When confronted with the AI vision of superintelligent computers of the future, computers that could serve as psychotherapists, judges, or physicians, many react with a force of feeling that surprises them. Some try to neutralize their discomfort by asserting the impossibility of artificial intelligence. Others insist that even if possible, it should not be allowed. The vehemence of response expresses our stake in maintaining the line between the natural and the artificial, between the human and the mechanical. Discussion about computers becomes charged with feelings about what is special about people: their creativity, their sensuality, their pain and pleasure. But paradoxically, when faced with a machine that shows any degree of "intelligence," many of these same people seem pulled toward treating the machine as though it were a person.

Jean is an airline ticket agent who uses a computer to make customer reservations. She first represented the computer as a neutral object—programmed, passive, completely under the control of its operators, threatening only in its impersonality. But then she moved on to more ambivalent descriptions.

Jean often confronts angry clients. The planes are overbooked; the computer has been programmed on the assumption that some people won't show, but occasionally everybody shows, and not everyone can go. Jean's excuse is, "The computer fouled up." I talk to her about what she means. Is this a "manner of speaking"? It is not. She seems to experience the computer as an autonomous entity that can act on its own behalf and is thus "blamable." This
anthropomorphization has consequences. It means that when things go wrong Jean need not call the policies of her company into question or lay blame on fellow workers. The anthropomorphized machine gives her an out. She can sympathize with her inconvenienced clients without jeopardizing her relationship with co-workers or her security as a “company person.”

The simplest force that makes the computer seem more than a machine among other machines is its behavior. Jean asked her computer questions and it gave her answers. She tells me that “of course people give the computer the flight schedules, they program them in so it will know the rules, but then it uses its own mind.” Like the ELIZA program whose psychotherapeutic dialogue drew people into its “confidence,” Jean’s computer behaves in a way that encourages her to treat it as a person.

Jean’s computer did not draw her into an explicitly metaphysical discourse about minds and machines. But like the child who argues about Merlin’s cheating, she has taken a step toward allowing a continuity between the psychologies of machines and people.

Jean’s anthropomorphization is naïve. She treats the computer as a black box, and she doesn’t look inside. But computers encourage their anthropomorphization because of more than their behavior. In an important sense the computer is irreducible. It is hard to block the temptation to personify the computer by saying what it “really” is, hard to block the suggestion that the computer “thinks” by saying what it “really” does. It is hard to capture the computer by seeing it in terms of familiar objects or processes that existed before it was invented. The computer is not “like” anything else in any simple sense.

**Anthropomorphization and Irreducibility**

Airplanes come in all shapes and can be described in all sorts of ways, but there is no conceptual problem in stating what they do: they fly. There is no equally elegant, compelling, or satisfying way of defining the computer by its function. You can say “it computes,” and the computer scientist can set up a conceptual frame of reference in order to define “the computable.” But even then what has been isolated as “the essential computer” presents no easy analogies with other objects in the world (as the airplane does the bird), except, of course, for its analogies with people.

Some might say that no matter how complex the computational product—a medical diagnosis, a move in a chess game—“all the computer really does is add.” In a certain sense this is true. But saying that a computer “decided to move the queen by adding” is a little bit like saying that Picasso “created Guernica by making brushstrokes.” Reducing things to this level of “local” description gives no satisfying way to grasp the whole. When we talk about perception—for example, how we grasp visual images—there is a tension between the atomic and the gestalt. Do we build up the picture from discrete bits of information or do we grasp it as a whole? Similarly, in thinking about computers, there is a tension between the local simplicity of the individual “acts” that comprise a program and the global complexity that can emerge when it is run. When you talk about the overall behavior of the program, descriptions of local simplicities seem off the point, and the temptation is to see the machine as something that borders on life.

Computers are certainly not the only machines that evoke anthropomorphization. We often talk about machines, and even to machines, as though they were people. We complain that a car “wants to veer left.” We park it on a slope and warn it to “stay put.” But usually, when we “talk to technology,” we know that any voluntary action we may have ascribed to a machine is really a series of unambiguously “mechanical” events. We know that the pressure of the emergency brake will prevent gravity from pulling a car down a hill. But when we play chess with a computer and say that the computer “decided” to move the queen, it is much harder to translate this decision into physical terms.

In the early nineteenth century, the exhibition of what was claimed to be a chess-playing automaton created a sensation. The machine was a fake. Inside the chess-playing machine there was a chess-playing man. No one knew how to make a machine with the complexity of behavior that goes into playing even poor chess. No one knew what such a machine might be made of—wheels, gears, levers, strings, pipes with flowing liquids? And even if one knew what “parts” to use, what principles could guide their organization?

It is now possible to walk into a toy store and buy a chess-playing machine that probably plays a better game than the nineteenth-century man in the machine. We have computers programmed to
play chess, but not because anyone found the right “parts” to use. Indeed, one might say that the breakthrough was the realization that any particular physical “part” is irrelevant. The physical parts of a computer mark states in a process. Most computers are made of electrical markers, but they don’t have to be. A computer has been constructed out of Tinkertoys parts. It was made at MIT to dramatize that what is important about a computer is the machine’s ability to embody a process, to specify a sequence of rules. In a very real sense, the chess machine and the Tinkertoys machine, like all computers, are not made of wood, or water, or silicon. They are made of logic. And thinking about the core of a machine as the exercise of logic leads people back to thinking of the computer as mind.

But despite these encouragements to personify computers, people have a stake in seeing themselves as different. They assert this difference, much as we saw children do. They speak of human love, sensuality, and emotion. They speak of the computer’s lack of consciousness, originality, and intention. In the end, many sum up their sense of difference in the statement “Computers are programmed; people aren’t,” or “Computers only do what they are programmed to do, nothing more, nothing less.” This last response has a history. It is associated with Lady Ada Lovelace, a friend and patroness of Charles Babbage, inventor of the “analytical engine,” the first machine that deserves to be called a computer in the modern sense. In a memoir she wrote in 1842 she became the first person known to go on record with a variant of “Computers only do what you tell them to do.”

From the perspective of this model of a computer program—let us call it “the Lovelace model”—people listen with incredulity to the AI scientist who says that the human mind can be captured by a program. They can’t understand how anyone could think that his or her mind works that way. But AI ideas are moving out and capturing the popular imagination because the AI community has generated a set of ideas that undermine the Lovelace model of what it means to be programmed. This chapter is about some of these ideas and how they are picked up by those who come into contact with them. People become sympathetic to the idea that machine intelligence has something to teach us about human intelligence, and even to the idea that the computer’s psychology may be close to our own, when they are drawn into thinking that somehow Lovelace must be wrong—that in some sense computers must be able to do more than “what you tell them to do.”

Beyond Lovelace

Jordan is a sculptor. Until recently, he never had anything to do with computers, except for the one that dispenses cash at the bank. Jordan’s relationship with computers changed when small chess-playing machines were introduced. Together with art, chess is his passion. He bought a chess machine and found it was good enough to give him an exciting game. Computers became more than un differentiated “smart machines.” Jordan wanted to know how this particular machine worked.

