Is the Brain a Digital Computer?

John R. Searle

April 25, 2002

I. Introduction. Strong AI, Weak AI and Cognitivism.

There are different ways to present a Presidential Address to the APA; the one I have chosen is simply to report on work that I am doing right now, on work in progress. I am going to present some of my further explorations into the computational model of the mind.*

The basic idea of the computer model of the mind is that the mind is the program and the brain the hardware of a computational system. A slogan one often sees is: "the mind is to the brain as the program is to the hardware." *

Let us begin our investigation of this claim by distinquishing three questions:

1. Is the brain a digital computer?

2. Is the mind a computer program?

3. Can the operations of the brain be simulated on a digital computer?

I will be addressing 1 and not 2 or 3. I think 2 can be decisively answered in the negative. Since programs are defined purely formally or syntactically and since minds have an intrinsic mental content, it follows immediately that the program by itself cannot constitute the mind. The formal syntax of the program does not by itself guarantee the presence of mental contents. I showed this a decade ago in the Chinese Room Argument (Searle,1980). A computer, me for example, could run the steps in the program for some mental capacity, such as understanding Chinese, without understanding a word of Chinese. The argument rests on the simple logical truth that syntax is not the same as, nor is it by itself sufficient for, semantics. So the answer to the second question is obviously "No".

The answer to 3. seems to me equally obviously "Yes", at least on a natural interpretation. That is, naturally interpreted, the question means: Is there some description of the brain such that under that description you could do a computational simulation of the operations of the brain. But since according to Church's thesis, anything that can be given a precise enough characterization as a set of steps can be simulated on a digital computer, it follows trivially that the question has an affirmative answer. The operations of the brain can be simulated on a digital computer in the same sense in which weather systems, the behavior of the New York stock market or the pattern of airline flights over Latin America can. So our question is not, "Is the mind a program?" The answer to that is, "No". Nor is it, "Can the brain be simulated?" The answer to that is, "Yes". The question is, "Is the brain a digital computer?" And for purposes of this discussion I am taking that question as equivalent to: "Are brain processes computational?"

One might think that this question would lose much of its interest if question 2 receives a negative answer. That is, one might suppose that unless the mind is a program, there is no interest to the question whether the brain is a computer. But that is not really the case. Even for those who agree that programs by themselves are not constitutive of mental phenomena, there is still an important question: Granted

that there is more to the mind than the syntactical operations of the digital computer; nonetheless, it might be the case that mental states are *at least* computational states and mental processes are computational processes operating over the formal structure of these mental states. This, in fact, seems to me the position taken by a fairly large number of people.

I am not saying that the view is fully clear, but the idea is something like this: At some level of description brain processes are syntactical; there are so to speak, "sentences in the head". These need not be sentences in English or Chinese, but perhaps in the "Language of Thought" (Fodor, 1975). Now, like any sentences, they have a syntactical structure and a semantics or meaning, and the problem of syntax can be separated from the problem of semantics. The problem of semantics is: How do these sentences in the head get their meanings? But that question can be discussed independently of the question: How does the brain work in processing these sentences? A typical answer to that latter question is: The brain works as a digital computer performing computational operations over the syntactical structure of sentences in the head.

Just to keep the terminology straight, I call the view that all there is to having a mind is having a program, Strong AI, the view that brain processes (and mental processes) can be simulated computationally, Weak AI. and the view that the brain is a digital computer, Cognitivism.

This paper is about Cognitivism, and I had better say at the beginning what motivates it. If you read books about the brain (say Shepherd (1983) or Kuffler and Nicholls (1976)) you get a certain picture of what is going on in the brain. If you then turn to books about computation (say Boolos and Jeffrey, 1989) you get a picture of the logical structure of the theory of computation. If you then turn to books about cognitive science, (say Pylyshyn, 1985) they tell you that what the brain books describe is really the same as what the computability books were describing. Philosophically speaking, this does not smell right to me and I have learned, at least at the beginning of an investigation, to follow my sense of smell.

II. The Primal Story

I want to begin the discussion by trying to state as strongly as I can why cognitivism has seemed intuitively appealing. There is a story about the relation of human intelligence to computation that goes back at least to Turing's classic paper (1950), and I believe it is the foundation of the Cognitivist view. I will call it the Primal Story:

> We begin with two results in mathematical logic, the Church–Turing thesis (or equivalently, the Churchs's thesis) and Turing's theorem. For our purposes, the Church–Turing thesis states that for any algorithm there is some Turing machine that can implement that algorithm. Turing's thesis says that there is a Universal Turing Machine which can simulate any Turing Machine. Now if we put these two together we have the result that a Universal Turing Machine can implement any algorithm whatever.

But now, what made this result so exciting? What made it send shivers up and down the spines of a whole generation of young workers in artificial intelligence is the following thought: Suppose the brain is a Universal Turing Machine.

