24.09x Minds and Machines

Ned Block, "The mind as the software of the brain"

Excerpts from Ned Block, "The mind as the software of the brain" (D. N. Osherson, L. Gleitman, S. M. Kosslyn, S. Smith and S. Sternberg, eds., *An Invitation to Cognitive Science*, MIT Press, 1995)[1]

Cognitive scientists, Block observes, "often say that the mind is the software of the brain". But what do they mean? Block's paper attempts to answer that question. In this class three parts of Block's lengthy paper will be particularly useful.

PART I

First, the part on the *Turing test*.

> One approach to the mind has been to avoid its mysteries by simply *defining* the mental in terms of the behavioral. This approach has been popular among thinkers who fear that acknowledging mental states that do not reduce to behavior would make psychology unscientific, because unreduced mental states are not intersubjectively accessible in the manner of the entities of the hard sciences. "Behaviorism", as the attempt to reduce the mental to the behavioral is called, has often been regarded as refuted, but it periodically reappears in new forms.
>
> Behaviorists don't define the mental in terms of just plain *behavior*, since after all something can be intelligent even if it has never had the chance to exhibit its intelligence. Behaviorists define the mental not in terms of behavior, but rather behavioral *dispositions*, the tendency to emit certain behaviors given certain stimuli. It is important that the stimuli and the behavior be specified non-mentalistically. Thus, intelligence could not be defined in terms of the disposition to give sensible responses to questions, since that would be to define a mental notion in terms of another mental notion (indeed, a closely related one). To see the difficulty of behavioristic analyses, one has to appreciate how mentalistic our ordinary behavioral descriptions are. Consider, for example, *throwing*. A series of motions that constitute throwing if produced by one mental cause might be a dance to get the ants off if produced by another.
>
> An especially influential behaviorist definition of intelligence was put forward by A. M. Turing (1950). Turing, one of the mathematicians who cracked the German code during World War II, formulated the idea of the universal Turing machine, which contains, in mathematical form, the essence of the

[1] Full text on Ned Block's website: http://www.nyu.edu/gsas/dept/philo/faculty/block/papers/msb.html.

programmable digital computer. Turing wanted to define intelligence in a way that applied to both men and machines, and indeed, to anything that is intelligent. His version of behaviorism formulates the issue of whether machines could think or be intelligent in terms of whether they could pass the following test: a judge in one room communicates by teletype (This was 1950!) with a computer in a second room and a person in a third room for some specified period. (Let's say an hour.) The computer is intelligent if and only if the judge cannot tell the difference between the computer and the person. Turing's definition finessed the difficult problem of specifying non-mentalistically the behavioral dispositions that are characteristic of intelligence by bringing in the discrimination behavior of a human judge. And the definition generalizes. *Anything* is intelligent just in case it can pass the Turing test.

Turing suggested that we replace the concept of intelligence with the concept of passing the Turing test. But what is the replacement *for*? If the purpose of the replacement is practical, the Turing test is not enormously useful. If one wants to know if a machine does well at playing chess or diagnosing pneumonia or planning football strategy, it is better to see how the machine performs in action than to make it take a Turing test. For one thing, what we care about is that it do well at detecting pneumonia, not that it do it in a way indistinguishable from the way a person would do it. So if it does the job, who cares if it doesn't pass the Turing test?

A second purpose might be utility for theoretical purposes. But machines that can pass the Turing test such as Weizenbaum's ELIZA (see below) have been dead ends in artificial intelligence research, not exciting beginnings.

A third purpose, the one that comes closest to Turing's intentions, is the purpose of *conceptual clarification*. Turing was famous for having formulated a precise mathematical concept that he offered as a replacement for the vague idea of mechanical computability. The precise concept (computability by a Turing machine) did everything one would want a precise concept of mechanical computability to do. No doubt, Turing hoped that the Turing test conception of intelligence would yield everything one would want from a definition of intelligence without the vagueness of the ordinary concept.