Our culture is rich in ways of thinking that answer Jordan’s question in Lovelace style: for example, the chess machine plays by rules that have been programmed into it. To this Jordan’s disappointed reaction was clear: computer chess has nothing to do with human chess. He was sure that he didn’t play by consulting rules and he rejected the machine’s intelligence despite its intelligent behavior: “It’s not thinking, it’s just rules.”

Several months later Jordan changed his mind when he heard that his chess machine doesn’t play by rules but by “heuristics,” something that is ultimately a rule, but which feels like something else, more like what we call a “rule of thumb.”

Since a chess program cannot check through and evaluate all the possibilities for each move, it relies on general guidelines, “heuristics,” to pick out a few it will examine in depth. The idea that a program can have a degree of flexibility and imprecision gave Jordan the sense that it possesses something akin to what he calls “intuition.” And with this, he could once again identify with the computer.

Jordan was able to see the chess computer as like him, as a “thinking thing,” once he moved from seeing its program as rigidly serial—one fixed action after another—to thinking of it as multiple processes in interaction. More elaborated versions of this kind of representation allow AI scientists to see in the notion of “program” not what is most inhuman about machines but what computers and
people must have in common. These "anti-Lovelace" representations figure in the writings of AI theoreticians. They have spread to their students and beyond. Recently, they have reached Hollywood. They are the representations that serve to break down people's resistance to seeing a continuity between computers and people.

**Tron and a Society of Mind**

I attend the Boston premiere of the Walt Disney movie *Tron*, advertised as "taking place inside of a computer." The film creates an "interior" computational landscape, and the hero, a hacker named Flynn, spends three-quarters of the film trapped there, prisoner within a system he has built. *Tron* shows the insides of a computer as a community of programs, each personified as an "actor" with a history, a personality, and a function within a complex political organization. We meet the inhabitants of this world at the time of a political crisis. A program called the Master Control Program, unaffectionately referred to throughout as the MCP, has assumed dictatorial powers. Brutal police programs are used to bring all other programs under control. There are skirmishes, battles, and finally, with Flynn's help, all-out warfare within the society.

When the film is over and the lights go on I see Marvin Minsky. Minsky has been charmed. "That was great," he says, "That's a whole lot better than bits! I am in the middle of writing a paper which proposes to outlaw the whole idea of bits. It's no way to think about what goes on inside of a computer."

When Minsky talks about "outlawing the whole idea of bits" he means changing the first image most people get of computers. It begins with the idea that the computer is made of electronic switches that are either on or off (these are the bits) and builds up to the Lovelace step-by-step model of a program. If you are trying to use the computer as a model of mind, the bit and Lovelace models are unsatisfying, just as thinking of "rules" was unsatisfying for Jordan trying to identify with his chess machine. I ask Minsky what he wants to put in place of the bits. He answers, with a look that makes it clear that the answer should be evident, "A society, of course, just like in *Tron*." "Society" is his mnemonic for multiple, simultaneously interacting programs within a complex computer system. In the *Tron* landscape Minsky has found an image, however fanciful, for what he has in mind.

The image of the computer as a "bit machine" and the image of the computer as a society of programs are not irreconcilable. Any computer has the bits that Minsky wants to outlaw, but the question here is at what level we choose to understand things. It has often been suggested that the relationship of bits to program is analogous to that of brain to mind. In the "brain" there are physical things—neurons, synapses, electrical impulses. In the "mind" there are images, concepts, ideas, language, and thought. Similarly, within the computer there are physical things (the bits), and then, at the level of "mind," there are programs.

In some sense, the activity of the neurons must be the physical manifestation of the same events we experience as thoughts and images, but usually the detail of how this happens is irrelevant for using one's mind: for feeling, thinking, communicating. On the other hand, in the case of neurological disorder, priorities change. Then it becomes important to find someone who knows how to probe, stimulate, and record the activity of the physical brain.

In a computer, different levels of understanding are appropriate at different times. The repairman focuses his attention on bits, but if you want to think of mind as computer, you are not likely to be drawn to the level of on/off switches or Lovelace models. You are more likely to be swept into the world of *Tron*.

*Tron* does more than assert the primacy of software over hardware, of program over electrical circuit. Its presentation of a computer system as an unruly society is in dramatic contrast with the linear model of programming. Even if the members of the *Tron* society had been produced by Lovelacian programming, their interaction leads to the playing out of an unpredictable drama.²

The idea of a computer system as a "society" of competing programs is one of several key ideas from the AI community that challenge the image of the computer as following step-by-step instructions in a literal-minded way and make it easier for people to think of mind as machine. Another one of these is that the computer is capable of "learning." In that case, you don't have to tell the computer everything it needs to know; you have to tell it how to learn. And yet another, which combines these two, is an idea that I will refer to as "emergence": you don't have to tell the computer everything it needs to know; you have to arrange for it to
The Idea of Emergence

One of the most famous and influential "learning" programs was written by Arthur Samuel in the late 1950s. This program played checkers according to built-in rules. In that sense, it did what it was programmed to do. But it was also programmed to modify its rules based on its "experience." It played many games, against many opponents, and it did get better, finally achieving the status of a "world class" checkers player. But the dramatic moment in this program's life was not the day it beat a champion, but the day it beat its creator. The program became good enough to beat Arthur Samuel at his own game.

Among those struck by the drama of that moment was Norbert Wiener. For the mathematician usually regarded as the founder of cybernetics, the machine triumphing over its creator symbolized a new era. In God and Golem, Inc. he suggested that the implications border on the theological: "Can God play a significant game with his own creature? Can any creator, even a limited one, play a significant game with his own creature?"

In the checkers program, Samuel took a decisive technical step beyond the Lovelace model of programming. To show how, consider the ways one might determine whether one arrangement of pieces on a checkerboard is more "favorable" than another. One "rule of thumb" is to count pieces. In fact, children often decide whether they are "winning" on this basis alone. Another method would be to look at how many of the enemy pieces are "under attack." A program that acted simply by following either of these two rules would play a very weak game. To make a stronger program you could go in either of two directions. The "Lovelace"
direction would consist of adding systematic precise rules for measuring "being ahead," rules that would be increasingly subtle and complex. The other way, "heuristic programming," uses a collection of disparate rules of thumb. From this perspective, you take the two "weak" rules (counting pieces and counting threats) together, recognizing that each has a place in the game even though insufficient by itself. Then you might add other rules to make a collection of "agents" within the program, each with a particular point of view, each limited, but valid in its own way.

This was Samuel's strategy. Once he had chosen it, the problem for him, as it would be for the manager of any organization whose individual members are confined to their own specialties, was how to create a structure within which the agents could interact. Samuel's solution was to have them cast weighted votes. What was most innovative about his program was that the program itself modified the weights depending on the track record of the different agents and of the program as a whole. Samuel's program worked on a kind of "society" model. The choice of which move to make was based on a "democratic" process in which a number of agents, each with a partial, limited view of the situation, could vote for or against a particular move, and the majority would win. For this program, "learning" consisted of adjusting the number of votes that could be cast by any particular agent.

