Macalester Journal of Philosophy

Volume 11
Issue 1 *Spring 2002*

Article 8

3-10-2011

On Symbols in the Mind

Nate Bills
Macalester College

On Symbols in the Mind

Nate Bills

Introduction

In his 1992 book *The Rediscovery of the Mind,* John Searle set out to answer the question of whether we should consider the brain a digital computer. In the end, Searle responded with a conclusive 'No'. What was missing, he believed, was the philosophical link between the mathematics we practice, which exist in abstract terms, and the electronics we possess that are supposedly an implementation of those mathematics. "Since we have such advanced mathematics and such good electronics, we assume that somehow somebody must have done the basic philosophical work of connecting the mathematics to the electronics. But as far as I can tell, that is not the case."[12] To this day Searle wants nothing to do with what computational theorists say.

In this paper we'll examine what the computational theory of mind entails. Then we will look at the aspects of the computational theory of mind that Searle takes issue with. Finally, we will examine how Searle's critique matches up against the theory itself. Ultimately, I will argue that Searle's critique is misguided, and that the computational theory of mind can defend itself against Searle's objections.

Block's Functionalism

In 1950 Alan Turing put forth a behaviorist definition of intelligence which, over 50 years later, still remains as one of the most

[12] Searle 1, p.205

widely known theories of intelligence around. His theory was simple: place a judge in one room where s/he could communicate with a computer by teletype. If from his/her interaction with the computer the judge couldn't detect that s/he was communicating with a machine, then the computer would purportedly be "intelligent".[13]

Twenty-six years later, Ned Block served on a judicial committee whose purpose was to put Alan Turing's behavioral hypothesis to the test, in order to determine if we had built a building a machine that could deceive our fellow humans. The panel was restricted to asking questions that fell within a certain range to purposefully prevent the judges from "tricking" the computers. For instance, nobody was allowed to ask, "Is Washington DC bigger than a breadbox?"[14] As a result of these restrictions, it turned out that nearly half the panel members were duped into thinking that ELIZA, Joseph Weizenbaum's programmed machine, was human.

If anything, Turing's formulation of the behaviorist concept served to replace what was, at that time, a vague definition of mechanical computability. Today, it appears that it fails to accomplish this mission. What it fails to test is future-oriented intelligence. Suppose that a computer could be programmed with enough responses that, in theory, the judges wouldn't be able to tell the difference *even if they weren't restricted in their questioning.* A machine like this would be programmed with a certain (finite) number of strings that would represent propositions that the judge might make, in the form of

---

[13] Pinker, p.67-68
[14] Block 2, p.2

'A1...An'. So, the judge might ask a question that corresponds to statement 'A987'. The computer would then locate this proposition, and proceed to locate its corresponding response, 'B987'. Then the judge would type another message. This time the computer would search out the appropriate response to 'B987' from a long array of 'C' propositions. The process would go on from there. It is important to realize that such a machine is only possible in theory. It has been calculated that the number of sentences required of the program, even if it were to last for only one hour, would surpass the number of molecules in the known universe (and by a good deal at that!). But suppose this were possible. Doubtless the machine would pass the Turing test. But would it be *intelligent?* Block answers with a firm no. Or if it does, he says, it "has the intelligence of a juke-box."[15]

Of course the thought experiment from above is only a more extensively prepared version of Weizenbaum's ELIZA. But the moral is simple. Intelligence can't be relegated to performance or behavior. Instead it must be defined as 'idealized performance', that is, it must be thought of as revolving around the *potential* for certain types of behavior, not behavior itself. This sentiment was nicely echoed by Noam Chomsky who once wrote, "treating sciences of the mind as 'the behavioral sciences' is like calling physics the science of meter readings." Solutions to traditional mind/body problems require an understanding of what it is that actually is taking place in our minds when they're operating. Serious reflection on the subject is enough to

---

[15] Block 2, p.5

show that the behaviorist approach is inadequate for gaining this understanding.

