38 The Critique of Cognitive Reason

John R. Searle

Introduction: The Shaky Foundations of Cognitive Science

For over a decade, really since the beginnings of the discipline, I have been a practicing "cognitive scientist." In this period I have seen much valuable work and progress in the field. However, as a discipline, cognitive science suffers from the fact that several of its most cherished foundational assumptions are mistaken. It is possible to do good work on the basis of false assumptions, but it is more difficult than need be; and in this chapter I want to expose and refute some of those false assumptions. They derive from the pattern of mistakes that I described earlier.

Not everybody in cognitive science agrees on the foundational principles, but there are certain general features of the mainstream that deserve a separate statement. If I were a mainstream cognitive scientist, here is what I would say:

Neither the study of the brain as such nor the study of consciousness as such is of much interest and importance to cognitive science. The cognitive mechanisms we study are indeed implemented in the brain, and some of them find a surface expression in the consciousness, but our interest is in the intermediate level where the actual cognitive processes are inaccessible to consciousness. Though in fact implemented in the brain, they could have been implemented in an indefinite number of hardware systems. Brains are there, but inessential. The processes which explain cognition are unconscious not only in fact, but in principle. For example, Chomsky's rules of universal grammar (1986), or Marr's rules of vision (1982), or Fodor's language of thought (1975) are not the sort of phenomena that could become conscious. Furthermore, these processes are all computational. The basic assumption behind cognitive science is that the brain is a computer and mental processes are computational. For that reason many of us think that artificial intelligence (AI) is the heart of cognitive science. There is some dispute among us as to whether or not the brain is a digital computer

From J. Searle, *The rediscovery of the mind* (1992). Cambridge, MA: MIT Press. Reprinted by permission.

of the old-fashioned von Neumann variety or whether it is a connectionist machine. Some of us, in fact, manage to have our cake and eat it too on this question, because we think the serial processes in the brain are implemented by a parallel connectionist system (e.g., Hobbs 1990). But nearly all of us agree on the following: Cognitive mental processes are unconscious; they are, for the most part, unconscious in principle, and they are computational.

I disagree with just about every substantive claim made in the previous paragraph, and I have already criticized some of them in earlier chapters, most notably the claim that there are mental states that are deep unconscious. The main aim of this chapter is to criticize certain aspects of the computational claim.

I think it will help explain what makes the research program seem so implausible to me if we nail the question down to a concrete example right away: In AI great claims have been made for programs run on SOAR.[1] Strictly speaking, SOAR is a type of computer architecture and not a program, but programs implemented on SOAR are regarded as promising examples of AI. One of these is embodied in a robot that can move blocks on command. So, for example, the robot will respond appropriately to the command "Pick up a cube-shaped block and move it three spaces to the left." To do this, it has both optical sensors and robot arms, and the system works because it implements a set of formal symbol manipulations that are connected to transducers that receive inputs from the optical sensors and send outputs to the motor mechanisms. But my problem is: What has all that got to do with actual human behavior? We know for example many of the details about how a human being does it in real life. First, she must be *conscious*. Furthermore she must *hear and understand* the order. She must *consciously see* the blocks, she must *decide* to carry out the command, and then she must perform the *conscious voluntary intentional action* of moving the blocks. Notice that these claims all support counterfactuals: for example, no consciousness, no movement of blocks. Also we know that all this mental stuff is caused by and realized in the neurophysiology. So before we ever get started on computer modeling, we know that there are two sets of levels: mental levels, many of them conscious, and neurophysiological levels.

Now where are the formal symbol manipulations supposed to fit into this picture? This is a fundamental foundational question in cognitive science, but you would be amazed at how little attention is paid to it. The absolutely crucial question for any computer model is, "How *exactly* does the model relate to the reality being modeled?" But unless you read skeptical critics like the present author, you will find very little discussion of this issue. The general answer, which is supposed to evade the demand for more detailed specific answers, is that between the level of intentionality in the human (what Newell [1982] calls "the knowledge level") and the various neurophysiological levels, there is an interme-

diate level of formal symbol manipulation. Now our question is, empirically speaking, what could that possibly mean?

If you read books about the brain (say, Shepherd 1983; or Bloom and Lazerson 1988), 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 1984), they tell you that what the brain books describe is really the same as what the computation 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.