Construed as a proposal about how to make the concept of intelligence precise, there is a gap in Turing's proposal: we are not told how the judge is to be chosen. A judge who was a leading authority on genuinely intelligent machines might know how to tell them apart from people. For example, the

expert may know that current intelligent machines get certain problems right that people get wrong. Turing acknowledged this point by jettisoning the claim that being able to pass the Turing Test is a necessary condition of intelligence, weakening his claim to: passing the Turing Test is a sufficient condition for intelligence. He says "May not machines carry out something which ought to be described as thinking but which is very different from what a man does? This objection is a very strong one, but at least we can say that if, nevertheless, a machine can be constructed to play the imitation game satisfactorily, we need not be troubled by this objection".[2] In other words, a machine that *does pass* is necessarily intelligent, even if some intelligent machines fail.

But the problem of how to specify the qualities of the judge goes deeper than Turing acknowledges, and compromises the Turing test as a sufficient condition too. A stupid judge, or one who has had no contact with technology, might think that a radio was intelligent. People who are naive about computers are amazingly easy to fool, as was demonstrated in the First Turing Test at the Boston Computer Museum in 1991. A version of Weizenbaum's ELIZA (described in the next paragraph) was classified as human by five of ten judges. The test was "restricted" in that the computer programmers were given specific topics that their questions would be restricted to, and the judges were forbidden to ask "tricky" questions. For example, if the topic were Washington D.C., a judge was not supposed to ask questions like "Is Washington D.C. bigger than a bread box". However, the winning program's topic was "whimsical conversation", a "smart-aleck" way of interacting that all the judges fell in with immediately, and one that would have had the same effect, even without set topics. Further, the restrictions to non-tricky questions weren't enforced. (I speak as one of the referees who failed to enforce them.) For the most part, the computer-naive judges didn't really know how to formulate a tricky question. Thus the variant of ELIZA may be regarded as having done well in an unrestricted Turing test. Variants on ELIZA (all written by Joseph Weintraub) also were victorious in the Second and Third Turing Tests. The upshot is that a completely stupid computer program has *already* been shown to pass a reasonable facsimile of a Turing Test.

ELIZA is a simple program written by Joseph Weizenbaum that in one version imitates a psychiatrist. It employs a group of simple but effective strategies.

---

[2] p. 3 of the 24.09x version of "Computing machinery and intelligence".

For example, it looks for "key words" on a list supplied by the programmer, e.g., `I', `you', `alike', `father', and `everybody'. The words are ordered; for example, `father' comes before `everybody', so if you type in "My father is afraid of everybody," the machine will spit back one of its "father" responses, such as "WHAT ELSE COMES TO MIND WHEN YOU THINK OF YOUR FATHER?" If you type in "Everybody laughed at me," you will get one of its responses to `everybody', such as "WHO IN PARTICULAR ARE YOU THINKING OF?" It also has techniques that simultaneously transform `you' into `I', and `me' into `you' so that if you type in "You don't agree with me," it can reply: "WHY DO YOU THINK THAT I DON'T AGREE WITH YOU?" It also *stores* sentences containing certain key words such as `my'. If your *current* input contains no key words, but if you earlier said "My boyfriend made me come here," the program will say "DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE?" It also has stock responses to `Parlez vous Francais' (or any other sentence with `Francais' in it), namely "I SPEAK ONLY ENGLISH". If all else fails, the machine has a list of last ditch responses such as "WHO IS THE PSYCHIATRIST HERE, YOU OR ME?"

This type of program is easy to trip up if you know how it works. For example, in the First Turing Test, one judge said "You're trying to make this difficult for me aren't you?" and the program replied roughly as follows: "Would you prefer it if it was not true that you're trying to make this difficult for you aren't you?" As Shieber points out, the pronoun transposition rules can't handle "tag questions" that end with, e.g. "aren't you?"