What is the functionalist theory? To understand functionalism, it is obligatory to see how a computational system actually works. Thus the first section of this paper is committed to clarifying what computational 'function' actually is. To guide us, we will walk through a simple computation that can be performed on a machine and examine it with the aid of a methodology known as *functional analysis*. Much of today's speculation in the field of cognitive science revolves around a model of the mind that explains each level of cognitive competence (intelligence) by appealing to progressively simpler cognitive functions. But ostensibly this regression of semi-intelligent '*Homunculi' must* end somewhere. The best way to 'get to the bottom of this' is to look and see.

Consider a simple logical function that a machine might solve. Lets say $m * n = a$. The machine goes about solving the problem in the following way. First, it sets aside three separate registers, M, N, and A, each of which corresponds to its lowercase counterpart.[16] The next step is for it to place a representation of 0 in register A. Register N is then checked to see if it contains a representation of 0. If it does, the program terminates and the answer shows up as 0. If it does not, N is decreased by 1, and *m* is added to register A. So now the register N contains a representation of *n*-1, and register A contains a representation of 0. At this point the program loops back and starts over, however this time it skips the first two steps and goes straight to the third. Once again it checks if N is 0, subtracts 1 from N, adds *m* to A, etc… Eventually the program will halt when N equals 0. At that point, register A contains an exact representation of the answer, *a.* The computer doesn't simply multiply *m* * *n* in order to determine *a.* It can only go about its business by reducing the function *multiply* to more base level functions like 'add', 'subtract', 'set register to 0', and 'check register for 0'. The question now begs itself. How does the computer compute *these* basic functions? Mustn't it break these down further too?

To answer that it is necessary to pay attention to how computers work at this lower level. Understanding addition, however, requires a couple prerequisites, a basic grasp on binary notation and also on the concept of gates (We will only look at one basic function in this paper. Examining each function individually would be far too cumbersome, and would be unneccesary to the overall goals of the paper) . In the first, binary notation, it is essential to know where it splits with standard notation. Zero and one are represented alike in both binary and decimal notation. However, the number two is represented differently in the former. Here, '2' is represented by the symbol '10'. Thus, 1 + 0 = 1 in both systems. On the other hand, 1 + 1 = 2 in decimal notation, whereas 1 + 1 = 10 in binary.

The other concept deserving of an explanation is that of the gate. Remember that there are two kinds. Each accepts two inputs and emits a single output. With the AND gate, if both inputs are '1's (from here on, all numbers in quotations will signify that the number is only a representation of the number, not the number itself), the output is a '1'. Otherwise, if one or both of the inputs are 0, then the output is a '0'.

---

[16] Block 2, p.7-17

With an EXCLUSIVE-OR gate, it's a little different. This type of gate can be thought of as a 'difference detector'. It emits a '0' if both inputs are the same (i.e., '1'/'1' or '0'/'0'). It emits a '1' if the inputs are different (i.e., '1'/'0' or '0'/'1').

Now we are equipped to see how the more primitive functions are performed in binary notation. When two digits are to be added, both digits are attached to both the AND gate and the EXCLUSIVE-OR gate (these two digits might be the digits in the M and A registers described above). When adding two numbers the AND function always goes first, followed by the EXCLUSIVE-OR function. If the problem is ('1' + '0' = *a*), the AND function will go first and produce a '0'. The EXCLUSIVE-OR function will then follow. In this case it will produce a '1'. Put together, they make '01', which the machine reads as 1. The two functions will produce '00' if they are adding two '0's. The machine reads this as 0. They will produce '10' if they are adding two '1's. The machine, as you remember, understands this as 2.

What's the point of all of this? What this is all leading up to is that at some point, all computation must bottom out in what Block describes as 'primitive' processors. In this instance, these primitive processors are the AND and the EXCLUSIVE-OR gates. At this level, and no lower, the symbolic must 'touch' so to speak, the physico-chemical aspect of whatever is computing the symbols. Here the concepts of object (the gate) and function (also the gate) are indistinguishable. They are one and the same. But what *exactly* is happening here? What do statements like 'The computer reads this as' or 'understands this as' intend? A way to view this is as follows: the multiplier's and adder's states are symbols, but they are symbols that are 'about' the numbers they represent. In other words, the computer understands the symbols as we might understand them. The symbol in the N register of the multiplier function is 'about' the variable *n* that is the answer to the equation. The question, then, is what are the zeroes and ones of binary notation 'about'? In computers the zeroes and ones correspond directly to physical states existing at a particular gate. When the two input switches are both closed, an electromagnet turns on and pulls the third switch closed, completing the circuit. When either or both of the two input switches remain open, the circuit remains incomplete. These two kinds of circuits, either complete or incomplete, correspond directly to the symbols ' 1' and ' 0', in that order. But at this level, the physical states of the circuits are not *'about'* number*s* like the states existing in the adder or multiplication functions. They are simply physical states that *function* as symbols.