In *God and Golem*, Norbert Wiener discussed whether the Samuel program "merely did what Samuel programmed it to do." It did, but in this case the programming had, in a certain sense, instructed it to take off on its own to make its own decisions. Lady Lovelace's dictum begins to have a different feel when obedience means autonomy. For Wiener the working of the Samuel program started to feel like overstepping an ancient taboo: the taboo of speaking of "living beings and machines in the same breath." 8

The idea of the Samuel-style voting procedures (in more technical language, called "perceptrons") generated excitement as a model for getting machines to make intelligent decisions of all sorts, not just in checkers. But it soon became clear that it was exceptional for a decision procedure to be reducible to a perceptron. Not all learning could be made as simple. What worked for checkers would work for little else. 9

In the classical perceptron that Samuel used, intelligence emerged by simple addition of "votes." What didn't work out mathematically for a large number of cases was the idea of "simple addition." But what remained a strong metaphor within the AI world was the idea of a society of limited agents whose intelligence is emergent from their interaction, the idea that a computer system as a whole will be significantly, qualitatively different than the sum of its parts. We use this idea when we think about living things and about creation: for example, the DNA that emerged from the inert molecules of the nonliving earth. AI has seized it to think of how intelligence might "grow" in a computer program in a way that might model the way it grows in a child.

**"Growing" Intelligence**

If we take a simple problem like trying to decide whether there are more black or white marbles on a tabletop, something that very young children can do, it is easy to write a Lovelace-style program that could solve it. Have one part of the program, one subprocedure, count the black marbles, have another subprocedure count the white marbles. Have a third subprocedure compare the numbers and report the color with the greatest number back to the top level of the program.

This Lovelace program does the job. But it presents a problem if you are trying to model how the child does it. In order to decide whether there are more black or more white marbles on the tabletop, the Lovelace system needs to know how to count, how to compare numbers, and, most complicated of all, how to integrate this counting and comparing ability into the solution of its problem. But children can say "which is more" long before they are able to do these things. If you are trying to model a growing mind, it would be more satisfying to imagine how "which is more" could be built up out of the kinds of things that small children can do. In other words, it would be more satisfying to model this piece of intelligence as "emergent."

You can imagine a small child telling the difference between a black marble and a white one, grabbing one of each, and putting them away. In order to tell which is more, all the child needs to do is repeat this action until there are either no marbles left or marbles of only one color. In the latter case, the child can then call out the color of the remaining marbles, and that color will be "more." This
description reflects, better than the Lovelace "counting program," the kinds of matching ability young children actually have. Thus, a computer program that follows it will have greater psychological plausibility. Some AI theorists have suggested ways of writing this kind of program. For example, Alan Newell and Herbert Simon, who want to construct theories of human thought in the form of programs, have put forth the idea of a "production system," a style of programming that moves away from Lovelace to get closer in spirit to the child solving the "which is more" problem by grabbing the marbles off the table.

A production system is a little society inhabited by agents each of which can recognize a specific condition and, when it arises, carry out a specific job. A first agent might be able to recognize black and white marbles (its "triggering condition") and put them aside, "grab them off the table." Another agent does nothing unless it sees a tabletop with only white marbles, in which case it says "more white." The third does nothing unless it sees a tabletop with only black marbles, in which case it says "more black." In traditional Lovelace programming, the programmer fixes in advance the order in which instructions will be executed. In a production system, the order in which the agents do their jobs is not determined by the program but by the environment in which the system finds itself.

The Lovelace program had a subprocedure that knew how to count, a relatively advanced ability. None of the three agents here (Newell and Simon call such agents "productions") has anything approaching this degree of smartness. All they can do is follow simple rules about grabbing pairs of marbles or making pronouncements about "more black" or "more white." In the Lovelace program, numerical ability is programmed in. In the production system, it emerges from the interaction of agents who don't have any. In the language of the students I interviewed, intelligence comes from having a number of "dumb agents" that can produce smart behavior when they work together.

Newell and Simon's production system and Samuel's checkers program can both be thought of as societies of agents that are restricted in a particular way. In Samuel's case, agents act only by voting. In the case of the productions, each agent has to wait its turn to act. Marvin Minsky and Seymour Papert have developed a society theory of emergent intelligence in which they lift these restrictions. They see agents with a wide diversity of roles, including the administration and the censoring of other agents.10

Papert uses this type of model to explain how children develop conservation of liquids, the knowledge that the quantity of a liquid doesn't change when you pour it from a short, stout glass into a tall, thin one. Young children consistently judge how much water there is by the water level, and so the higher level in the tall glass makes that water seem like "more."

Papert begins by describing three agents, each of which judges quantities in a different simplified way. The first judges quantity by height; anything that goes higher is "more." A second agent judges quantity by width, horizontal extent. And a third agent says that quantities are the same if they once were the same. This third agent, which Papert calls a "history" agent, "seems to sound like a conservationist child, but this is an illusion."11 The history agent knows only how to say that things are the same as they used to be, even things that have in fact changed. When the water is poured from the stout to the thin glass in front of a preconservationist child, "each of the three agents makes its own 'decision' and clamors for it to be adopted." For the preconservationist child, the height agent's voice speaks the loudest, but this changes as the child moves on to the next stage.

Papert believes that the child moves on to the next stage when the height and width agents get into a new relationship: they start to neutralize each other when they give contradictory opinions. This happens because a new agent called a "geometry" agent acts as a referee. When the height and width agents both say either more or less, the geometry agent passes on their common message to the "top level" of the information system that is this child. But if they disagree, the authority of the geometry agent is undermined, in which case the history agent's voice is loudest and the child "has conservation." The geometry agent doesn't know what the programs under it are called or what they do or why. It knows only that before it can make a move it has to hear from them, pass on their message if both are the same, go back to sleep if they are not.

Papert's account makes the appearance of conservation in the child's mind "emergent" from the coming into being of the geometry agent. But this agent was not introduced "in order" to produce conservation. Papert believes that in its most primitive form it was there long before as an agent capable of recognizing disagreement.
Thinking of Yourself as a Machine

Mark is an MIT junior and a computer-science major. He has always been interested in logic and systems. When asked to think back to his childhood games, he does not talk about playing with other children. He recalls how he "sorted two thousand Lego pieces by color, size, and shape." He liked the feel of sorting the pieces, and he liked making the Legos into complex structures, "things that you would never expect to be able to make from Legos." Although Mark spends most of his time working with computers, he sees himself as a professional in training rather than as a computer hacker. He cares about his schoolwork, gets top grades, and "tries to keep things other than computers in my life." One of these is being "Dungeon Master" for one of MIT's many ongoing games of Dungeons and Dragons. His game has been going for a couple of years, and he is pretty sure it is the best game in the Boston area.

The people who play my dungeon use it to express their personalities, and also to play out sides of their personalities that they hide in their everyday life. That's one of the things that is so fantastic about the game. But you have to make a great game for that to really happen. You have to hold them in your world, but you have to give them a lot of space to be themselves. It is a very complicated thing, an art.