Well, are there any good reasons for supposing the brain might be a Universal Turing Machine? Let us continue with the Primal Story

> It is clear that at least some human mental abilities are algorithmic. For example, I can consciously do long division by going through the steps of an algorithm for solving long division problems. It is furthermore a consequence of the Church–Turing thesis and Turing's theorem that anything a human can do algorithmically can be done on a Universal

> Turing Machine. I can implement, for example, the very same algorithm that I use for long division on a digital computer. In such a case, as described by Turing (l950), both I, the human computer, and the mechanical computer are implementing the same algorithm, I am doing it consciously, the mechanical computer nonconsciously. Now it seems reasonable to suppose there might also be a whole lot of mental processes going on in my brain nonconsciously which are also computational. And if so, we could find out how the brain works by simulating these very processes on a digital computer. Just as we got a computer simulation of the processes for doing long division, so we could get a computer simulation of the processs for understanding language, visual perception, categorization, etc.

"But what about the semantics? After all, programs are purely syntactical." Here another set of logico-mathematical results comes into play in the Primal Story.

> The development of proof theory showed that within certain well known limits the semantic relations between propositions can be entirely mirrored by the syntactic relations between the sentences that express those propositions. Now suppose that mental contents in the head are expressed syntactically in the head, then all we would need to account for mental processes would be computational processes between the syntactical elements in the head. If we get the proof theory right the semantics will take care of itself; and that is what computers do: they implement the proof theory.

We thus have a well defined research program. We try to discover the programs being implemented in the brain by programming computers to implement the same programs. We do this in turn by getting the mechanical computer to match the performance of the human computer (i.e. to pass the Turing Test) and then getting the psychologists to look for evidence that the internal processes are the same in the two types of computer.

Now in what follows I would like the reader to keep this Primal Story in mind—notice especially Turing's contrast between the conscious implementation of the program by the human computer and the nonconscious implementation of programs, whether by the brain or by the mechanical computer; notice furthermore the idea that we might just *discover* programs running in nature, the very same programs that we put into our mechanical computers.

If one looks at the books and articles supporting Cognitivism one finds certain common assumptions, often unstated, but nonetheless pervasive. **First**, it is often assumed that the only alternative to the view that the brain is a digital computer is some form of dualism. The idea is that unless you believe in the existence of immortal Cartesian souls, you must believe that the brain is a computer. Indeed, it often seems to be assumed that the question whether the brain is a physical mechanism determining our mental states and whether the brain is a digital computer are the same question. Rhetorically speaking, the idea is to bully the reader into thinking that unless he accepts the idea that the brain is some kind of computer, he is committed to some weird antiscientific views. Recently the field has opened up a bit to allow that the brain might not be an old fashioned von Neumann style digital computer, but rather a more sophisticated kind of parallel processing computational equipment. Still, to deny that the brain is computational is to risk losing your membership in the scientific community. **Second**, it is also assumed that the question whether brain processes are computational is just a plain empirical question. It is to be settled by factual investigation in the same way that such questions as whether the heart is a pump or whether green leaves do photosynthesis were settled as matters of fact. There is no room for logic chopping or conceptual analysis, since we are talking about matters of hard scientific fact. Indeed I think many people who work in this field would doubt that the title of this paper poses an appropriate philosophic question at all. "Is the brain really a digital computer?" is no more a philosophical question than "Is the neurotransmitter at neuro-muscular junctions really acetylcholene?"

Even people who are unsympathetic to Cognitivism, such as Penrose and Dreyfus, seem to treat it as a straightforward factual issue. They do not seem to be worried about the question what sort of claim it might be that they are doubting. But I am puzzled by the question: What sort of fact about the brain could constitute its being a computer? **Third**, another stylistic feature of this literature is the haste and sometimes even carelessness with which the foundational questions are glossed over. What exactly are the anatomical and physiological features of brains that are being discussed? What exactly is a digital computer? And how are the answers to these two questions supposed to connect? The usual procedure in these books and articles is to make a few remarks about 0's and 1's, give a popular summary of the Church–Turing thesis, and then get on with the more exciting things such as computer achievements and failures. To my surprise in reading this literature I have found that there seems to be a peculiar philosophical hiatus. On the one hand, we have a very elegant set of mathematical results ranging from Turing's theorem to Church's thesis to recursive function theory. On the other hand, we have an impressive set of electronic devices which we use every day. 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. On the contrary, we are in a peculiar situation where there is little theoretical agreement among the practitioners on such absolutely fundamental questions as, What exactly is a digital computer? What exactly is a symbol? What exactly is a computational process? Under what physical conditions exactly are two systems implementing the same program?

III. The Definition of Computation

Since there is no universal agreement on the fundamental questions, I believe it is best to go back to the sources, back to the original definitions given by Alan Turing.

According to Turing, a Turing machine can carry out certain elementary operations: It can rewrite a 0 on its tape as a 1, it can rewrite a 1 on its tape as a 0, it can shift the tape 1 square to the left, or it can shift the tape 1 square to the right. It is controlled by a program of instruction and each instruction specifies a condition and an action to be carried out if the condition is satisfied.