This will be made clearer by looking at the words of everyday language. Ordinarily it is no problem for us to distinguish between a word and its meaning. There is my name, 'Nate', which has four letters and rhymes with 'date', and then there is what my name refers to, a skinny, blonde-headed young adult in his last year of college. In other words, the letters of my name are 'about' the physical creature that now sits, jabbing away at his computer. At the same time, however, if you overheard a conversation about me at my family reunion, you might think my name were 'Nathaniel'. That wouldn't mean that I wasn't the same person, just that some members of my family employ an alternative symbol to denote my existence. Certain symbols in a computer can be thought of in the same way. In the equation $m * n = a$, any notation of symbol will suffice for the three variables. The number 2 may be used as such, or it may be exist in Roman numerals, i.e., II, or any other notation the computer is able to understand. The symbol is notation *independent.* However, a point must be reached where the symbols can no longer be treated like the example of my name. Once we examine computers at their binary level, at the primitive level of the adder's functions, we can observe a crossover taking place. The functions become notation *dependent.* The gates cannot compute any notation besides binary. It is for this reason that the functionalist stance is often summarized as being a 'Syntactic Engine' that drives a 'Semantic Engine', where 'syntax' refers to the symbols themselves, and 'semantics' refers to the meaning *implied by* those symbols.

In an age when computers are almost as advanced and sophisticated as the human mind, it is difficult to have a complete grasp on the idea of functionalism without sufficiently advanced background training. The point of the above discussion is obviously not to adequately train the reader. Rather, it mostly serves to familiarize the reader with the basic concepts involved in the theory. The idea is that the designer has found a machine whose physical aspects can be interpreted symbolically. Under this symbolic interpretation, there are regularities that co-exist with the regularities of the physical system. Block writes, "These symbolic regularities are isomorphic to rational relations among the semantic values of the symbols of a sort that are useful to us, [in this case the relation of addition]. It is the *isomorphism between these two* *functions* that explains how it is that a device that manipulates symbols manages to add numbers."[17]

The idea driving the computational theory of mind is that nature (evolution and learning) has developed a system where this same isomorphism takes place, albeit less synthetically. The hypothesis is that symbols or syntax in the brain reside within a rationally related system of symbols; that is, symbols trigger other symbols. If we imagine someone dipping his/her toe into a scalding-hot tub of bath-water, we can imagine how the symbol '*HOT!*' might trigger the symbol '*OUCH!*' which in turn might trigger the symbol that corresponds to the reflex that removes their toe from the water. The primitive processes responsible for this sequence do not, of course, 'understand' their role in the causal chain. The physical realities of the brain that act as symbols are merely being mapped onto other symbols, which then act to trigger the next symbols in the series. At a high enough level, these symbols begin to take on content as they start to compose something closer to phenomenological experience. The question is where this content derives from. There is content to our conscious experiences, so how do the symbols that compose our experience gain *their* content? The key to the answer is *function*, or use. Just as an unfamiliar symbol in our language - a new word perhaps that we've never encountered – can be difficult to understand when we first see it, the more we witness how it functions in our language, the more its meaning becomes obvious. The question of how repetitive exposure to a symbol changes the physical structure of the brain is still a mystery not

[17] Block 2, p16

well understood by philosophers or neuroscientists [18]. However, as Steven Pinker writes, this should be seen as a problem for cognitive scientists, not a mystery.