Strong AI, Weak AI, and Cognitivism

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."[2]

Let us begin our investigation of this claim by distinguishing 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?

In this chapter, I will be addressing 1, and not 2 or 3. In earlier writings (Searle 1980a, 1980b, and 1984), I have given a negative answer to 2. Because programs are defined purely formally or syntactically, and because 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 1980b). 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 demonstrably "No."

The answer to 3 seems to me equally demonstrably "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 given Church's thesis that 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 com-

puter 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 of 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 chapter is about cognitivism.

The Primal Story

Earlier I gave a preliminary statement of the assumptions of mainstream cognitive science, and now I want to continue 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 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 that 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, why was this result so exciting? Well, what made it send shivers up and down the spines of a whole generation of young workers in artificial intelligence was 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 (1950), 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 that there might be a whole lot of other mental processes going on in my brain nonconsciously that 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 processes 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.[3]

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.

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 the program, whether by the brain or by the mechanical computer. Notice also the idea that we might *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. I have discussed the reasons for this urge earlier. 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.

Second, it is also assumed that the question of 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, because we are talking about matters of hard scientific fact. Indeed, I think many people who work in this field would doubt that the question I am addressing is an appropriate philosophic question at all. "Is the brain really a digital computer?" is no more a philosophical question than "Is the neurotransmitter at neuromuscular junctions really acetylcholene?"

Even people who are unsympathetic to cognitivism, such as Penrose (1989) and Dreyfus (1972), seem to treat it as a straightforward factual issue. They do not seem to be worried about the question of 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 that

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 an algorithm? What exactly is a computational process? Under what physical conditions exactly are two systems implementing the same program?

The Definition of Computation

As 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 one square to the left, or it can shift the tape one square to the right. It is controlled by a program of instructions 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 that *could be used to* function as 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, 39).

Similar remarks are made by most of the people who write on this topic. For example, Ned 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 need not use electricity at all, but we can

use 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 that we can also treat as if it were a 0 or a 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!" (1984, 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 water pipes, 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.

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 carburetors and thermostats. Just as carburetors 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 carburetors and thermostats are defined in terms of the production of certain *physical* effects. That is why, for example, nobody says you can make carburetors 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 that 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 that 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 that 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, it is hard to see how to avoid the following results:

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 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.

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 "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 do not think that the problem of universal realizability is a serious one. I think it is 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. All these will produce the result that the pattern is not enough. There must be a causal structure sufficient to warrant counterfactuals. 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 is not just a sign that the processes are abstract, but that they are not intrinsic to the system at all. They depend on an interpretation from outside.* We were looking for some facts of the matter that 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 notions 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-editor program on my 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 occurring. It requires the assignment of a computational interpretation by some agent. Analogously, we might discover in nature objects that had the same sort of shape as chairs and that could therefore be used as chairs; but we could not discover objects in nature that were functioning as chairs, except relative to some agents who regarded them or used them as chairs.

To understand this argument fully, it is essential to understand the distinction between features of the world that are *intrinsic* and features that are *observer relative*. The expressions "mass," "gravitational attraction," and "molecule" name features of the world that are intrinsic. If all observers and users cease to exist, the world still contains mass, gravitational attraction, and molecules. But expressions such as "nice day for a picnic," "bathtub," and "chair" do not name intrinsic features of reality. Rather, they name objects by specifying some feature that has been assigned to them, some feature that is relative to observers and users. If there had never been any users or observers, there would still be mountains, molecules, masses, and gravitational attraction. But if there had never been any users or observers, there would be no such features as being a nice day for a picnic, or being a chair or a bathtub. The assignment of observer-relative features to intrinsic features of the world is not arbitrary. Some intrinsic features of the world facilitate their use as chairs and bathtubs, for example. But the feature of being a chair or a bathtub or a nice day for a picnic is a feature that only exists relative to users and observers. The point I am making here, and the essence of this argument, is that on the standard definitions of computation, computational features are observer relative. They are not intrinsic. The argument so far, then, can be summarized as follows:

The aim of natural science is to discover and characterize features that are intrinsic to the natural world. By its own definitions of computation and cognition, there is no way that computational cognitive science could ever be a natural science, because computation is not an intrinsic feature of the world. It is assigned relative to observers.[4] . . .