Mark spends many hours a week preparing for the Sunday-afternoon meeting of his dungeon: "at least five hours of preparation for each hour of play. And sometimes we play for five hours. It's a responsibility. I take it very seriously. It's one of the most creative things I do."

Mark has strong relationships with the members of his dungeon. He is more able to involve himself with them than with people he meets outside of the structure of the game. The dungeon offers him a safe world built on a complex set of rules. Within it he can use his artistry to build social situations much as he built his Lego constructions. Dungeons and Dragons allows Mark to be with people in a way that is as comfortable for him as dealing with things. It is his solution to a familiar dilemma: he needs and yet fears personal intimacy. He has found a way of being a loner without
being alone. One might say that for him Dungeons and Dragons becomes a social world structured like a machine.

There is another and more dramatic way in which Mark shows his preference for machinelike systems. He has elaborated a theory of psychology that leads him to see himself and everyone else as a machine. The theory, which he claims to be his own, is made up largely of ideas current at the Artificial Intelligence Laboratory at MIT. We see in Mark how these ideas are appropriated by someone who is not part of the AI world, but who is close enough to be a first link on the chain of these ideas moving out.13

Over the past year, he has built up a detailed picture of exactly how the Mark-machine works through a self-consciously introspective method. “Like Freud,” he tells me, “I don’t follow other people’s theories. I just think about myself and make up my own—the way Freud came up with his theories and then looked around him and fit things in.” When Freud “looked around” he “fit in” ideas from his scientific culture, from physics and biology. Mark’s is a computer culture. He uses computer systems to think about all complex systems, especially to consider the complexity of his own mind.

Mark begins with the idea that the brain is a computer. “This does not mean that the structure of the brain resembles the architecture of any present-day computer system, but the brain can be modeled using components emulated by modern digital parts. At no time does any part of the brain function in a way that cannot be emulated in digital or analog logic.” In Tron, the programs are complicated, psychological, and “motivated” beings. They do a lot of running around, planning, plotting, and fighting. In Mark’s model the computational actors in the brain are simple. Each is a little computer with an even smaller program, and each “knows only one thought.” Mark takes the “one thought” limitation seriously. “One processor might retain the visual impression of a computer cabinet. Another might retain the auditory memory of a keyboard being typed at. A third processor might maintain the visual image of a penguin.”

In Mark’s model, all of the processors have the same status: they are “observers” at a long trough. Everything that appears in the trough can be seen simultaneously by all the observers at every point along it. The trough with its observers is a multiproccessing computer system. Using computer jargon, Mark describes the trough as a “bus,” a trunk line that puts actors in contact with each other. Their communication options are very limited. Each looks at the trough and when something appears that relates to what “he” knows, “all he can do is put his knowledge in.”

The trough plus observers make up the central processor of the brain. The consciousness of the brain is only a reflection of what is in the trough at a given time. Consciousness is a passive observer looking at the trough. It does not even see everything in the trough, but only those things which are very strong, either because they were dumped in the trough by more than one observer or they were in the trough for a very long time.

Mark goes on to elaborate this notion of consciousness as a passive observer:

The processors, the observers, correspond to neurons in the brain. If a researcher on the edge of the brain could sample a number of the neurons and decode what impression was active in the brain at a given time, that researcher could be “seeing” what the brain was thinking. That researcher would be performing the function of the consciousness. Helpless to alter the chain reactions in the brain, this consciousness is rather carried along in the thought process, sensing whatever is the strongest impression at a given time. The consciousness is a helpless observer in the process we call thought. It is a byproduct of the local events between neurons.

Mark does not read philosophy. But his computer model of mind forces him to grapple with some of the oldest philosophical questions: the idea of free will and the question of a “self.” Mark’s theory has room for neither: the trough and the observers are a deterministic system. What we experience as consciousness is only a “helpless bystander” who gets the strongest signals filtered up to him. “Actions as well as thought,” says Mark, “are determined by the cacophony of the processor voices.”

There is no free will in Mark’s system. The individual’s feeling of conscious decision-making is an illusion, or rather an imposture: one of the processors, just as dumb as the others, has arrogated to itself the “seeming” role of consciousness. It has no power of decision, “it could just be a printer, attached to the computer. The strongest messages from the agents would just be printed out.” Consciousness is epiphenomenon. Mark says that even if there
were agents that could act with “free will,” it would still be “them” and not “him” who would have it. In Mark’s way of looking at things, there is no “me.”

You think you’re making a decision, but are you really? For instance, when you have a creative idea, what happens? All of a sudden, you think of something. Right? Wrong. You didn’t think of it. It just filtered through—the consciousness processor just sits there and watches this cacophony of other processors yelling onto the bus and skims off the top what he thinks is the most important thing, one thing at a time. A creative idea just means that one of the processors made a link between two unassociated things because he thought they were related.

In the course of my interview with Mark, creativity, individual responsibility, free will, and emotion were all being dissolved, simply grist for the little processors’ mills. I asked Mark if he thought that “mind” is anything more than the feeling of having one. His answer was clear: “You have to stop talking about your mind as though it were thinking. It’s not. It is just doing.”

Mark takes the idea of agents and runs with it as far as it can take him—to the demolition of the idea of free will, to the demolition of the idea that he has a “responsible” self. And when I ask him about emotions, he says, “OK, let’s model it on a piece of paper.”*

Even though Mark makes it clear that the world must wait for tomorrow’s multiprocessing technology to achieve the working intelligence that he has modeled, he turns his version of the “society theory” into something that uses today’s computer parts. His bus and simple processors are doing things that Mark thinks he knows how to make computers do (like recognize an “image of a penguin”). And when something comes up that he doesn’t know how to make a computer do, he can postpone the problem by claiming that intelligence will emerge through the interaction of the processors.

Mark finesses problems that would require making his model more complicated or more specific. He talks about intelligence emerging from “two or three or four hundred stupid agents.” But considering the highly specific skill that he gives to each of them (“one knows how to recognize a picture of a computer, another knows how to recognize a picture of a computer cabinet”), three or four hundred seem hopelessly few, even to allow the baby to recognize the objects seen from the crib, the playpen, and the high chair. Mark talks about his multiprocessor “model,” but what he really has is a multiprocessor metaphor. Despite its generality and vagueness he finds his metaphor powerful. First, he can identify with it directly. He can put himself in the place of the agents. They look, they shout, they struggle, they assert their piece of smartness in the crowd. And it is powerful because Mark’s daily experience with computers makes his theory seem real to him. Without the computer Mark’s theory would feel to him to be nothing more than “wishy-washy hand-waving,” the slur that engineers use to deride the psychological models of the precomputational past. He believes that in his model the problem of intelligence has been reduced to a technical one. Since he sees himself as a technical person, this makes him happy. It gives him a sense of being very powerful indeed. It means the appropriation of psychology by the engineers. What sweet revenge if the “ugly ones” turned out to be the gurus of the mind.

Mark’s way of talking is exceptional only because his ideas are elaborated and he has such utter confidence in them. But the idea of thinking of the self as a set of computer programs is widespread among students I interviewed at Harvard and MIT who were familiar with large computer systems. Like Mark, they find that the complexity of these systems offers a way to think about their minds.