Searle's Skepticism

Searle's most famous (or infamous, depending on who you are) charge against the computational theory of mind came in the form of what is now known as the Chinese Room argument, and was intended to show that programmed syntax is incapable of generating genuine understanding. Pinker provides a clear outline:

> "A man who knows no Chinese is put in a room. Pieces of paper with squiggles on them are slipped under the door. The man has a long list of complicated instructions such as "Whenever you see [squiggle squiggle squiggle], write down [squaggle squaggle squaggle]." Some of the rules tell him to slip his scribbles back out under the door. He gets good at following the instructions. Unknown to him, the squiggles and squaggles are Chinese characters, and the instructions are an artificial intelligence program for answering the questions about stories in Chinese. As far as a person on the other side of the door knows, there is a native Chinese speaker in the room. Now, if understanding consists of running a suitable computer program, the guy must understand Chinese, because he is running such a program. But the guy doesn' t understand Chinese, not a word of it; he's just manipulating symbols. Therefore, understanding – and, by extension, any aspect of intelligence – is not the same as symbol manipulation or computation."[19]

---

[18] see Pinker's *How the Mind Works*
[19] Pinker, p.93

Much has been written in response to Searle's argument that semantics is not intrinsic to syntax. It seems the reason why Searle has found such appeal within the philosophic community is because of the argument's ability to distort our grasp on word 'understand'. It's preposterous that someone who was literally *translating* symbols into Chinese could actually have any *understanding* of it. The issue, however, is that its not so obvious that this is the case once we speed the simulation up into real-time. If the man *memorizes* the instructions (which Searle permits), and becomes intimately familiar with the instructions, to the point where he can translate the sentences almost instantaneously, then its not so clear at all that what the man retains couldn't be called 'understanding.'

Recently, Searle has rescinded his argument, claiming instead that he had absent-mindedly missed what he now sees to be the heart of the criticism. It's not so much that semantics isn't intrinsic to syntax, he says. Rather, it's that syntax is in no way intrinsic to *physics.*[20] The problem, Searle insists, lies in what is called 'multiple realizability'. I failed to mention earlier (though it may have been clear) that there is nothing about the primitive processors *in-themselves* that prevents them from being realized through varying physical mediums. Block gives a nice example of an AND gate being constructed out of 3 hunks of cheese, three mice, and a hungry cat.[21] Others have used examples involving carrier pigeons. The point is that it is the *function* of the physical structure that matters, not the physical structure itself. Why does Searle take odds with this 'multiple realizability'? The reason is because it

---

[20] Searle 1, p.210
[21] Block 2, p.11

shows that symbolic properties are *observer-dependent.* That is, the patterns that we call syntax require an outside observer to bring them to life.[22] It is not *intrinsic* to the world in the same way that, for instance, mass or gravitational attraction are. If we are looking for something that is both observer-dependent *and* intrinsic to our world, Searle insists that we're barking up the wrong tree. "Computational states are not *discovered within* the physics," he writes, "they are *assigned to* the physics." His objection is articulated as follows:

> *For any object there is some description of that object such that under that description the object is a digital computer. For any program and for any sufficiently complex object, there is some description of the object under which it is implementing the program. Thus for example the wall behind my back is right now implementing the Wordstar program, because there is some pattern of molecule movements that is isomorphic with the formal structure of Wordstar. But if the wall is implementing Wordstar, then if it is a big enough wall it is implementing any program, including any program implemented in the brain.*[23]

Is a wall really a computer? Perhaps, but if computers are only *interpreted* syntax, like Searle claims, then we won't be left with a very useful definition of computation. The problem is rooted in the fact that his argument misrepresents the isomorphism between the symbolic and the semantic. Searle wants to show us that by the definitions set up by philosophers thus far, *everything* is a digital computer. But as Block

---

[22] Another example Searle gives is that of a tree in the woods that one finds to be perfectly formulated for sitting. But, he insists it is in no way 'a chair'.
[23] Searle 1, p.208-09

writes, "...the isomorphism has to include not just a particular computation that the machine *does perform,* but all the computations that the machine *could have* performed."[24] This is not trivial. To further explicate this point, recall the simple binary equation '1' + '0' = '1'. "Now here is the point," he continues. "In order for the wall to be a computer, 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." Searle acknowledges this, but insists that there is *still* no fact of the matter that the brain is a *specific* computer. That may be true, but it in no way threatens the basic premises of the computational theory of mind. Philosophers may never show that computation as we understand it is an *intrinsic* property of the mind, but the point Searle misses is that *nobody is arguing this.* Proponents of the computational theory, Block included, aren't striving for a proof reasoned out *a priori* that there is an identity existing between minds and modern machines. Rather, the functional thesis is a *hypothesis.* It looks to open new doors for cognitive research, not to place the stamp of truth on the objective nature of the human mind. The question Searle raises about the observer-relative properties of computers, then, is largely irrelevant.

