INTRODUCTION
(Not) Like a Rock

Here's how January 21, 2000 panned out for three different elements of the natural order.

Element 1: A Rock

Here is a day in the life of a small, gray-white rock nestling amidst the ivy in my St. Louis backyard. It stayed put. Some things happened to it: there was rain, and it became wet and shiny; there was wind, and it was subtly eroded; my cat chased a squirrel nearby, and this made the rock sway. That's about it, really. There is no reason to believe the rock had any thoughts, or that any of this felt like anything to the rock. Stuff happened, but that was all.

Element 2: A Cat

Lolo, my cat, had a rather different kind of day. About 80% of it was spent, as usual, asleep. But there were forays into the waking, wider world. Around 7 A.M. some inner stirring led Lolo to exit the house, making straight for the catflap from the warm perch of the living room sofa. Outside, bodily functions doubtless dominated, at least at first. Later, following a brief trip back inside (unerringly routed via the catflap and the food tray), squirrels were chased and dangers avoided. Other cats were dealt with in ways appropriate to their rank, station, girth, and meanness. There was a great deal of further sleeping.

Element 3: Myself

My day was (I think) rather more like Lolo's than like the rock's. We both (Lolo and I) pursued food and warmth. But my day included, I suspect, rather more outright

contemplation. The kind of spiraling meta-contemplation, in fact, that has sometimes gotten philosophy a bad name. Martin Amis captured the spirit well:

> I experienced thrilling self-pity. "What will that mind of your get up to next?" I said, recognizing the self-congratulation behind this thought and the self-congratulation behind that recognition, and the self-congratulation behind recognizing that recognition.
>
> Steady on. (Martin Amis, *The Rachel Papers*, p. 96)

I certainly did some of that. I had thoughts, even "trains of thought" (reasonable sequences of thinkings such as "It's 1 P.M. Time to eat. What's in the fridge?" and so on). But there were also thoughts about thoughts, as I sat back and observed my own trains of thought, alert for colorful examples to import into this text.

What, then, distinguishes cat from rock, and (perhaps) person from cat? What are the mechanisms that make thought and feeling possible? And what further tricks or artifices give my own kind of mindfulness its peculiar self-aware tinge? Such questions seem to focus attention on three different types of phenomena:

1. The feelings that characterize daily experience (hunger, sadness, desire, and so on)
2. The flow of thoughts and reasons
3. The meta-flow of thoughts about thoughts (and thoughts about feelings), of reflection on reasons, and so on.

Most of the research programs covered in this text have concentrated on the middle option. They have tried to explain how my thought that it is 1 P.M. could lead to my thought about lunch, and how it could cause my subsequent lunch-seeking actions. All three types of phenomena are, however, the subject of what philosophers call "mentalistic discourse." A typical example of mentalistic discourse is the appeal to beliefs (and desires) to explain actions. The more technical phrase "propositional attitude psychology" highlights the standard shape of such explanations: such explanations pair mental attitudes (believing, hoping, fearing, etc.) with specific propositions ("that it is raining," "that the coffee is in the kitchen," "that the squirrel is up the tree," etc.) so as to explain intelligent action. Thus in a sentence such as "Pepa hopes that the wine is chilled," the that-construction introduces a proposition ("the wine is chilled") toward which the agent is supposed to exhibit some attitude (in this case, hoping). Other attitudes (such as believing, desiring, fearing, and so on) may, of course, be taken to the same proposition. Our everyday understandings of each other's behavior involve hefty doses of propositional attitude ascription: for example, I may explain Pepa's reluctance to open the wine by saying "Pepa believes that the wine is not yet chilled and desires that it remain in the fridge for a few more minutes."