The point that this program illustrates is that a simple program can be extraordinarily successful in activities akin to passing the Turing Test…Weizenbaum's program is not sophisticated or complex by current standards…yet this type of program is better at passing the Turing Test than anything else written to date, as is shown by the three victories in a row in the Turing Tests mentioned above. Imagine how convincing a program would be produced if the Defence budget were devoted to this task for a year! But even if a high budget government initiative produced a program that was good at passing the Turing Test, if the program was just a bundle of tricks like the Weizenbaum program, with question types all thought of in advance, and canned responses placed in the machine, the machine would not be intelligent.

One way of dealing with the problem of the specification of the judge is to make some sort of characterization of the mental qualities of the judge part of the formulation of the Turing Test. For example, one might specify that the judge be moderately knowledgeable about computers and good at thinking, or better, good at thinking about thinking. But including a specification of the mental qualities of the judge in the description of the test will ruin the test as a way of *defining* the concept of intelligence in non-mentalistic terms. Further, if we are going to specify that the judge be good at thinking about thinking, we might just as well give up on having the judge judge which contestants are humans or machines and just have the judge judge which contestants think. And then what the idea of the Turing Test would amount to is: a machine thinks if our best thinkers (about thinking) think it thinks. Although this sounds like a platitude, it is actually false. For even our best thinkers are fallible. The most that can be claimed is that if our best thinkers think that something thinks, then it is rational for us to believe that it does.

I've made much of the claim that judges can be fooled by a mindless machine that is just a bag of tricks. "But," you may object, "How do we know that *we* are not just a bag of tricks?" Of course, in a sense perhaps we are, but that isn't the sense relevant to what is wrong with the Turing Test. To see this point, consider the ultimate in unintelligent Turing Test passers, a hypothetical machine that contains *all conversations of a given length* in which the machine's replies make sense. Let's stipulate that the test lasts one hour. Since there is an upper bound on how fast a human typist can type, and since there are a finite number of keys on a teletype, there is an upper bound on the "length" of a Turing Test conversation. Thus there are a finite (though more than astronomical) number of different Turing Test conversations, and there is no contradiction in the idea of *listing them all*.

Let's call a string of characters that can be typed in an hour or less a "typable" string. In principle, all typable strings could be generated, and a team of intelligent programmers could throw out all the strings which cannot be interpreted as a conversation in which at least one party (say the second contributor) is making sense. The remaining strings (call them the sensible strings) could be stored in an hypothetical computer (say, with marks separating the contributions of the separate parties), which works as follows. The judge types in something. Then the machine locates a string that starts with the judge's remark, spitting back its next element. The judge then types something else. The machine finds a string that begins with the judge's first

contribution, followed by the machine's, followed by the judge's next contribution (the string will be there since all sensible strings are there), and then the machine spits back its fourth element, and so on. (We can eliminate the simplifying assumption that the judge speaks first by recording *pairs* of strings; this would also allow the judge and the machine to talk at the same time.) Of course, such a machine is only logically possible, not physically possible. The number of strings is too vast to exist, and even if they could exist, they could never be accessed by any sort of a machine in anything like real time. But since we are considering a proposed definition of intelligence that is supposed to capture the *concept* of intelligence, conceptual possibility will do the job. If the concept of intelligence is supposed to be exhausted by the ability to pass the Turing Test, then even a universe in which the laws of physics are very different from ours should contain exactly as many unintelligent Turing test passers as married bachelors, namely zero.

Note that the choice of one hour as a limit for the Turing Test is of no consequence, since the procedure just described works for *any* finite Turing Test.

The following variant of the machine may be easier to grasp. The programmers start by writing down all typable strings, call them $A_1...A_n$. Then they think of *just one* sensible response to each of these, which we may call $B_1...B_n$. (Actually, there will be fewer Bs than As because some of the As will take up the entire hour.) The programmers may have an easier time of it if they think of themselves as simulating some definite personality, say my Aunt Bubbles, and some definite situation, say Aunt Bubbles being brought into the teletype room by her strange nephew and asked to answer questions for an hour. So each of the Bs will be the sort of reply Aunt Bubbles would give to the preceeding A. For example, if A_{73} is "Explain general relativity", B_{73} might be "Ask my nephew, he's the professor." What about the judge's replies to each of the Bs? The judge can give any reply up to the remaining length limit, so below each of the Bs, there will sprout a vast number of Cs (vast, but fewer than the number of Bs, since the time remaining has decreased). The programmers' next task is to produce just one D for each of the Cs. So if the B just mentioned is followed by a C which is "xyxyxyxyxyxyxy!" (Remember, the judge doesn't have to make sense), the programmers might make the following D "My nephew warned me that you might type some weird messages".