Eliot, an MIT biology major, calls his agents the “gallery of stupid” who “input into clusters of special-interest groups inside his brain.” And then these special-interest groups fight it out and forward the result of their debates to his conscious self.

Should I study, should I go to sleep? Most of the time, it comes to a debate. I am aware of the conflict, but the debate gets played...
out by the agents. And it continues while I sleep. Sometimes I decide to go to sleep, and I awake with no other thought but to study. The agents have closed their debate. And the signals that are most powerfully coming up to consciousness are the study signals.

Ned, an MIT premedical student, also thinks of his mind as a multiprocessor:

Some of the agents are a little smarter than the others. The way an op-amp is smarter than a transistor, but that still makes them a long way off from having consciousness or free will. Consciousness and free will are illusions created by having many of the smarter processors and many of the dumber processors linked together by billions and billions of neural connections.

Children debating the metaphysical status of computer toys led me to say that computers bring philosophy into everyday life. What made the children's discussions philosophical was not their conclusions, but the structure of their argument, the effort to resolve intellectual tensions. But Elliot and Ned use the computer presence to bypass the mind/body problem: for them the reduction of mind to matter is unproblematic, as is the assertion that free will is an illusion. There is no tension, no need to examine their beliefs. We could say that Elliot and Ned are discussing free will and the mind/body problem, but their way of doing so suggests a qualification: sometimes computers bring philosophy into everyday life only to take it away again.

But if, for them, the mind/body problem does not exist, the reliance of their theories on the idea of multiple agents leaves them with a different problem: does the self exist. Has it been dissolved in the fragmentation of quarreling processors?

Challenging the “I”

A model of mind as multiprocessor leaves you with a “decentralized” self: there is no “me,” no “I,” no unitary actor. Mark expressed this when he admonished me not to talk about my mind as though “I” was thinking. “All that there is is a lot of processors—not thinking, but each doing its little thing.” Elliot put the same thought jokingly: “Nobody is home—just a lot of little bodies.”

But theories that deny and “decenter” the “I” challenge most people's day-to-day experience of having one. The assumption that there is an “I” is solidly built into ordinary language, so much so that it is almost impossible to express “anti-ego” theory in language. From the moment that we begin to write or speak, we are trapped in formulations such as “I want,” “I do,” “I think.” Even as we articulate a “decentered” theory there is a pull back away from it, sometimes consciously, sometimes not.

Of course, “mind as multiprocessor” is not the first challenge to the “I” that has encountered the resistance of everyday “commonsense” psychology and everyday speech. The idea of the Freudian unconscious is also incompatible with our belief that we “know what we want.” We don’t, says Freud. “Our” wishes are hidden from “us” by a complex process of censorship and repression. We are driven by forces quite outside our knowledge or control.

Freud divided the “I,” but the theorists who followed him moved toward restoring it. They did so by focusing on the ego turned outward toward reality. They began to see it as capable of integrating the psyche. To them, the ego seemed almost a psychic hero as it battled id and superego at the same time that it tried to cope with the demands of the everyday. Anna Freud wrote of its powerful artillery, the mechanisms of defense, which helped it in its struggles, and Heinz Hartmann argued that the ego had an aspect that was not tied up in the individual's neurotic conflicts: it had a “conflict-free zone.” This “unhampered” aspect of the ego was free to act and choose, independent of constraints. It almost seemed the seat for a reborn notion of the will, the locus of moral responsibility. Intellectual historian Russell Jacoby, writing of psychoanalytic ego psychology’s reborn, autonomous “I,” described it as the “forgetting of psychoanalysis.”

A theory that had called the “self” into question now had a reassuring notion of the ego as a stable “objective” platform from which to view the world. A decentered theory had been recentered. A subversive theory had been normalized. Ego psychology is the version of the unconscious most acceptable to the conscious.

As a theory aspires to move from the world of high science to that of the popular culture, there is a natural pressure to cast it into more acceptable forms. Theories that call the self into question...
live in a natural state of tension. Although much of their power comes from the fact that they offer concrete images through which to express our sense of being constrained or driven by forces beyond our control, they are also under constant pressure from that other side of our experience—our sense of ourselves as selves. This was true of psychoanalysis. It is true of multiprocessor models of mind.

The Reconstituted Center

Mark begins with the flat assertion that there is no self, no conscious actor. "Consciousness is just a feeling of thinking." But even as he makes his case, the contradictions slip in. He claims that consciousness "just sits there and watches this cacophony of other processors yelling onto the bus and skims off the top what he thinks is the most important thing, one thing at a time." But who is "he"? Could this skimming really be done by a processor that is "passive as a printer"?

Mark's description of how he developed his theory, his description of his introspective method, illustrates the contradiction in his position.

My only way to know all this, my only way to tell, is by closing my eyes and trying to figure out what I'm doing at any given time. My theory was not developed as a "neat hack" to explain something, but rather "I thought this was going on" based on the small amount of evidence that filtered through to my consciousness of what's going on in my mind.

Here there is the reappearance of a "subject," something either smart enough to make up the theory itself or able to send ideas that come up "back down the line" for further processing. But according to Mark's theory as he states it rigorously, neither is possible: the consciousness processor is dumb ("just a printer") and, at most, has the power to "send back a bit or two." Mark brings in something like a self despite himself. He would like to accept the pure decentered theory, but this is not easy for anyone. We are pulled back to common-sense ways of thinking that are familiar since childhood and supported by language, ways of thinking that make us feel that there is a self, the "I" is in control.

Throughout history, philosophers have challenged what seems obvious to common sense. Some have argued that we don't really know that objects exist or that other people have minds, or that there really are relations of cause and effect. The eighteenth-century Scottish philosopher David Hume held to this last view: events might follow one another, but we never have the right to say that one caused the other. While this might be easy enough in the philosophy seminar, in ordinary life we are obliged to act and talk as though we had the right to make such assumptions. And thus we understand Hume's *cri de coeur*: "Skepticism can be thought but not lived."

The decentered theory of mind, whether in psychoanalytic or computational terms, is as hard to live with as Hume's skepticism, and for similar reasons. The cornerstone of ordinary life is that there is an "I" that causes things to happen. A philosopher like Hume takes away the causality. Decentered psychologies take away the "I." Thus one sympathizes when Mark slips and reintroduces the "I." Others who have been caught up in multiprocessor models of mind take more deliberate actions to make them more "livable."

Some "recenter" their multiprocessor models by talking about one of the agents in the society of programs, usually one they call consciousness, as "not quite so dumb as the rest." When this happens, the other agents acquire a new role. Certain kinds of intelligence can still emerge from their interaction, but this intelligence is at the service of the smarter agent.

"I believe that the mind is just a computer," says Susan, "but one of the programs that runs on it gets to see the stuff that filters up from the dumber agents all of the time." This agent might start out "as dumb as all the rest," but, according to Susan, it has certain "pattern-recognition" properties and "it begins to grow in its ability to manipulate data." The dumb agents get to "talk to each other, but the consciousness agent has contact with the whole, and with the outside world, with all of the sensory systems." As a mediator, it develops new abilities. Freud's ego stood between id, superego, and impinging realities. Susan's theory replaces mind with a society of dumb agents, but her "pattern-recognizing" agent is an ego reborn.