Such ways of speaking (and thinking) pay huge dividends. They support a surprising degree of predictive success, and are the common currency of many of our social and practical projects. In this vein, the philosopher Jerry Fodor suggests that commonsense psychology is *ubiquitous*, almost *invisible* (because it works so well), and practically *indispensable*. For example, it enables us to make precise plans on the basis of someone's 2-month-old statement that they will arrive on flight 594 on Friday, November 20, 1999. Such plans often work out—a truly amazing fact given the number of physical variables involved. They work out (when they do) because the statement reflects an intention (to arrive that day, on that flight) that is somehow an active shaper of my behavior. I desire that I should arrive on time. You know that I so desire. And on that basis, with a little cooperation from the world at large, miracles of coordination can occur. Or as Fodor more colorfully puts it:

> If you want to know where my physical body will be next Thursday, mechanics—our best science of middle-sized objects after all, and reputed to be pretty good in its field—is *no use to you at all*. Far the best way to find out (usually in practice, the only way to find out) is: *ask me*! (Fodor, 1987, p. 6, original emphasis)

Commonsense psychology thus works, and with a vengeance. But why? Why is it that treating each other as having beliefs, hopes, intentions, and the like allows us successfully to explain, predict, and understand so much daily behavior? Beliefs, desires, and so on are, after all, invisible. We see (what we take to be) their effects. But no one has ever actually seen a belief. Such things are (currently? permanently?) unobservable. Commonsense psychology posits these unobservables, and looks to be committed to a body of law-like relations involving them. For example, we explain Fred's jumping up and down by saying that he is happy because his sister just won the Nobel Prize. Behind this explanation lurks an implicit belief in a law-like regularity, viz. "if someone desires x, and x occurs, then (all other things being equal) they feel happy." All this makes commonsense psychology look like a theory about the invisible, *but causally potent*, roots of intelligent behavior. What, then, can be making the theory true (assuming that it is)? What *is* a belief (or a hope, or a fear) such that it can cause a human being (or perhaps a cat, dog, etc.) to act in an appropriate way?

Once upon a time, perhaps, it would have been reasonable to respond to the challenge by citing a special kind of spirit-substance: the immaterial but causally empowered seat of the mental [for some critical discussion, see Churchland (1984), pp. 7–22, and Appendix I of the present text]. Our concerns, however, lie squarely with attempts that posit nothing extra—nothing beyond the properties and organization of the material brain, body, and world. The goal is a fully materialistic story in which mindware emerges as *nothing but* the playing out of ordinary physical states and processes in the familiar physical world. Insofar as the mental is in any way *special*, according to these views, it is special because it depends on some

particular and unusual ways in which ordinary physical stuff can be built, arranged, and organized.

Views of this latter kind are broadly speaking *monistic*: that is to say, they posit only one basic *kind* of stuff (the material stuff) and attempt to explain the distinctive properties of mental phenomena in terms that are continuous with, or at least appropriately grounded in, our best understanding of the workings of the nonmental universe. A common, but still informative, comparison is with the once-lively (sic) debate between vitalists and nonvitalists. The vitalist held that living things were quite fundamentally different from the rest of inanimate nature, courtesy of a special extra force or ingredient (the "vital spark"), that was missing elsewhere. This is itself a kind of dualism. The demonstration of the fundamental unity of organic and inorganic chemistry (and the absence, in that fundament, of anything resembling a vital spark) was thus a victory—as far as we can tell—for a kind of monism. The animate world, it seems, is the result of *nothing but* the fancy combination of the same kinds of ingredients and forces responsible for inanimate nature. As it was with the animate, so materialists (which is to say, nearly all those working in contemporary cognitive science, the present author included) believe it must be with the mental. The mental world, it is anticipated, must prove to depend on nothing but the fancy combination and organization of ordinary physical states and processes.

Notice, then, the problem. The mental certainly *seems* special, unusual, and different. Indeed, as we saw, it *is* special, unusual, and different: thoughts give way to other thoughts and actions in a way that *respects reasons*: the thought that the forecast was sun (to adapt the famous but less upbeat example) causes me to apply sunscreen, to don a Panama hat, and to think "just another day in paradise." And there is a qualitative feel, a "something it is like" to have a certain kind of mental life: I *experience* the stabbings of pain, the stirrings of desire, the variety of tastes, colors, and sounds. It is the burden of materialism to somehow get to grips with these various special features in a way that is continuous with, or appropriately grounded in, the way we get to grips with the rest of the physical world—by some understanding of material structure, organization, and causal flow. This is a tall order, indeed. But, as Jerry Fodor is especially fond of pointing out, there is at least one good idea floating around—albeit one that targets just one of the two special properties just mentioned: reason-respecting flow.