Think of conversations as paths downward through a tree, starting with an A_i from the judge, a reply, B_i from the machine, and so on. See Figure 1. For each A_i-B_i-C^i_j that is a beginning to a conversation, the programmers must produce a D that makes sense given the A, B, and C that precede it.

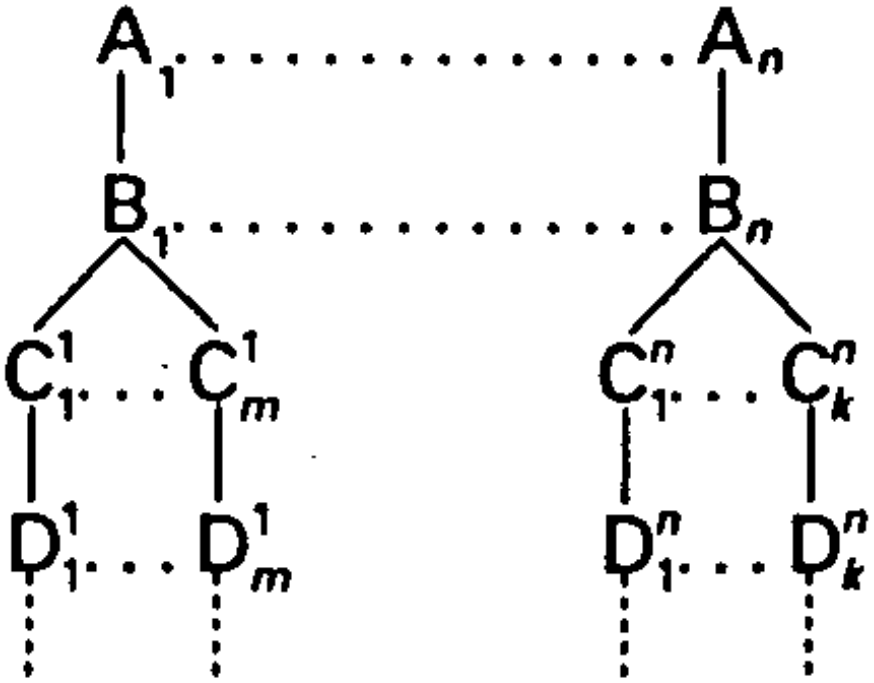

Figure 1: A conversation is any path from the top to the bottom.

The machine works as follows. The judge goes first. Whatever the judge types in (typos and all) is one of A_1...A_n. The machine locates the particular A, say A_{2398}, and then spits back B_{2398}, a reply chosen by the programmers to be appropriate to A_{2398}. The judge types another message, and the machine again finds it in the list of Cs that sprout below B_{2398}, and then spits back the pre-recorded reply (which takes into account what was said in A_{2398} and B_{2398}). And so on. Though the machine can do as well in the one hour Turing Test as Aunt Bubbles, it *has the intelligence of a juke-box*. Every clever remark it produces was specifically thought of by the programmers as a response to the previous remark of the judge in the context of the previous conversation.

Though this machine is too big to exist, there is nothing incoherent or contradictory about its specification, and so it is enough to refute the behaviorist interpretation of the Turing Test that I have been talking about.