Further 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 the 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 as 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, for example, a printout, 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 electrochemical 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. That concrete visual event is as specific and as concrete as a hurricane or the digestion of a meal. We can, with the computer, make an information processing model of that event or of its production, as we can make an information processing 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 "information" 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 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 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 the fact that the visual experience is a concrete conscious event and is produced in the brain by specific electrochemical 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. . . .

Notes

1. SOAR is a system developed by Alan Newell and his colleagues at Carnegie Mellon University. The name is an acronym for "State, Operator, And Result." For an account see Waldrop 1988.

2. This view is announced and defended in a large number of books and articles many of which appear to have more or less the same title, e.g., *Computers and Thought* (Feigenbaum and Feldman, eds., 1963), *Computers and Thought* (Sharples et al. 1988), *The Computer and the Mind* (Johnson-Laird 1988), *Computation and Cognition* (Pylyshyn 1984), "The Computer Model of the Mind" (Block 1990), and of course, "Computing Machinery and Intelligence" (Turing 1950).

3. This whole research program has been neatly summarized by Gabriel Segal (1991) as follows: "Cognitive science views cognitive processes as computations in the brain. And computation consists in the manipulation of pieces of syntax. The content of the syntactic objects, if any, is irrelevant to the way they get processed. So, it seems, content can

figure in cognitive explanations only insofar as differences in content are reflected in differences in the brain's syntax" (p. 463).

4. Pylyshyn comes very close to conceding precisely this point when he writes, "The answer to the question what computation is being performed requires discussion of semantically interpreted computational states" (1984, 58). Indeed. And who is doing the interpreting?

References

Block, N. (1990). The computer model of the mind. In D. N. Osherson and E. E. Smith, eds., *Thinking: An invitation to cognitive science*, vol. 3. Cambridge, MA: MIT Press.

Bloom, F. E., and A. Lazerson (1988). *Brain, mind, and behavior*, 2d ed. New York: W. H. Freeman.

Boolos, G. S., and R. C. Jeffrey (1989). *Computability and logic*. Cambridge: Cambridge University Press.

Chomsky, N. (1986). *Knowledge of language: Its nature, origin and use*. New York and Philadelphia: Praeger Special Studies.

Dreyfus, H. L. (1972). *What computers can't do*. New York: Harper and Row.

Feigenbaum, E. A., and J. Feldman, eds. (1963). *Computers and thought*. New York: McGraw-Hill.

Fodor, J. A. (1975). *The language of thought*. New York: Crowell.

Hobbs, J. R. (1990). Matter, levels, and consciousness. *Behavioral and Brain Sciences* 13, 610–611.

Johnson-Laird, P. N. (1988). *The computer and the mind*. Cambridge, MA: Harvard University Press.

Marr, D. (1982). *Vision*. San Francisco: W. H. Freeman.

Newell, A. (1982). The knowledge level. *Artificial Intelligence* 18, 87–127.

Penrose, R. (1989). *The emperor's new mind*. Oxford: Oxford University Press.

Pylyshyn, Z. W. (1984). *Computation and cognition: Toward a foundation for cognitive science*. Cambridge, MA: MIT Press.

Searle, J. R. (1980a). Intrinsic intentionality: Reply to criticisms of Minds, Brains, and Programs. *Behavioral and Brain Sciences* 3, 450–456.

Searle, J. R. (1980b). Minds, brains, and programs. *Behavioral and Brain Sciences* 3, 417–424.

Searle, J. R. (1984). *Minds, brains, and science: The 1984 Reith lectures*. Cambridge, MA: Harvard University Press.

Segal, G. (1991). Review of Garfield, J., *Belief in psychology. Philosophical Review* 100, 463–466.

Sharples, M., D. Hogg, C. Hutchinson, S. Torrence, and D. Young (1988). *Computers and thought: A practical introduction to artificial intelligence.* Cambridge, MA: MIT Press.

Shepherd, G. M. (1983). *Neurobiology.* Oxford: Oxford University Press.

Turing, A. (1950). Computing machinery and intelligence. *Mind* 59, 433–460.

Waldrop, M. M. (1988). Toward a unified theory of cognition; and SOAR: A unified theory of cognition. *Science* 241, 27–29, 296–298.