That is the standard definition of computation, but, taken literally, it is at least a bit misleading. If you open up your home computer you are most unlikely to find any 0's and 1's or even a tape. But this does not really matter for the definition. To find out if an object is really a digital computer, it turns out that we do not actually have to look for 0's and 1's, etc.; rather we just have to look for something that we could *treat as* or *count as* or *could be used to* function as a 0's and 1's. Furthermore, to make the matter more puzzling, it turns out that this machine could be made out of just about anything. As Johnson-Laird says, "It could be made out of cogs and levers like an old fashioned mechanical calculator; it could be made out of a hydraulic system through which water flows; it could be made out of transistors etched into a silicon chip through which electric current flows; it could even be carried out by the brain. Each of these machines uses a different medium to represent binary symbols. The positions of cogs, the presence or absence of water, the level of the voltage and perhaps nerve impulses" (Johnson Laird, 1988, p. 39).

Similar remarks are made by most of the people who write on this topic. For example, Ned Block (Block, 1990), shows how we can have electrical gates where the 1's and 0's are assigned to voltage levels of 4 volts and 7 volts respectively. So we might think that we should go and look for voltage levels. But Block tells us that 1 is only "conventionally" assigned to a certain voltage level. The situation grows more puzzling when he informs us further that we did not need to use electricity at all but we could have used an elaborate system of cats and mice and cheese and make our gates in such as way that the cat will strain at the leash and pull open a gate which we can also treat as if it were a 0 or

1. The point, as Block is anxious to insist, is "the irrelevance of hardware realization to computational description. These gates work in different ways but they are nonetheless computationally equivalent" (p. 260). In the same vein, Pylyshyn says that a computational sequence could be realized by "a group of pigeons trained to peck as a Turing machine!" (Pylyshn, 1985, p. 57)

But now if we are trying to take seriously the idea that the brain is a digital computer, we get the uncomfortable result that we could make a system that does just what the brain does out of pretty much anything. Computationally speaking, on this view, you can make a "brain" that functions just like yours and mine out of cats and mice and cheese or levers or water pipes or pigeons or anything else provided the two systems are, in Block's sense, "computationally equivalent" . You would just need an awful lot of cats, or pigeons or waterpipes, or whatever it might be. The proponents of Cognitivism report this result with sheer and unconcealed delight. But I think they ought to be worried about it, and I am going to try to show that it is just the tip of a whole iceberg of problems.

IV. First Difficulty: Syntax is not Intrinsic to Physics.

Why are the defenders of computationalism not worried by the implications of multiple realizability? The answer is that they think it is typical of functional accounts that the same function admits of multiple realizations. In this respect, computers are just like carburettors and thermostats. Just as carburettors can be made of brass or steel, so computers can be made of an indefinite range of hardware materials.

But there is a difference: The classes of carburettors and thermostats are defined in terms of the production of certain *physical* effects. That is why, for example, nobody says you can make carburettors out of pigeons. But the class of computers is defined syntactically in terms of the *assignment* of 0's and 1's. The multiple realizability is a consequence not of the fact that the same physical effect can be achieved in different physical substances, but that the relevant properties are purely syntactical. The physics is irrelevant except in so far as it admits of the assignments of 0's and 1's and of state transitions between them.

But this has two consequences which might be disastrous:

1. The same principle that implies multiple realizability would seem to imply universal realizability. If computation is defined in terms of the assignment of syntax then everything would be a digital computer, because any object whatever could have syntactical ascriptions made to it. You could describe anything in terms of 0's and 1's.

2. Worse yet, syntax is not intrinsic to physics. The ascription of syntactical properties is always relative to an agent or observer who treats certain physical phenomena as syntactical.

Now why exactly would these consequences be disastrous?

Well, we wanted to know how the brain works, specifically how it produces mental phenomena. And it would not answer that question to be told that the brain is a digital computer in the sense in which stomach, liver, heart, solar system, and the state of Kansas are all digital computers. The model we had was that we might discover some fact about the operation of the brain which would show that it is a computer. We wanted to know if there was not some sense in which brains were *intrinsically* digital computers in a way that green leaves intrinsically perform photosynthesis or hearts intrinsically pump blood. It is not a matter of us arbitrarily or "conventionally" assigning the word "pump" to hearts or "photosynthesis" to leaves. There is an actual fact of the matter. And what we were asking is, "Is there in that way a fact of the matter about brains that would make them digital computers?" It does not answer that question to be told, yes, brains are digital computers because everything is a digital computer.

On the standard textbook definition of computation,

1. For any object there is some description of that object such that under that description the object is a digital computer.

2. For any program there is some sufficiently complex object such that 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 which 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.

I think the main reason that the proponents do not see that multiple or universal realizability is a problem is that they do not see it as a consequence of a much deeper point, namely that the "syntax" is not the name of a physical feature, like mass or gravity. On the contrary they talk of "syntactical engines" and even "semantic engines" as if such talk were like that of gasoline engines or diesel engines, as if it could be just a plain matter of fact that the brain or anything else is a syntactical engine.