Much of Searle's critique of functionalism revolves around a similar idea. That idea is that philosophers and scientists who are devoted the computational theory of mind are, in his words, *anthropomorphizing* the non-conscious processes that go on in our brains.[25] It is his belief that we are doing this much in the same way that scientists anthropomorphized plants before the Darwinian revolution. Consider two statements about plants: 1) *Because it wants to survive,* the plant turns its leaves toward the sun. 2) Variable secretions of auxin cause plants to turn their leaves toward the sun.[26] The distinction is supposedly that the first statement contains only *'as-if'* intentionality, whereas the second contains the real thing. But what's the difference? In the first, the verb 'want', doesn' t actually correspond to a conscious desire that the plant has. Just in the same way that someone might see a brown lawn and say 'that lawn's thirsty', neither plant actually retains anything that corresponds to human desires. 'Intentionality', if the reader is unaware, is just shorthand for conscious desires wants and beliefs. But, since the plants can' t actually be the possessors of these types of attributes, Searle describes the first sentence as only *seeming* to have intentionality, not as really having it. The second sentence, however, he believes to be different. Here there only exists a description on the 'hardware' level. There is no attempt to ascribe anything like intentionality to the plant *at all.*

Searle believes we ought to think of the brain as we think of sentence number two. There are brain processes (hardware), and there is consciousness. Period. End of discussion. He writes that a 'perfect

---

[24] Block 2, p.17

[25] By 'non-conscious process' is meant any process which in theory is inaccessible to consciousness

[26] Searle 1, p.230

science of the mind' would be stated within a vocabulary of hardware.[27] Part of this scientific discourse would include talk of how the hardware functions, but the functions *in and of themselves,* cannot constitute a fact. Searle holds that the first sentence above is the route that the functionalists have taken. This is because functionalists believe that the physical medium of the brain is not nearly so important as its function. But Searle maintains that for something to 'function' it must always function *as-if* it were something else. If there is symbol processing in the brain, it is only functioning *as-if* it were the actual physical processes in the brain. If the strongest claim that we can make about the symbol or information processing in the brain is that it functions *as-if it were the actual* physical processes, then, for Searle, the question is, 'why think of it as information processing at all?'

What is to be made of Searle's charge that *as-if* intentional states should be eliminated? The best way to go about answering a question like this is to look and see. If there are physical processes and consciousness, and no more, then by inspecting neural tissue itself would provide the answer. If there is no information processing, then everything can be reduced to physical states in the brain. Is Searle right?

It turns out that the answer is no. Neuroscientists have shown that the actual material, the meat and flesh of our brains, is nothing special. In fact, one person's brain is, for all intents and purposes, materially indistinguishable from another. Pinker sums this view up in an entertaining fashion: "In the same way that all books are physically just different combinations of the same seventy-five or so characters, and all movies are physically just different patterns of charges along the tracks of a videotape, the mammoth tangle of spaghetti of the brain may all look alike when examined strand by strand."[28] This statement says a lot. The sheer variety of human thought is astonishing in itself. But it is even more astonishing to realize that this unmitigated heterogeneity cannot be attributed to anything physical at all *whatsoever*. Searle accepts as true that there is hardware and there is consciousness, and no more. But if hardware can't be granted causal role in our thought, then where does this leave us? We're left with the age-old mind-body problem once again, with the ghost in the machine!