According to Andy, one of the things that the "consciousness" agent knows how to do is to "flip off" its connection to the dumb processors. On the computer system he uses, this "interrupt" function is known as Control G. In Andy's model of how his mind
works, "Control C-ing" allows whatever bits of information happen to be stored in the consciousness agent to undergo reprocessing, "like putting them in a new stew—they get to mix around, interact more than they usually get to do when they just filter up to consciousness." Compared to the trillions of bits of information in the system at any one time, the bits that are made available for reprocessing are rare. But there are enough of them to allow a sense of free will to emerge. "When the system shuts off you do whatever comes out of the new mix. And it is heavily influenced by messages from your body, whether you are sexually turned on, whether you are tired or feeling exhilarated—it is influenced by all of these chemical things, whereas the dumb processors are usually isolated from all of that." In Andy's picture, the "I" emerges from egoless cognition touched by the sensual and animal.

Some of the students I interviewed think of their minds as computer systems, but they reconstitute an "I" in a more direct way than Andy. It is not just a question of a decentered system being periodically interrupted to allow for a moment of conscious action and free will. They split the system, putting a multiprocessor in charge of all cognitive functions, but reserving the "emotional" things for another power. "I see my mind as two systems that have worked out a process of peaceful or maybe not very peaceful coexistence," says Arlene, a computer-science major who is studying acting in her spare time.

There is a computational part, that's the part with the agents that somehow through their interaction have real intelligence come through. This part does my reasoning, my logic, my math homework, my ability to learn history. But then I have another system. It is built up from instincts. Evolution. My animal part. It is involved with love, feelings, relating to people. It can't control the computer part. But it lives with it. Sometimes fights with it. And this is the part that gives me the feeling of being me.

Amy is taking her first programming course. Three months into the course, she "discovered it... I realized how good I am at programming" and she became taken with "all the possible analogies between my mind and a computer. I heard about Minsky saying 'the mind is a meat machine,' and I thought that must be right." Now six months into the course, she is modifying her ideas about what kind of machine the mind might be. She thinks that there must be a special agent that intervenes in the messages "that come up from the brain's central processing unit." What makes the agent special is that it has its own sources of information. These sources are not purely rational: "You could think of it as God or you could call it evolution—I think that one of the agents has been influenced by evolution." She says the special agent acts as a "buffer" for final decision-making. The messages from the CPU rest there for a moment.

In general I see my mind in terms of continual processing by internal programs. But the weight given to the output of these programs can be influenced by emotion. And then when they come up to consciousness, they come up to a level where there is this other kind of agent—the special agent. The one in touch with my history and with evolution.

Theories that challenge the "I" need to be cast in more acceptable forms. We saw how in ego psychology psychoanalysis constructed a version of the unconscious acceptable to the conscious, one that brought psychoanalysis back into line with more traditional and reassuring models of mind. A similar process of normalization is at work in the case of machine models of mind. The students I interviewed used ideas about multiprocessing and emergence to describe how their minds worked. In their hands, the technical ideas that were developed within the artificial intelligence community were cast in nontechnical forms, such as the "agents at the trough." But then, there was a tension because these images of decentered minds do not sit easily with the sense of being a person and acting in the world.

The discrepancy between theory and the experience of the self shows itself in different ways. Mark does not recognize it at all, and it is visible only in the form of inconsistencies in his descriptions of his theory and himself. In the cases of the other students, Susan, Andy, Arlene, and Amy, the strain of fully accepting a computational model of the mind that dissolved the "I" was handled more explicitly. It led them to substantial modifications of the theory. They destroy the purity of the multiprocessing model; they are relying on ad-hoc fixes to soften the theory's fragmentation of self and destruction of the notion of free will.
Frank, a devout Catholic, an MIT junior in computer science who hopes someday to work in artificial intelligence research, does not rely on a “local fix” to make space for other commitments, in his case a commitment to the idea of soul. He thinks that the brain is hardware. He is not put off by Marvin Minsky’s description of it as a “meat machine.” The brain has some rudimentary capabilities—vision, sensory perception, instinctual reactions—and beyond this there are the familiar computational agents whose interaction could potentially lead to intelligence. But here is the point where Frank parts company with many of his friends. The complexity, the elegance, and the possibilities of the human mind machines are so great that souls come to inhabit them.

If you go along with a straight computational model, the idea of free will has to get dumped along with the idea that there is a self. That is, to say the least, disturbing to those of us who would like to think that we control our own actions. But wait. We do indeed control our own actions, but only if we consider ourselves to be our souls and not just the collection of neurons that form our individual meat machines. The soul exists. It has free will. It is I.

Frank is skeptical about multiprocessor theories that refer to emergent intelligence as requiring a “critical mass” of agents around. What is needed is not critical mass, but just the right kind of “delicate programming.” The combination of perfectly programmed agents is what could make intelligence emerge. There is an analogy with the early bath of organic molecules that gave birth to life. It had to be just the right molecules in the right proportions. The agents of the mind need to be delicately programmed, and this, says Frank, “is the province of the soul.” It is the soul that infuses the meat machine with what makes us human. But the soul has only one way of acting, and that is through its interaction with our hardware: “It twiddles a few bits here and there.” Frank sees the relationship of soul to brain as not quite a split between spirit and matter. It is more like the hacker to hardware.

The brain is a computer and the soul sort of programs. I think of it as a combination of doing some rewiring and keying in some bytes, but there’s another aspect to that, too. And that is that the soul, in addition to having the console in front of it, there would also be a direct line through the human form. This part is hard to say. It bothers me. But the soul is not in a simple relationship to the body. It is like a programmer and a computer. There is a harmony. A fit. The soul is sometimes sitting there at the console, at least one part of it is, and part of it is in the machine. It is continuously aware of the state of the machine and can change any part of it at will at any time. So I like to see that as emerging, the soul is not just typing in, it’s a spiritual thing which inhabits this computer.

Sometimes when Frank programs he gets the feeling that he is “part of the machine.” He knows “it is only like a hallucination,” but having had an experience of feeling both programmer and part of the thing being programmed makes it easier for him to imagine this kind of relationship for the soul in the machine. It certainly doesn’t seem impossible, not even implausible. When experiences with the computer make him feel he is building something that is also a part of him, he considers them points of contact with the experience of the soul. In this sense, working with computers has become part of his religious practice.

I run into Frank while he is in the middle of an argument with a friend, Mike, who challenges Frank’s attempt to reconcile his religious and computational views. If Frank really thinks that souls inhabit only human computers and that this is the natural order of things, isn’t artificial intelligence with its project for a non-human mind a doomed project, perhaps even a new kind of blasphemy? Frank’s answer surprises Mike and surprises me.