Note that there is an upper bound on how long any particular Aunt Bubbles machine can go on in a Turing Test, a limit set by the length of the strings it has been given. Of course *real people* have their upper limits too, given that real people will eventually quit or die. However, there is a very important difference between the Aunt Bubbles machine and a real person. We can define `competence' as idealized performance. Then, relative to appropriate idealizations, it may well be that real people have an infinite competence to go on. That is, if humans were provided with unlimited memory and with motivational systems that give passing the Turing test infinite weight, they could go on for ever (at least according to conventional wisdom in cognitive science). This is definitely not the case for the Aunt Bubbles machine. But this difference provides no objection to the Aunt Bubbles machine as a refutation of the Turing Test conception of intelligence, because the notion of competence is not behavioristically acceptable, requiring as it does for its specification, a distinction among components of the mind. For example, the mechanisms of thought must be distinguished from the mechanisms of memory and motivation.

"But," you may object, "isn't it rather chauvinist to assume that a machine must process information in just the way *we* do to be intelligent?" Answer: Such an assumption would indeed be chauvinist, but I am not assuming it. The point against the Turing Test conception of intelligence is not that the Aunt Bubbles machine wouldn't process information the way we do, but rather that the way it does process information is unintelligent despite its performance in the Turing Test.

Ultimately, the problem with the Turing test for theoretical purposes is that it focuses on performance rather than on competence. Of course, performance is evidence for competence, but the core of our understanding of the mind lies with mental competence, not behavioral performance. The behviorist cast of mind that leads to the Turing test conception of intelligence also leads to labeling the sciences of the mind as "the behavioral sciences". But as Chomsky has pointed out, that is like calling physics the science of meter readings.

PART II

The second pertinent part of Block's paper is about intelligence and intentionality.

Our discussion so far has centered on the computational approach to one aspect of the mind, intelligence. But there is a different aspect of the mind

that we have not yet discussed, one that has a very different relation to computational ideas, namely intentionality.

For our purposes, we can take intelligence to be a capacity, a capacity for various intelligent activities such as solving mathematics problems, deciding whether to go to graduate school, and figuring out how spaghetti is made. (Notice that this analysis of intelligence as a capacity to solve, figure out, decide, and the like, is a mentalistic analysis, not a behaviorist analysis.)

Intentionality is aboutness. Intentional states represent the world as being a certain way. The thought that the moon is full and the perceptual state of seeing that the moon is full are both about the moon and they both represent the moon as being full. So both are intentional states. We say that the *intentional content* of both the thought and the perceptual state is *that the moon is full*. A single intentional content can have very different behavioral effects, depending on its relation to the person who has the content. For example, the fear that there will be nuclear war might inspire one to work for disarmament, but the belief that there will be nuclear war might influence one to emigrate to Australia. (Don't let the spelling mislead you: intending is only one kind of intentional state. Believing and desiring are others.) Intentionality is an important feature of many mental states, but many philosophers believe it is not "the mark of the mental." There are bodily sensations, the experience of orgasm, for example, that are genuine mental states but have no intentional content. (Well, maybe there is a bit of intentional content to this experience, e.g. locational content, but the phenomenal content of the experience, what it is like to have it, is clearly not exhausted by that that intentional content.)

The features of thought just mentioned are closely related to features of language. Thoughts represent, are about things, and can be true or false; and the same is true of *sentences.* The sentence `Bruce Springsteen was born in the USSR' is about Springsteen, represents him as having been born in the Soviet Union, and is false. It would be surprising if the intentional content of thought and of language were independent phenomena, and so it is natural to try to reduce one to the other or to find some common explanation for both. We will pursue this idea below, but before we go any further, let's try to get clearer about just what the difference is between intelligence and intentionality.