Searle is adamant that the brain works just like any other organ. Just in the same way that the heart pumps our blood, the brain pumps out consciousness. In each instance the causal chain is a short one; the organ causes its appropriate effect. If Searle's physicalism were true, then presumably identical organs would then produce identical effects. If two plants stood in similar relations with the sun, then each plant would secrete an equivalent amount of auxin, and this would cause the plant to shift its physical structure in accordance with what would best suit the plant. Likewise, in animals, similar encounters would cause the release of similar levels of some chemical, which would then..., what? Cause the animal or person to behave similarly? If Searle's critique is right, it goes against most empirical evidence that comes from our day to day lives. Different people can behave in drastically different ways in similar

---

[27] IB, p.233

[28] Pinker, p.25

situations. This is true in spite of the fact that we all possess the same standard equipment.

At this point, perhaps we should swing our focus back around to functionalism itself. The heart of the functionalist argument is that of multiple realizability. It is not the concrete physical structure of the brain that explains consciousness. Instead, symbol manipulation existing within a structurally appropriate apparatus is what accounts for conscious phenomena. There must be no doubt that functions can be interpreted endlessly. They certainly can. We know that the heart functions to circulate blood through our bodies, but we can also say that it functions to exert gravitational force on the moon. Both are accurate statements. But are they equally meaningful? The answers to this will inevitably be tied to how we choose to define the concepts *function* and *meaning*. Before we saw impossibility of separating the two. We might look at it like this: obj%cts d%rive th%ir m%aning from th%ir function. Of course, in this instance the percentage sign functions as, or means the letter 'e'. What is key to remember about symbols is that they are not physical objects. They don't, in other words, exert a gravitational force upon the moon. So, as Searle points out, symbols will always be interpreted *as we want them to be.* We don't *need* to interpret the percentage sign as an 'e'. We might protest, "only a genuine 'e' can stand for an 'e'!" Though, with time comes convention, and with convention comes acceptance. So this isn't much of a problem. It *is* a problem, however, if the concept of function looses its meaning because its definition is too broad. For this reason, the definition of functions ought to be tightened.

Function cannot be thought of retrospectively. The purpose of functionalism is to *explain* phenomena, not attribute seemingly unrelated objects to them. While the purpose of functionalism is to explain, to explain a function is to explain its *purpose.* Nothing can have a function that is purposeless. The bulk of the argument, then, is that function is intimately wound up with the concept of design. It may be the case that in some causal sense a heart functions to exert forces on distant objects, but it would be a mistake to assert that this is one of its purposes. The purpose of the heart is to pump blood and provide the body with oxygen, and this is what accounts for the hearts function. The heart is a means to a systematic end, namely, survival.

Throughout his argument, Searle often cites knowledge of long, complicated strings of causally related chemical processes in order to display his verbal dexterity when it comes to understanding the latest in cognitive research. "Is the brain a digital computer?" is no more a philosophical question than "Is the neurotransmitter at neuromuscular junctions really acetylcholene?" (Searle, P.204). But the question I want to ask is this: what is the question that philosophers of mind are trying to answer? What is it that really matters about our brains, facts or concept? Earlier we saw that in the case of the primitive processors in computers, what is lacking is the perfect explanation as to *how* they actually work. But the 'how' isn't a question for philosophers to answer. The 'how' belongs to another discipline altogether, perhaps electronic circuitry theory or something of the like. The same can be said for the physico-chemical processes in the brain. As far as I'm concerned, the neurotransmitter at neuromuscular junctions can be Kool-Aid. It shouldn't make a difference for the philosopher. The chemistry is not important. What *matters* is what the chemical (or soft drink!) does. What matters is its function. Perhaps, over the long haul, cognitive scientists will discover how such processes are actually performed on a microscopic level. But this won't be a big day for philosophers. Even if the best way for scientists to explain to us how the brain works is through a metaphor involving three mice, some cheese, and a hungry cat, this by itself should be sufficient.

Bibliography

Block, Ned *Functionalism,*
http://www.nyu.edu/gsas/dept/philo/faculty/block/papers/functionalism.html

Block, Ned *"The Mind as the Software of the Brain",*
http://www.nyu.edu/gsas/dept/philo/faculty/block/papers/msb.html

Pinker, Steven *How the Mind Works,* Norton, New York, 1997

Searle 1, John R. *The Rediscovery of the Mind,* MIT press, Cambridge, 1994

Searle 2, John R. *Mind, Language and Society,* Basic Books, New York, 1998