Souls reside in people because they are special, there hasn’t been anything else like them in the world. Nothing else with the computational capacity. But now what I feel is that we might, people might be at the stage where they can replicate their computational capabilities in their own manner. And then, if that happens, it may come to pass, if God allows it to happen, that after we connect together several trillion NAND gates or their equivalents, it may come to pass that some adventurous soul will decide to use that machine as its “receptacle” in this universe.*

---

* NAND gates are one of the elementary “logic components” for building digital circuits. They have a special property: they are universal. In principle, all other logic components can be built out of them.
Until the "soul invasion," the only intelligence we can create will be limited—a good chess program, a slightly better version of ELIZA, programs that operate by stock reasoning, mathematical analysis. But "it's all left-brain stuff—nothing that requires too much of the right brain. What it would not have are those things we cannot quantize. It would not have the ability to feel. It would be unemotional. It would certainly not be self-aware. These uniquely human functions are the province of the soul, and it is the soul that allows us to feel."

Frank has his heroes. Edwin Land, the inventor of instantaneous photography, the creator and until recently chairman of the board of the Polaroid Corporation, is one of them because he was something of a "hacker." In the enthusiasm of his identification with Land, Frank moves Land's alma mater from Harvard to closer to home. "He flunked out of MIT in order to devote himself to the things that MIT stands for. I like that." Marvin Minsky is another hero, and here the relationship is reverential. "The noblest of activities," says Frank, "is to create vessels for the wandering souls, but it can't just be a smart computer with a lot of knowledge stuffed in, a machine with a lot of showy programs and 'bells and whistles.' The aesthetic has to be right. "You have to want to capture the essence of humanity. What I like about Minsky is that he seems to want to make a computer that a soul would want to live in."

**Appropriate Models of Mind**

The question of what it takes for people to feel a kinship with computational views of mind is important. When there is this sense of kinship, there is the groundwork for the development of a new psychological culture, a "computational culture" with new metaphors for thinking about the mind as program and information processor. This culture spreads through the diffusion of computational ideas much as the psychoanalytic culture spread through the diffusion of Freudian ideas.

When we interpret our dreams or comment on our friends' slips of the tongue, we are weaving psychoanalysis into the world of the everyday. What kind of processes make theories of mind capable of "moving out" from the scientific and academic environments in which they are born to a larger culture? What makes a science of mind "appropriable"?

Freudian theory suggests the beginnings of an answer, one way to think about its own appeal. Consider Freud's theory of slips. It is a theory of why people make slips: we try to repress unacceptable wishes, but they break through all the same. By extension, I read it as a theory of why people like to think about slips: they allow us contact with these taboo wishes. And, most central to my concern, I read it as a theory of why people are attracted to Freudian ideas: Freudian interpretations offer us a way to come closer to aspects of ourselves, like sexuality and aggression, which we censor but at the same time want to be in contact with, the uncivilized that makes us human.

Computer models are seductive because they too put us in contact with issues that are both threatening and fascinating. The question here is not which theory, the psychoanalytic or the computational, is true, but rather how these very different ways of thinking about ourselves capture our imagination. Behind the popular acceptance of the Freudian theory was a nervous, often guilty preoccupation with the self as sexual; behind the widespread interest in computational interpretations is an equally nervous preoccupation with the self as a machine. Playing with psychoanalytic and computational theories allows us to play with aspects of our nature that we experience as taboo.

People are afraid to think of themselves as machines, that they are controlled, predictable, determined, just as they are afraid to think of themselves as "driven" by sexual or aggressive impulses. But in the end, even if fearful, people want to explore their sexual and aggressive dimensions; hence, the evocative power and popular appeal of psychoanalytic ideas. Similarly, although fearful, people want to find a way to think about what they experience as the machine aspect of their natures; this is at the heart of the computer's holding power. Thinking about the self as a machine includes the feeling of being "run" from the outside, out of control because in the control of something beyond the self. Exploring the parts of ourselves that we do not feel in control of is a way to begin to own them, a way to feel more whole.

Thus, the "Freudian" explanation for Freud's appeal explains part of the appeal of computational models: each model allows us to think about highly charged materials. Another element that makes many computational models appealing is more dependent
on their form than on their content. "Appropriable" theories of mind, ideas that move out into the culture at large, tend to be those in which we can become "actively" involved. They tend to be theories that we can "play with."

In the case of psychoanalysis, what we play with are "Freudian objects" such as dreams, slips, and jokes. We become used to looking for them and manipulating them, both seriously and playfully. And as we do so, psychoanalytic theory, or at least the idea that there is an unconscious, starts to feel "natural." In the case of computational models of mind, there are what I call "carrier objects" that encompass both the physical computer and ideas that grow out of programming.

Recall, for example, what is at work to help people appropriate Donald Norman's computational model of slips and, through it, the idea of mind as computer. According to the model, people, like complex computer programs, become momentarily derailed and make slips. And when people "slip" it doesn't reflect sexual fantasies or forbidden desires. It simply reveals information overload and the mechanisms of computation.

You pour ketchup into your coffee instead of cream. Norman lets you imagine a computer in your place and think of its workings as a process involving numerous subprograms. One program is called up to locate the cream, another to locate the coffee, another to get a fix on the location of the hand, another to get it going in the right direction, and others to check that it is on the right path, to verify that the position of hand, creamer, cup have not changed, and so on. In this buzz of activity it is hardly surprising that one of these programs might go wrong. At a given moment, the program that verifies that the position of X has not changed might "mistakenly" take X to be ketchup instead of cream, because another program on which it depends is still registering ketchup as "the object of greatest salience," a morsel of information left over from a previous phase of the meal during which ketchup was appropriately being poured on a cheeseburger. Once you have identified with the computer you cannot fail to identify with the error, since each of us knows that we make many more when "we" are consciously involved in a scenario as complex as this one. We are able to identify with the programs and so become sympathetic to their confusion.

Norman's slips theory is "appropriable" because its description of how a computer works sounds so much like a confused person that people identify with the computer to the point of believing that it makes slips for the same reason they do. From here it is a small step to feeling that we make slips because we are computers and that we are computers because of the way we make slips.

As a formal, logical argument this is circular. But as a description of how people adopt a point of view, of how they weave it into their intuitions, it is quite straightforward. Norman's theory found its way out of academic journals—a version of it appeared in Psychology Today and then in Reader's Digest—and was widely read. People easily imagine themselves in the role of the procedures, and acting out these roles feels enough like Norman's theory to give the sense of understanding it. One may forget the details of the theory, but it offers an experience of "playing computer" and of feeling comfortable in the role that has a longer-lasting effect. The experience breaks down resistance to seeing mind as machine.

The computational theory of slips is the kind of theory that passes easily into the general culture. It is able to take a nontechnical form, people can imagine themselves "in the theory," it provides a handle to think about the everyday, and it touches on emotionally charged concerns. Mark's theory of the dumb multiprocessing agents had all of these qualities. Like Norman's, it anthropomorphized machine processes so that it was easy to identify with them, one could put oneself in the place of agents around the trough, it was a way to think about fragmented identity, a problematic aspect of the self. Such experiences are the stuff upon which a computer culture is built.