One way to get a handle on the distinction between intelligence and intentionality is to note that in the opinion of many writers on this topic, you

can have intentionality without intelligence. Thus John McCarthy (the creator of the artificial intelligence language LISP) holds that thermostats have intentional states in virtue of their capacity to represent and control temperature. And there is a school of thought that assigns content to tree rings in virtue of their representing the age of the tree. But no school of thought holds that the tree rings are actually intelligent. An intelligent system must have certain intelligent capacities, capacities to *do* certain sorts of things, and tree rings can't do these things. Less controversially, words on a page and images on a TV screen have intentionality. For example, my remark earlier in this paragraph to the effect that McCarthy created LISP is about McCarthy. But words on a page have no intelligence. Of course, the intentionality of words on a page is only derived intentionality, not original intentionality. Derived intentional content is inherited from the original intentional contents of intentional systems such as you and me. We have a great deal of freedom in giving symbols their derived intentional content. If we want to, we can decide that `McCarthy' will now represent Minsky or Chomsky. Original intentional contents are the intentional contents that the representations of an intentional system have *for* that system. Such intentional contents are not subject to our whim. Words on a page have derived intentionality, but they do not have any kind of intelligence, not even derived intelligence, whatever that would be.

Conversely, there can be intelligence without intentionality. Imagine that an event with negligible (but importantly, non-zero) probability occurs: In their random movement, particles from the swamp come together and by chance result in a molecule-for-molecule duplicate of your brain. The swamp brain is arguably intelligent, because it has many of the same capacities that your brain has. If we were to hook it up to the right inputs and outputs and give it an arithmetic problem, we would get an intelligent response. But there are reasons for denying that it has the intentional states that you have, and indeed, for denying that it has any intentional states at all. For since we have not hooked it up to input devices, it has never had any information from the world. Suppose your brain and it go through an identical process, a process that in your case is the thinking of the thought that Bernini vandalized the Pantheon. The identical process in the swamp-brain has the phenomenal features of that thought, in the sense of `phenomenal content' indicated in the discussion of orgasm above. What it is like for you to think the thought is just what it is like for the swamp-brain. But, unlike you, the swamp-brain has no idea who Bernini was, what the Pantheon is, or what vandalizing is. No

information about Bernini has made any kind of contact with the swamp-brain; no signals from the Pantheon have reached it either. Had it a mouth, it would merely be mouthing words. So no one should be happy with the idea that the swamp-brain is thinking the thought that Bernini vandalized the Pantheon.

The upshot: what makes a system intelligent is what it can do, what it has the capacity to do. So intelligence is future-oriented. What makes a system an intentional system, by contrast, is in part a matter of its causal history; it must have a history that makes its states represent the world, i.e., have aboutness. Intentionality has a past-oriented requirement. A system can satisfy the future-oriented needs of intelligence while flunking the past-oriented requirement of intentionality. (Philosophers disagree about just how future-oriented intentionality is, whether thinking about something requires the ability to "track" it; but there should be little disagreement that there is some past-oriented component.)

Now let's see what the difference between intelligence and intentionality has to do with the computer model of the mind. Notice that the method of functional analysis that explains intelligent processes by reducing them to unintelligent mechanical processes *does not explain intentionality*. The parts of an intentional system can be just as intentional as the whole system. In particular, the component processors of an intentional system can manipulate symbols that are about just the same things that the symbols manipulated by the whole system are about.

PART IIII

Finally, the third pertinent part, about Searle's Chinese room argument:

As we have seen, the idea that a certain type of symbol processing can be what *makes* something an intentional system is fundamental to the computer model of the mind. Let us now turn to a flamboyant frontal attack on this idea by John Searle. Searle's strategy is one of avoiding quibbles about specific programs by imagining that cognitive science of the distant future can come up with the program of an actual person who speaks and understands Chinese, and that this program can be implemented in a machine. Unlike many critics of the computer model, Searle is willing to grant that perhaps this can be done so as to focus on his claim that *even if this can be done, the machine will not have intentional states.*