The idea, in a supercompressed nutshell, is that the power of a thought (e.g., that the forecast is sun) to cause further thoughts and actions (to apply sunscreen, to think "another day in paradise") is fully explained by what are broadly speaking *structural* properties of the system in which the thought occurs. By a structural property I here mean simply a physical or organizational property: something whose nature is explicable *without* invoking the specific thought-content involved. An example will help. Consider the way a pocket calculator outputs the sum of two numbers given a sequence of button pushings that we interpret as inputting "2"

"+" "2." The calculator need not (and does not) understand anything about numbers for this trick to work. It is simply structured so that those button pushings will typically lead to the output "4" as surely as a river will typically find the path of least resistance down a mountain. It is just that in the former case, but not the latter, there has been a process of design such that the physical stuff became organized *so as* its physical unfoldings would reflect the arithmetical constraints governing sensible (arithmetic-respecting) transitions in number space. Natural selection and lifetime learning, to complete the (supercompressed) picture, are then imagined to have sculpted our *brains* so that certain structure-based physical unfoldings respect the constraints on sensible sequences of thoughts and sensible thought–action transitions. Recognition of the predator thus causes running, hiding, and thoughts of escape, whereas recognition of the food causes eating, vigilance, and thoughts of where to find more. Our whole reason-respecting mental life, so the story goes, is just the unfolding of what is, at bottom, a physical and structural story. Mindfulness is just matter, nicely orchestrated.

(As to that *other* distinctive property, "qualitative feel," let's just say—and see Appendix II—that it's a problem. Maybe that too is just a property of matter, nicely orchestrated. But how the orchestration *yields* the property is in this case much less clear, even in outline. So we'll be looking where the light is.)

In the next eight chapters, I shall expand and pursue that simple idea of mindware (selected aspects!) as matter, nicely orchestrated. The chase begins with a notion of mind as a kind of souped-up pocket calculator (mind as a familiar kind of computer, but built out of meat rather than silicon). It proceeds to the vision of mind as dependent on the operation of a radically different *kind* of computational device (the kind known as artificial neural networks). And it culminates in the contemporary (and contentious) research programs that highlight the complex interactions among brains, bodies, and environmental surroundings (work on robotics, artificial life, dynamics, and situated cognition).

The narrative is, let it be said, biased. It reflects my own view of what we have learned in the past 30 or 40 years of cognitive scientific research. What we have learned, I suggest, is that there are many deeply different ways to put flesh onto that broad, materialistic framework, and that some once-promising incarnations face deep and unexpected difficulties. In particular, the simple notion of the brain as a kind of symbol-crunching computer is probably too simple, and too far removed from the neural and ecological realities of complex, time-critical interaction that sculpted animal minds. The story I tell is thus a story of (a kind of) *inner symbol flight.* But it is a story of progress, refinement, and renewal, not one of abandonment and decay. The sciences of the mind are, in fact, in a state of rude health, of exuberant flux. Time, then, to start the story, to seek the origins of mind in the whirr and buzz of well-orchestrated matter.

1 MEAT MACHINES
Mindware as Software

1.1 Sketches

1.2 Discussion

- A. Why Treat Thought as Computation?
- B. Is Software an Autonomous Level in Nature?
- C. Mimicking, Modeling, and Behavior
- D. Consciousness, Information, and Pizza

1.3 A Diversion

1.4 Suggested Readings

1.1 Sketches

The computer scientist Marvin Minsky once described the human brain as a meat machine—no more no less. It is, to be sure, an ugly phrase. But it is also a striking image, a compact expression of both the genuine scientific excitement and the rather gung-ho materialism that tended to characterize the early years of cognitive scientific research. Mindware—our thoughts, feelings, hopes, fears, beliefs, and intellect—is cast as nothing but the operation of the biological brain, the meat machine in our head. This notion of the brain as a meat *machine* is interesting, for it immediately invites us to focus not so much on the material (the meat) as on the machine: the way the material is organized and the kinds of operation it supports. The same machine (see Box 1.1) can, after all, often be made of iron, or steel, or tungsten, or whatever. What we confront is thus both a rejection of the idea of mind as immaterial spirit-stuff and an affirmation that mind is best studied from a kind of engineering perspective that reveals the nature of the machine that all that wet, white, gray, and sticky stuff happens to build.