Mark's computational theory of psychology is personal. He has given little thought to where it stands in relation to other theories. But a growing culture needs something more. It needs to be conscious of itself as a culture, it needs symbols, it needs heroes, it needs antiheroes. Such elements exist in the worlds of computer subcultures. The hackers had LISP as symbol of the true, the beautiful, and the good; IBM's FORTRAN was their symbol of degeneracy. One sure sign of the movement of the computer culture into the larger world is the appearance of a literature that carries these out to a larger public.

Douglas Hofstadter's 1978 Gödel, Escher, Bach: An Eternal Golden Braid is an example of that literature. It affirms computation as a culture—along with its symbols, its language, its humor, its ways of
breaking down resistance to the idea of mind as machine. Hofstadter begins his book with a declaration of purpose:

Computers by their very nature are the most inflexible, desireless, rule-following of beasts. Fast though they may be, they are nonetheless the epitome of unconsciousness. How, then, can intelligent behavior be programmed? Isn't this the most blatant of contradictions in terms? One of the major theses of this book is to urge each reader to confront the apparent contradiction head on, to savor it, to turn it over, to take it apart, to wallow in it, so that in the end the reader might emerge with new insights into the seemingly unbreachable gulf between the formal and the informal, the animate and the inanimate, the flexible and the inflexible.17

Hofstadter goes about this by providing puzzles, thought experiments, riddles, and dialogues, many of them dedicated to offering a way to address a stumbling block to the idea that the mind is a computer: "If machines are perfect and I am human, how can my mind be a computer?" The answer: the perfect machine is by its nature imperfect.

The mathematician and the logician know this idea in the technical form of Gödel's theorem. After a period in the late nineteenth and early twentieth centuries when a proliferation of increasingly powerful formal methods created hopes of a universal mathematics that could solve all mathematical problems, Kurt Gödel produced a dramatic theorem that scuttled once and for all any hopes that there could be such a thing.18 In principle, says Gödel, every formal system is limited. Any mathematics that would be sufficiently powerful to seem to promise completeness, universality, would necessarily be incomplete: it would be possible to ask it questions that could not be answered using its methods. This idea that a system is incomplete because it is strong goes against common sense.

Gödel's thoroughly mathematical proof that a formal system (this could be a machine) is vulnerable because strong had been accessible only to a mathematical elite. Hofstadter brings it to a larger circle. His strategies are numerous. Most powerful among them is a technique he borrows from Lewis Carroll: dialogues between characters that Carroll himself used, for example Achilles and the Tortoise, to which Hofstadter adds others as necessary, among these the Crab, a personage who is the subject of an infinitely repeating drawing by Escher.

In my favorite dialogue, Achilles is talking to the Tortoise about whether it is possible to have a perfect phonograph.19 The answer turns out to be no. And the way of discussing the possibilities for phonograph perfection, like those for mathematical perfection and machine perfection, begins with the idea of self-reference.

A crystal glass can be shattered by a singer who hits exactly the right note. By extension, for any piece of matter, including a phonograph, there is a pitch that will shatter it. If a record whose "song" contains this pitch is played on the phonograph, the machine will "self destruct." Thus any phonograph must be imperfect in one of two ways. It can be imperfect because it is unable to play notes with enough fidelity to reproduce the destructive tone, in which case it does not destroy itself, but is obviously imperfect, imperfect through weakness. Or it can be imperfect through strength, in which case it is able to reproduce notes in full fidelity, and the record with that "certain note" will shatter it to pieces.

The paradox at the heart of Gödel's theorem is this: if the formal system is really powerful there will be a question that can be posed within it which it cannot answer. If the phonograph is really perfect, it will be able to reproduce the sound that would shatter it to pieces. And, indeed, the Tortoise has been driving his friend the Crab to distraction by shattering an increasingly expensive series of phonographs with specially designed records that he calls "Music to Break Phonographs By."

"Music to Break Phonographs By," like the proof of Gödel's theorem, plays on the idea of self-reference.* In the history of mathematics, self-reference has been controversial. Bertrand Russell and Alfred North Whitehead, among many other logicians,

---

* Gödel wanted to prove that arithmetic is "incomplete." The problem is to find a statement about numbers that cannot be proved or disproved. Gödel found one by drawing on the famous paradox of self-reference. If a proposition P says that the proposition P is false, by an argument known since the time of the Greeks such a proposition can be neither true nor false. Assume P is true. P asserts its own falsity. So, if true, it must be false. Assume it is false. It says it is false, so it must be true. Gödel found an arithmetical equivalent of this paradox. He assigned code numbers to propositions in such a way that a certain proposition P says that the proposition whose code number is n cannot be proved. And when we look up the code we find that n is the code number for the proposition P itself. The proposition P thus declares itself to be unprovable. Proving it would be absurd, disproving it equally so.
had placed a ban on it: propositions that referred to themselves (like the statement "This statement is false") were declared meaningless. But Gödel found ways of doing indirectly something akin to what had been outlawed, smuggling enough self-reference back into arithmetic to show that there is at least one proposition in it that cannot be proven true or proven false.

Gödel made a breach in the wall erected by logicians against self-reference, but it remained small because Gödel's propositions seemed esoteric. The computer presence enlarged the breach because, in programming, self-reference is an everyday matter of the greatest practical use, a powerful tool for building complex programs. Within the AI and hacker community, Gödel's theorem became a symbol that "we," the computer culture, had won a battle against "them," the logicians. Gödel's "hack" had broken the system. Hofstadter dramatized the struggle and put his readers in a position to feel participant in the triumph of a new intellectual aesthetic. The mighty had fallen. Russell and Whitehead had been sacrificed on the altar of computational logic.

The growing computer culture draws its strength from awareness of its roots and from aggressive assertion of continuity with the culture into which it is penetrating. By placing self-reference at the center of his intellectual world Hofstadter is able to recruit Bach and Escher into the computer culture. Bach (who used it in his endlessly rising canons) and Escher (who used it in his endlessly rising staircases) are made into cultural heroes just as Russell and Whitehead are made into cultural enemies. After over seven hundred pages of Gödel, Escher, Bach, Hofstadter's readers have little sympathy with the traditional logicians, not just because they wanted to ban paradox, but because they belonged to an intellectual culture that saw in paradox problems rather than power. Hofstadter succeeds in getting his readers to sense themselves as part of a new culture, a computer culture, strong enough to shrug off the culture of Russell, Whitehead, and traditional philosophy and logic.

Hofstadter began Gödel, Escher, Bach with a declaration of purpose: to bridge the gap between the mental and the mechanical. Through the phonograph story—if you are sufficiently strong you are by definition vulnerable and incomplete—and his many other examples of what he calls "strange loops," Hofstadter gives people a language for talking about the questions that come up when we consider whether we are machines. "If we are machines, are we perfect?" asks Frank, the student who put the soul into the machine. No, he comes back saying, "if we are the most powerful kind of machines, the kind that God would create," we are limited, vulnerable, weak. "You see, Professor Turkle, what Achilles and the Tortoise and the Crab are saying is that if we are machines, we are human."