I think it is probably possible to block the result of universal realizability by tightening up our definition of computation. Certainly we ought to respect the fact that programmers and engineers regard it as a quirk of Turing's original definitions and not as a real feature of computation. Unpublished works by Brian Smith, Vinod Goel, and John Batali all suggest that a more realistic definition of computation will emphasize such features as the causal relations among program states, programmability and controllability of the mechanism, and situatedness in the real world. But these further restrictions on the definition of computation are no help in the present discussion because the really deep problem is that syntax is essentially an observer relative notion. The multiple realizability of computationally equivalent processes in different physical media was not just a sign that the processes were abstract, but that they were not intrinsic to the system at all. They depended on an interpretation from outside. We were looking for some facts of the matter which would make brain processes computational; but given the way we have defined computation, there never could be any such facts of the matter. We can't, on the one hand, say that anything is a digital computer if we can assign a syntax to it and then suppose there is a factual question intrinsic to its physical operation whether or not a natural system such as the brain is a digital computer.

And if the word "syntax" seems puzzling, the same point can be stated without it. That is, someone might claim that the notion of "syntax" and "symbols" are just a manner of speaking and that what we are really interested in is the existence of systems with discrete physical phenomena and state transitions between them. On this view we don't really need 0's and 1's; they are just a convenient shorthand. But, I believe, this move is no help. A physical state of a system is a computational state only relative to the assignment to that state of some computational role, function, or interpretation. The same problem arises without 0's and 1's because notions such as computation, algorithm and program do not name intrinsic physical features of systems. Computational states are not *discovered within* the physics, they are *assigned* to the physics.

This is a different argument from the Chinese Room Argument and I should have seen it ten years ago but I did not. The Chinese Room Argument showed that semantics is not intrinsic to syntax. I am now making the separate and different point that syntax is not intrinsic to physics. For the purposes of the original argument I was simply assuming that the syntactical characterization of the computer was unproblematic. But that is a mistake. There is no way you could discover that something is intrinsically a digital computer because the characterization of it as a digital computer is always relative to an observer who assigns a syntactical interpretation to the purely physical features of the system. As

applied to the Language of Thought hypothesis, this has the consequence that the thesis is incoherent. There is no way you could discover that there are, intrinsically, unknown sentences in your head because something is a sentence only relative to some agent or user who uses it as a sentence. As applied to the computational model generally, the characterization of a process as computational is a characterization of a physical system from outside; and the identification of the process as computational does not identify an intrinsic feature of the physics, it is essentially an observer relative characterization.

This point has to be understood precisely. I am not saying there are *a priori* limits on the patterns we could discover in nature. We could no doubt discover a pattern of events in my brain that was isomorphic to the implementation of the vi program on this computer. But to say that something is *functioning as* a computational process is to say something more than that a pattern of physical events is occuring. It requires the assignment of a computational interpretation by some agent. Analogously, we might discover in nature objects which had the same sort of shape as chairs and which could therefore be used as chairs; but we could not discover objects in nature which were functioning as chairs, except relative to some agents who regarded them or used them as chairs.

V. Second Difficulty: The Homunculus Fallacy is Endemic to Cognitivism.

So far, we seem to have arrived at a problem. Syntax is not part of physics. This has the consequence that if computation is defined syntactically then nothing is intrinsically a digital computer solely in virtue of its physical properties. Is there any way out of this problem? Yes, there is, and it is a way standardly taken in cognitive science, but it is out of the frying pan and into the fire. Most of the works I have seen in the computational theory of the mind commit some variation on the homunculus fallacy. The idea always is to treat the brain as if there were some agent inside it using it to compute with. A typical case is David Marr (1982) who describes the task of vision as proceeding from a two-dimensional visual array on the retina to a three-dimensional description of the external world as output of the visual system. The difficulty is: Who is reading the description? Indeed, it looks throughout Marr's book, and in other standard works on the subject, as if we have to invoke a homunculus inside the system in order to treat its operations as genuinely computational.

Many writers feel that the homunculus fallacy is not really a problem, because, with Dennett (1978), they feel that the homunculus can be "discharged". The idea is this: Since the computational operations of the computer can be analyzed into progressively simpler units, until eventually we reach simple flip-flop, "yes-no", "1-0" patterns, it seems that the higher-level homunculi can be discharged with progressively stupider homunculi, until finally we reach the bottom level of a simple flip-flop that involves no real homunculus at all. The idea, in short, is that recursive decomposition will eliminate the homunculi.

It took me a long time to figure out what these people were driving at, so in case someone else is similarly puzzled I will explain an example in detail: Suppose that we have a computer that multiplies six times eight to get forty-eight. Now we ask "How does it do it?" Well, the answer might be that it adds six to itself seven times.* But if you ask "How does it add six to itself seven times?", the answer might be that, first, it converts all of the numerals into binary notation, and second, it applies a simple algorithm for operating on binary notation until finally we reach the bottom level at which the only instructions are of the form, "Print a zero, erase a one." So, for example, at the top level our intelligent homunculus says "I know how to multiply six times eight to get forty-eight". But at the next lower-level he is replaced by a stupider homunculus who says "I do not actually know how to do multiplication, but I can do addition." Below him are some stupider ones who say "We do not actually know how to do addition or multiplication, but we know how to convert decimal to binary." Below these are stupider ones who say "We do not know anything about any of this stuff, but we know how to operate on binary

symbols." At the bottom level are a whole bunch of a homunculi who just say "Zero one, zero one". All of the higher levels reduce to this bottom level. Only the bottom level really exists; the top levels are all just *as-if*.