What exactly is meant by casting the brain as a machine, albeit one made out of meat? There exists a historical trend, to be sure, of trying to understand the workings of the brain by analogy with various currently fashionable technologies: the telegraph, the steam engine, and the telephone switchboard are all said to have had their day in the sun. But the "meat machine" phrase is intended, it should now be clear, to do more than hint at some rough analogy. For with regard to the very special class of machines known as computers, the claim is that the brain (and, by

Box 1.1

THE "SAME MACHINE"

In what sense can "the same machine" be made out of iron, or steel, or whatever? Not, obviously, in the strict sense of numerical identity. A set of steel darts and a set of tungsten ones cannot be the *very same* (numerically identical) set of darts. The relevant sense of sameness is, rather, some sense of *functional* sameness. You can make a perfectly *good* set of darts out of either material (though not, I suppose, out of jello), just as you can make a *perfectly* good corkscrew using a myriad (in this latter case quite radically) different designs and materials. In fact, what *makes* something a corkscrew is simply that it is designed as, and is capable of acting as, a cork-removing device. The notion of a brain as a meat machine is meant to embody a similar idea: that what matters about the brain is not the stuff it is made of but the way that stuff is organized so as to support thoughts and actions. The idea is that this capability depends on quite abstract properties of the physical device that could very well be duplicated in a device made, say, out of wires and silicon. Sensible versions of this idea need not claim then that *any* material will do: perhaps, for example, a certain stability over time (a tendency not to rapidly disorganize) is needed. The point is just that given that certain preconditions are met the same functionality can be pressed from multiple different materials and designs. For some famous opposition to this view, see Searle (1980, 1992).

not unproblematic extension, the mind) actually *is* some such device. It is not that the brain is somehow *like* a computer: everything is like everything else in some respect or other. It is that neural tissues, synapses, cell assemblies, and all the rest are just nature's rather wet and sticky way of building a hunk of honest-to-God computing machinery. Mindware, it is then claimed, is found "in" the brain in just the way that software is found "in" the computing system that is running it.

The attractions of such a view can hardly be overstated. It makes the mental special without making it ghostly. It makes the mental depend on the physical, but in a rather complex and (as we shall see) liberating way. And it provides a ready-made answer to a profound puzzle: how to get sensible, reason-respecting behavior out of a hunk of physical matter. To flesh out this idea of nonmysterious reason-respecting behavior, we next review some crucial developments[1] in the history (and prehistory) of artificial intelligence.

[1]The next few paragraphs draw on Newell and Simon's (1976) discussion of the development of the Physical Symbol Hypothesis (see Chapter 2 following), on John Haugeland's (1981a), and on Glymour, Ford, and Hayes' (1995).

One key development was the appreciation of the power and scope of formal logics. A decent historical account of this development would take us too far afield, touching perhaps on the pioneering efforts in the seventeenth century by Pascal and Leibniz, as well as on the twentieth-century contributions of Boole, Frege, Russell, Whitehead, and others. A useful historical account can be found in Glymour, Ford, and Hayes (1995). The idea that shines through the history, however, is the idea of finding and describing "laws of reason"—an idea whose clearest expression emerged first in the arena of formal logics. Formal logics are systems comprising sets of symbols, ways of joining the symbols so as to express complex propositions, and rules specifying how to legally derive new symbol complexes from old ones. The beauty of formal logics is that the steadfast application of the rules guarantees that you will never legally infer a false conclusion from true premises, even if you have no idea what, if anything, the strings of symbols actually mean. Just follow the rules and truth will be preserved. The situation is thus a little (just a little) like a person, incompetent in practical matters, who is nonetheless able to successfully build a cabinet or bookshelf by following written instructions for the manipulation of a set of preprovided pieces. Such building behavior can look as if it is rooted in a deep appreciation of the principles and laws of woodworking: but in fact, the person is just blindly making the moves allowed or dictated by the instruction set.