The argument is based on a thought experiment. Imagine yourself given a job in which you work in a room (the Chinese room). You understand only English. Slips of paper with Chinese writing on them are put under the input door, and your job is to write sensible Chinese replies on other slips, and push them out under the output door. How do you do it? You act as the CPU (central processing unit) of a computer, following the computer program mentioned above that describes the symbol processing in an actual Chinese speaker's head. The program is printed in English in a library in the room. This is how you follow the program. Suppose the latest input has certain unintelligible (to you) Chinese squiggles on it. There is a blackboard on a wall of the room with a "state" number written on it; it says `*17*'. (The CPU of a computer is a device with a finite number of states whose activity is determined solely by its current state and input, and since you are acting as the CPU, your output will be determined by your intput and your "state". The `17' is on the blackboard to tell you what your "state" is.) You take book 17 out of the library, and look up these particular squiggles in it. Book 17 tells you to look at what is written on your scratch pad (the computer's internal memory), and given both the input squiggles and the scratch pad marks, you are directed to change what is on the scratch pad in a certain way, write certain other squiggles on your output pad, push the paper under the output door, and finally, change the number on the state board to `*193*'. As a result of this activity, speakers of Chinese find that the pieces of paper you slip under the output door are sensible replies to the inputs..

But you know nothing of what is being said in Chinese; you are just following instructions (in English) to look in certain books and write certain marks. According to Searle, since you don't understand any Chinese, the system of which you are the CPU is a mere Chinese simulator, not a real Chinese understander. Of course, Searle (rightly) rejects the Turing Test for understanding Chinese. His argument, then is that since the program of a real Chinese understander is not sufficient for understanding Chinese, no symbol-manipulation theory of Chinese understanding (or any other intentional state) is correct about what *makes* something a Chinese understander. Thus the conclusion of Searle's argument is that the fundamental idea of thought as symbol processing is wrong even if it allows us to build a machine that can duplicate the symbol processing of a person and thereby duplicate a person's behavior.

The best criticisms of the Chinese room argument have focused on what Searle—anticipating the challenge—calls the systems reply. The systems

reply has a positive and a negative component. The negative component is that we cannot reason from "Bill has never sold uranium to North Korea" to "Bill's company has never sold uranium to North Korea". Similarly, we cannot reason from "Bill does not understand Chinese" to "The system of which Bill is a part does not understand Chinese. There is a gap in Searle's argument. The positive component goes further, saying that the whole system--man + program + board + paper + input and output doors--does understand Chinese, even though the man who is acting as the CPU does not. If you open up your own computer, looking for the CPU, you will find that it is just one of the many chips and other components on the main circuit-board. The systems reply reminds us that the CPUs of the thinking computers we hope to have someday will not *themselves* think—rather, they will be *parts* of thinking systems.

Searle's clever reply is to imagine the paraphernalia of the "system" *internalized* as follows. First, instead of having you consult a library, we are to imagine you *memorizing* the whole library. Second, instead of writing notes on scratch pads, you are to memorize what you would have written on the pads, and you are to memorize what the state blackboard would say. Finally, instead of looking at notes put under one door and passing notes under another door, you just use your *own body* to listen to Chinese utterances and produce replies. (This version of the Chinese room has the additional advantage of generalizability so as to involve the complete behavior of a Chinese-speaking system instead of just a Chinese note exchanger.) But as Searle would emphasize, when you seem to Chinese speakers to be conducting a learned discourse with them in Chinese, all you are aware of doing is thinking about what noises the program tells you to make next, given the noises you hear and what you've written on your mental scratch pad.

I argued above that the CPU is just one of many components. If the whole system understands Chinese, that should not lead us to expect the CPU to understand Chinese. The effect of Searle's internalization move—the "new" Chinese Room—is to attempt to destroy the analogy between looking inside the computer and looking inside the Chinese Room. If one looks inside the computer, one sees many chips in addition to the CPU. But if one looks inside the "new" Chinese Room, all one sees is *you*, since you have memorized the library and internalized the functions of the scratchpad and the blackboard. But the point to keep in mind is that although the non-CPU components are no longer easy to see, they are not gone. Rather, they are

internalized. If the program requires the contents of one register to be placed in another register, and if you would have done this in the original Chinese Room by copying from one piece of scratch paper to another, in the new Chinese Room you must copy from one of your mental analogs of a piece of scratch paper to another. You are implementing the system by doing what the CPU would do and you are simultaneously simulating the non-CPU components. So if the positive side of the systems reply is correct, the total system that you are implementing does understand Chinese.