Various authors (e.g. Haugeland (1981), Block (1990)) describe this feature when they say that the system is a syntactical engine driving a semantic engine. But we still must face the question we had before: What facts intrinsic to the system make it syntactical? What facts about the bottom level or any other level make these operations into zeros and ones? *Without a homunculus that stands outside the recursive decomposition, we do not even have a syntax to operate with*. The attempt to eliminate the homunculus fallacy through recursive decomposition fails, because the only way to get the syntax intrinsic to the physics is to put a homunculus in the physics.

There is a fascinating feature to all of this. Cognitivists cheerfully concede that the higher levels of computation, e.g. "multiply 6 times 8" are observer relative; there is nothing really there that corresponds directly to multiplication; it is all in the eye of the homunculus/beholder. But they want to stop this concession at the the lower levels. The electronic circuit, they admit, does not really multiply 6X8 as such, but it really does manipulate 0's and 1's and these manipulations, so to speak, add up to multiplication. But to concede that the higher levels of computation are not intrinsic to the physics is already to concede that the lower levels are not intrinsic either. So the homunculus fallacy is still with us.

For real computers of the kind you buy in the store, there is no homunculus problem, each user is the homunculus in question. But if we are to suppose that the brain is a digital computer, we are still faced with the question "And who is the user?" Typical homunculus questions in cognitive science are such as the following: "How does the visual system compute shape from shading; how does it compute object distance from size of retinal image?" A parallel question would be, "How do nails compute the distance they are to travel in the board from the impact of the hammer and the density of the wood?" And the answer is the same in both sorts of case: If we are talking about how the system works intrinsically neither nails nor visual systems compute anything. We as outside homunculi might describe them computationally, and it is often useful to do so. But you do not understand hammering by supposing that nails are somehow intrinsically implementing hammering algorithms and you do not understand vision by supposing the system is implementing, e.g, the shape from shading alogorithm.

VI. Third Difficulty: Syntax Has No Causal Powers.

Certain sorts of explanations in the natural sciences specify mechanisms which function causally in the production of the phenomena to be explained. This is especially common in the biological sciences. Think of the germ theory of disease, the account of photosynthesis, the DNA theory of inherited traits, and even the Darwinian theory of natural selection. In each case a causal mechanism is specified, and in each case the specification gives an explanation of the output of the mechanism. Now if you go back and look at the Primal Story it seems clear that this is the sort of explanation promised by Cognitivism. The mechanisms by which brain processes produce cognition are supposed to be computational, and by specifying the programs we will have specified the causes of cognition. One beauty of this research program, often remarked, is that we do not need to know the details of brain functioning in order to explain cognition. Brain processes provide only the hardware implementation of the cognitive programs, but the program level is where the real cognitive explanations are given. On the standard account, as stated by Newell for example, there are three levels of explanation, hardware, program, and intentionality(Newell calls this last level, the knowledge level), and the special contribution of cognitive science is made at the program level.

But if what I have said so far is correct, then there is something fishy about this whole project. I

used to believe that as a causal account the cognitivist's theory was at least false; but I now am having difficulty formulating a version of it that is coherent even to the point where it could be an empirical thesis at all. The thesis is that there are a whole lot of symbols being manipulated in the brain, 0's and 1's flashing through the brain at lightning speed and invisible not only to the naked eye but even to the most powerful electron microscope, and it is these which cause cognition. But the difficulty is that the 0's and 1's as such have no causal powers at all because they do not even exist except in the eyes of the beholder. The implemented program has no causal powers other than those of the implementing medium because the program has no real existence, no ontology, beyond that of the implementing medium. Physically speaking there is no such thing as a separate "program level".

You can see this if you go back to the Primal Story and remind yourself of the difference between the mechanical computer and Turing's human computer. In Turing's human computer there really is a program level intrinsic to the system and it is functioning causally at that level to convert input to output. This is because the human is consciously following the rules for doing a certain computation, and this causally explains his performance. But when we program the mechanical computer to peform the same computation, the assignment of a computational interpretation is now relative to us, the outside homunculi. And there is no longer a level of intentional causation intrinsic to the system. The human computer is consciously following rules, and this fact explains his behavior, but the mechanical computer is not literally following any rules at all. It is designed to behave exactly as if it were following rules, and so for practical, commercial purposes it does not matter. Now Cognitivism tells us that the brain functions like the commercial computer and this causes cognition. But without a homunculus, both commercial computer and brain have only patterns and the patterns have no causal powers in addition to those of the implementing media. So it seems there is no way Cognitivism *could* give a causal account of cognition.

However there is a puzzle for my view. Anyone who works with computers even casually knows that we often do in fact give causal explanations that appeal to the program. For example, we can say that when I hit this key I got such and such results because the machine is implementing the vi program and not the emacs program; and this looks like an ordinary causal explanation. So the puzzle is, how do we reconcile the fact that syntax, as such, has no causal powers with the fact that we do give causal explanations that appeal to programs? And, more pressingly, would these sorts of explanations provide an appropriate model for Cognitivism, will they rescue Cognitivism? Could we for example rescue the analogy with thermostats by pointing out that just as the notion "thermostat" figures in causal explanations independently of any reference to the physics of its implementation, so the notion "program", might be explanatory while equally independent of the physics.

