-live&scope=site

Winograd, T. (2002). Understanding, Orientations, and Objectivity. In Preston, J., & Bishop, M., *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence* (pp. 8094). New York, NY: Oxford University Press.