"But how can it be", Searle would object, "that you implement a system that understands Chinese even though *you* don't understand Chinese?" The systems reply rejoinder is that you implement a Chinese understanding system without yourself understanding Chinese or necessarily even being aware of what you are doing under that description. The systems reply sees the Chinese Room (new and old) as an English system implementing a Chinese system. What you are aware of are the thoughts of the English system, for example your following instructions and consulting your internal library. But in virtue of doing this Herculean task, you are also implementing a real intelligent Chinese-speaking system, and so your body houses two genuinely distinct intelligent systems. The Chinese system also thinks, but though you implement this thought, you are not aware of it.

The systems reply can be backed up with an addition to the thought experiment that highlights the division of labor. Imagine that you take on the Chinese simulating as a 9-5 job. You come in Monday morning after a weekend of relaxation, and you are paid to follow the program until 5 PM. When you are working, you concentrate hard at working, and so instead of trying to figure out the meaning of what is said to you, you focus your energies on working out what the program tells you to do in response to each input. As a result, during working hours, you respond to everything just as the program dictates, except for occasional glances at your watch. (The glances at your watch fall under the same category as the noises and heat given off by computers: aspects of their behavior that is not part of the machine description but are due rather to features of the implementation.) If someone speaks to you in English, you say what the program (which, you recall, describes a real Chinese speaker) dictates. So if during working hours someone speaks to you in English, you respond with a request in Chinese to speak Chinese, or even an inexpertly pronounced "No speak English," that was once memorized by the Chinese speaker being simulated, and which you the English speaking system may even fail to recognize as English. Then,

come 5 PM, you stop working, and react to Chinese talk the way any monolingual English speaker would.

Why is it that the English system implements the Chinese system rather than, say, the other way around? Because you (the English system whom I am now addressing) are following the instructions of a program in English to make Chinese noises and not the other way around. If you decide to quit your job to become a magician, the Chinese system disappears. However, if the Chinese system decides to become a magician, he will make plans that he would express in Chinese, but then when 5 P.M. rolls around, you quit for the day, and the Chinese system's plans are on the shelf until you come back to work. And of course you have no commitment to doing *whatever* the program dictates. If the program dictates that you make a series of movements that leads you to a flight to China, you can drop out of the simulating mode, saying "I quit!" The Chinese speaker's existence and the fulfillment of his plans depends on your work schedule and your plans, not the other way around.

Thus, you and the Chinese system cohabit one body. In effect, Searle uses the fact that you are not aware of the Chinese system's thoughts as an argument that it has no thoughts. But this is an invalid argument. Real cases of multiple personalities are often cases in which one personality is unaware of the others.

It is instructive to compare Searle's thought experiment with the string-searching Aunt Bubbles machine described at the outset of this paper. This machine was used against a behaviorist proposal of a behavioral *concept* of intelligence. But the symbol manipulation view of the mind is not a proposal about our everyday concept. To the extent that we think of the English system as implementing a Chinese system, that will be because we find the symbol-manipulation theory of the mind plausible as an empirical theory.

There is one aspect of Searle's case with which I am sympathetic. I have my doubts as to whether there is anything it is like to be the Chinese system, that is, whether the Chinese system is a phenomenally conscious system. My doubts arise from the idea that perhaps consciousness is more a matter of implementation of symbol processing than of symbol processing itself. Though surprisingly Searle does not mention this idea in connection with the Chinese Room, it can be seen as the argumentative heart of his position. Searle has argued independently of the Chinese Room that intentionality requires consciousness. But this doctrine, if correct, can shore up the Chinese

Room argument. For if the Chinese system is not conscious, then, according to Searle's doctrine, it is not an intentional system either.

Even if I am right about the failure of Searle's argument, it does succeed in sharpening our understanding of the nature of intentionality and its relation to computation and representation.