To explore this puzzle let us try to make the case for Cognitivism by extending the Primal Story to show how the Cognitivist investigative procedures work in actual research practice. The idea, typically, is to program a commercial computer so that it simulates some cognitive capacity, such as vision or language. Then, if we get a good simulation, one that gives us at least Turing equivalence, we hypothesize that the brain computer is running the same program as the commercial computer, and to test the hypothesis we look for indirect psychological evidence, such as reaction times. So it seems that we can causally explain the behavior of the brain computer by citing the program in exactly the same sense in which we can explain the behavior of the commerical computer. Now what is wrong with that? Doesn't it sound like a perfectly legitimate scientific research program? We know that the commercial computer's conversion of input to output is explained by a program, and in the brain we discover the same program, hence we have a causal explanation.

Two things ought to worry us immediately about this project. First, we would never accept this mode of explanation for any function of the brain where we actually understood how it worked at the neurobiological level. Second we would not accept it for other sorts of system that we can simulate

computationally. To illustrate the first point, consider for example the famous account of "What the Frog's Eye Tells the Frogs Brain" (Lettvin, et al. 1959 in McCulloch, 1965) The account is given entirely in terms of the anatomy and physiology of the frog's nervous system. A typical passage, chosen at random goes like this:

> "1. Sustained Contrast Detectors.
>
> An unmyelinated axon of this group does not respond when the general illumination is turned on or off. If the sharp edge of an object either lighter or darker than the background moves into its field and stops, it discharges promptly and continues discharging, no matter what the shape of the edge or whether the object is smaller or larger than the receptive field." (p. 239).

I have never heard anyone say that all this is just the hardware implementation, and that they should have figured out which program the frog was implementing. I do not doubt that you could do a computer simulation of the frog's "bug detectors". Perhaps someone has done it. But we all know that once you understand how the frog's visual system actually works, the "computational level" is just irrelevant.

To illustrate the second point, consider simulations of other sorts of systems. I am for example typing these words on a machine that simulates the behavior of an old fashioned mechanical typewriter.* As simulations go, the word processing program simulates a typewriter better than any AI program I know of simulates the brain. But no sane person thinks: "At long last we understand how typewriters work, they are implementations of word processing programs." It is simply not the case in general that computational simulations provide causal explanations of the phenomena simulated.

So what is going on? We do not in general suppose that computational simulations of brain processes give us any explanations in place of or in addition to neurobiological accounts of how the brain actually works. And we do not in general take "X is a computational simulation of Y" to name a symmetrical relation. That is, we do not suppose that because the computer simulates a typewriter that therefore the typewriter simulates a computer. We do not suppose that because a weather program simulates a hurricane, that the causal explanation of the behavior of the hurricane is provided by the program. So why should we make an exception to these principles where unknown brain processes are concerned? Are there any good grounds for making the exception? And what kind of a causal explanation is an explanation that cites a formal program?

Here, I believe, is the solution to our puzzle. Once you remove the homunculus from the system, you are left only with a pattern of events to which someone from outside could attach a computational interpretation. Now the only sense in which the specification of the pattern by itself provides a causal explanation is that if you know that a certain pattern exists in a system you know that some cause or other is responsible for the pattern. So you can, for example, predict later stages from earlier stages. Furthermore, if you already know that the system has been programmed by an outside homunculus you can give explanations that make reference to the intentionality of the homunculus. You can say, e.g. this machine behaves the way it does because it is running vi. That is like explaining that this book begins with a bit about happy families and does not contain any long passages about a bunch of brothers, because it is Tolstoy's *Anna Karenina* and not Dostoevsky's *The Brothers Karamozov.* But you cannot explain a physical system such as a typewriter or a brain by identifying a pattern which it shares with its computational simulation, because the existence of the pattern does not explain how the system actually works *as a physical system.* In the case of cognition the pattern is at much too high a level of abstraction to explain such concrete mental (and therefore physical) events as the occurrence of a visual perception or the understanding of a sentence.

Now, I think it is obvious that we cannot explain how typewriters and hurricanes work by pointing

to formal patterns they share with their computational simulations. Why is it not obvious in the case of the brain?

Here we come to the second part of our solution to the puzzle. In making the case for Cognitivism we were tacitly supposing that the brain might be implementing algorithms for cognition, in the same sense that Turing's human computer and his mechanical computer implement algorithms. But it is precisely that assumption which we have seen to be mistaken. To see this ask yourself what happens when a system implements an algorithm. In the human computer the system consciously goes through the steps of the algorithm, so the process is both causal and logical; logical, because the algorithm provides a set of rules for deriving the output symbols from the input symbols; causal, because the agent is making a conscious effort to go through the steps. Similarly in the case of the mechanical computer the whole system includes an outside homunculus, and with the homunculus the system is both causal and logical; logical because the homunculus provides an interpretation to the the processes of the machine; and causal because the hardware of the machine causes it to go through the processes. But these conditions cannot be met by the brute, blind, nonconscious neurophysiological operations of the brain. In the brain computer there is no conscious intentional implementation of the algorithm as there is in the human computer, but there can't be any nonconscious implementation as there is in the mechanical computer either, because that requires an outside homunculus to attach a computational interpretation to the physical events. The most we could find in the brain is a pattern of events which is formally similar to the implemented program in the mechanical computer, but that pattern, as such, has no causal powers to call its own and hence explains nothing.

In sum, the fact that the attribution of syntax identifies no further causal powers is fatal to the claim that programs provide causal explanations of cognition. To explore the consequences of this let us remind ourselves of what Cognitivist explanations actually look like. Explanations such as Chomsky's account of the syntax of natural languages or Marr's account of vision proceed by stating a set of rules according to which a symbolic input is converted into a symbolic output. In Chomsky's case, for example, a single input symbol, S, is converted into any one of a potentially infinite number of sentences by the repeated application of a set of syntactical rules. In Marr's case, representations of a two dimensional visual array are converted into three dimensional "descriptions" of the world in accordance with certain algorithms. Marr's tripartite distinction between the computational task, the algorithmic solution of the task and the hardware implementation of the algorithm, has (like Newell's distinctions) become famous as a statement of the general pattern of the explanation.

If you take these explanations naively, as I do, it is best to think of them as saying that it is just as if a man alone in a room were going through a set of steps of following rules to generate English sentences or 3D descriptions, as the case might be. But now, let us ask what facts in the real world are supposed to correspond to these explanations as applied to the brain. In Chomsky's case, for example we are not supposed to think that the agent consciously goes through a set of repeated applications of rules; nor are we supposed to think that he is unconsciouly thinking his way through the set of rules. Rather the rules are "computational" and the brain is carrying out the computations. But what does that mean? Well, we are supposed to think that it is just like a commercial computer. The sort of thing that corresponds to the ascription of the same set of rules to a commercial computer is supposed to correspond to the ascription of those rules to the brain. But we have seen that in the commercial computer the ascription is always observer relative, the ascription is made relative to a homunculus who assigns computational interpretations to the hardware states. Without the homunculus there is no computation, just an electronic circuit. So how do we get computation into the brain without a homunculus? As far as I know neither Chomsky nor Marr ever addressed the question or even thought there was such a question. But without a homunculus there is no explanatory power to the postulation of the program states. There is just a physical mechanism, the brain, with its various real physical and

physical/mental causal levels of description.

VII. Fourth Difficulty: The Brain Does Not Do Information Processing.

In this section I turn finally to what I think is, in some ways, the central issue in all of this, the issue of information processing. Many people in the "cognitive science" scientific paradigm will feel that much of my discussion is simply irrelevant and they will argue against it as follows:

> "There is a difference between the brain and all of these other systems you have been describing and this difference explains why a computational simulation in the case of the other systems is a mere simulation whereas in the case of the brain a computational simulation is actually duplicating and not merely modeling the functional properties of the brain. The reason is that the brain, unlike these other systems, is an *information processing* system. And this fact about the brain is, in your words, "intrinsic". It is just a fact about biology that the brain functions to process information, and since we can also process the same information computationally, computational models of brain processes have a different role altogether from computational models of, for example, the weather.
>
> So there is a well defined research question: "Are the computational procedures by which the brain processes information the same as the procedures by which computers process the same information?"

What I just imagined an opponent saying embodies one of the worst mistakes in cognitive science. The mistake is to suppose that in the sense in which computers are used to process information, brains also process information. To see that that is a mistake contrast what goes on in the computer with what goes on in the brain. In the case of the computer, an outside agent encodes some information in a form that can be processed by the circuitry of the computer. That is, he or she provides a syntactical realization of the information that the computer can implement in, for example, different voltage levels. The computer then goes through a series of electrical stages that the outside agent can interpret both syntactically and semantically even though, of course, the hardware has no intrinsic syntax or semantics: It is all in the eye of the beholder. And the physics does not matter provided only that you can get it to implement the algorithm. Finally, an output is produced in the form of physical phenomena which an observer can interpret as symbols with a syntax and a semantics.

But now contrast that with the brain. In the case of the brain, none of the relevant neurobiological processes are observer relative (though of course, like anything they can be described from an observer relative point of view) and the specificity of the neurophysiology matters desperately. To make this difference clear, let us go through an example. Suppose I see a car coming toward me. A standard computational model of vision will take in information about the visual array on my retina and eventually print out the sentence, "There is a car coming toward me". But that is not what happens in the actual biology. In the biology a concrete and specific series of electro-chemical reactions are set up by the assault of the photons on the photo receptor cells of my retina, and this entire process eventually results in a concrete visual experience. The biological reality is not that of a bunch of words or symbols being produced by the visual system, rather it is a matter of a concrete specific conscious visual event; this very visual experience. Now that concrete visual event is as specific and as concrete as a hurricane or the digestion of a meal. We can, with the computer, do an information processing model of that event or of its production, as we can do an information model of the weather, digestion or any other phenomenon, but the phenomena themselves are not thereby information processing systems.

In short, the sense of information processing that is used in cognitive science, is at much too high a level of abstraction to capture the concrete biological reality of intrinsic intentionality. The "informa-

tion" in the brain is always specific to some modality or other. It is specific to thought, or vision, or hearing, or touch, for example. The level of information processing which is described in the cognitive science computational models of cognition, on the other hand, is simply a matter of getting a set of symbols as output in response to a set of symbols as input.

We are blinded to this difference by the fact that the same sentence, "I see a car coming toward me", can be used to record both the visual intentionality and the output of the computational model of vision. But this should not obscure from us the fact that the visual experience is a concrete event and is produced in the brain by specific electro-chemical biological processes. To confuse these events and processes with formal symbol manipulation is to confuse the reality with the model. The upshot of this part of the discussion is that in the sense of "information" used in cognitive science it is simply false to say that the brain is an information processing device.

VIII. Summary of the Argument.

This brief argument has a simple logical structure and I will lay it out:

1. On the standard textbook definition, computation is defined syntactically in terms of symbol manipulation.

2. But syntax and symbols are not defined in terms of physics. Though symbol tokens are always physical tokens, "symbol" and "same symbol" are not defined in terms of physical features. Syntax, in short, is not intrinsic to physics.

3. This has the consequence that computation is not discovered in the physics, it is assigned to it. Certain physical phenomena are assigned or used or programmed or interpreted syntactically. Syntax and symbols are observer relative.

4. It follows that you could not *discover* that the brain or anything else was intrinsically a digital computer, although you could assign a computational interpretation to it as you could to anything else. The point is not that the claim "The brain is a digital computer" is false. Rather it does not get up to the level of falsehood. It does not have a clear sense. You will have misunderstood my account if you think that I am arguing that it is simply false that the brain is a digital computer. The question "Is the brain a digital computer?" is as ill defined as the questions "Is it an abacus?", "Is it a book?", or "Is it a set of symbols?", "Is it a set of mathematical formulae?"

5. Some physical systems facilitate the computational use much better than others. That is why we build, program, and use them. In such cases we are the homunculus in the system interpreting the physics in both syntactical and semantic terms.

6. But the causal explanations we then give do not cite causal properties different from the physics of the implementation and the intentionality of the homunculus.

7. The standard, though tacit, way out of this is to commit the homunculus fallacy. The humunculus fallacy is endemic to computational models of cognition and cannot be removed by the standard recursive decomposition arguments. They are addressed to a different question.

8. We cannot avoid the foregoing results by supposing that the brain is doing "information processing". The brain, as far as its intrinsic operations are concerned, does no information processing. It is a specific biological organ and its specific neurobiological processes cause specific forms of intentionality. In the brain, intrinsically, there are neurobiological processes and sometimes they cause consciousness. But that is the end of the story.*

References

Block, Ned (1990, forthcoming). "The Computer Model of the Mind".

Boolos, George S. and Jeffrey, Richard C. (1989). *Computability and Logic*. Cambridge University Press, Cambridge Mass.

Dennett, Daniel C. (1978). *Brainstorms: Philosophical Essays on Mind and Psychology*. MIT Press, Cambridge, Mass.

Feigenbaum, E.A. and Feldman, J. ed. (1963). *Computers and Thought*. McGraw-Hill Company, New York and San Francisco.

Fodor, J. (1975) *The Language of Thought*. Thomas Y. Crowell, New York.

Haugeland, John ed. (1981). *Mind Design*. MIT Press, Cambridge, Mass.

Johnson-Laird, P. N. (1988). *The Computer and the Mind*. Harvard University Press, Cambridge Mass.

Kuffler, Stephen W. and Nicholls, John G. (1976). *From Neuron to Brain*. Sinauer Associates, Sunderland, Mass.

Lettvin, J.Y., Maturana, H.R., McCulloch, W.S., and Pitts, W.H. (1959). What the Frog's Eye Tells the Frog's Brain. *Proceedings of the Institute of Radio Engineers*, 47, 1940-1951, reprinted in McCulloch, 1965, pp. 230-255.

Marr, David (1982). *Vision*. W.H. Freeman and Company, San Francisco.

McCulloch, Warren S. (1965). *The Embodiments of Mind*. MIT Press, Cambridge, Mass.

Pylyshyn (1985). *Computation and Cognition*. MIT Press, Cambridge, Mass.

Searle, John R. (1980). "Minds, Brains and Programs", *The Behavioral and Brain Sciences*. 3, pp. 417-424.

Searle, John R. (1984). *Minds, Brains and Science*. Harvard University Press, Cambridge, Mass.

Sharples, M., Hogg, D., Hutchinson, C., Torrance, S., and Young, D. (1988). *Computers and Thought*. MIT Press, Cambridge, Mass. and London.

Shepherd, Gordon M. (1983). *Neurobiology*. Oxford University Press, New York and Oxford.

Turing, Alan (1950). "Computing Machinery and Intelligence." *Mind*, 59, 433-460.