Formal logics show us how to preserve at least one kind of semantic (meaning-involving: see Box 1.2) property without relying on anyone's actually appreciating the meanings (if any) of the symbol strings involved. The seemingly ghostly and ephemeral world of meanings and logical implications is respected, and in a certain sense recreated, in a realm whose operating procedures do not rely on meanings at all! It is recreated as a realm of marks or "tokens," recognized by their physical ("syntactic") characteristics alone and manipulated according to rules that refer only to those physical characteristics (characteristics such as the shape of the symbol—see Box 1.2). As Newell and Simon comment:

> Logic . . . was a game played with meaningless tokens according to certain purely syntactic rules. Thus progress was first made by walking away from all that seemed relevant to meaning and human symbols. (Newell and Simon, 1976, p. 43)

Or, to put it in the more famous words of the philosopher John Haugeland:

> If you take care of the syntax, *the semantics will take care of itself.* (Haugeland, 1981a, p. 23, original emphasis)

This shift from meaning to form (from semantics to syntax if you will) also begins to suggest an attractive liberalism concerning actual physical structure. For what matters, as far as the identity of these formal systems is concerned, is not, e.g., the precise shape of the symbol for "and." The shape could be "AND" or "and" or "&" or "∧" or whatever. All that matters is that the shape is used consistently and that the rules are set up so as to specify how to treat strings of symbols joined by that shape: to allow, for example, the derivation of "A" from the string "A and

Box 1.2

SYNTAX AND SEMANTICS

Semantic properties are the "meaning-involving" properties of words, sentences, and internal representations. *Syntactic* properties, at least as philosophers tend to use the term, are nonsemantic properties of, e.g., written or spoken words, or of any kinds of inscriptions of meaningful items (e.g., the physical states that the pocket calculator uses to store a number in memory). Two synonymous written words ("dog" and "chien") are thus semantically identical but syntactically distinct, whereas ambiguous words ("bank" as in river or "bank" as in high street) are syntactically identical but semantically distinct. The idea of a *token* is the idea of a specific syntactic item (e.g., *this* occurrence of the word "dog"). A pocket calculator manipulates physical tokens (inner syntactic states) to which the operation of the device is sensitive. It is by being sensitive to the distinct syntactic features of the inner tokens that the calculator manages to behave in an arithmetic-respecting fashion: it is set up *precisely* so that syntax-driven operations on inner tokens standing for numbers respect meaningful arithmetical relations between the numbers. Taking care of the syntax, in Haugeland's famous phrase, thus allows the semantics to take care of itself.

B." Logics are thus first-rate examples of *formal systems* in the sense of Haugeland (1981a, 1997). They are systems whose essence lies not in the precise physical details but in the web of legal moves and transitions.

Most games, Haugeland notes, are formal systems in exactly this sense. You can play chess on a board of wood or marble, using pieces shaped like animals, movie stars, or the crew of the star ship Enterprise. You could even, Haugeland suggests, play chess using helicopters as pieces and a grid of helipads on top of tall buildings as the board. All that matters is again the web of legal moves and the physical distinguishability of the tokens.

Thinking about formal systems thus liberates us in two very powerful ways at a single stroke. Semantic relations (such as truth preservation: if "A and B" is true, "A" is true) are seen to be respected in virtue of procedures that make no intrinsic reference to meanings. And the specific physical details of any such system are seen to be unimportant, since what matters is the golden web of moves and transitions. Semantics is thus made unmysterious without making it brute physical. Who says you can't have your cake and eat it?

The next big development was the formalization (Turing, 1936) of the notion of computation itself. Turing's work, which predates the development of the dig-

ital computer, introduced the foundational notion of (what has since come to be known as) the Turing machine. This is an imaginary device consisting of an infinite tape, a simple processor (a "finite state machine"), and a read/write head. The tape acts as data store, using some fixed set of symbols. The read/write head can read a symbol off the tape, move itself one square backward or forward on the tape, and write onto the tape. The finite state machine (a kind of central processor) has enough memory to recall what symbol was just read and what state it (the finite state machine) was in. These two facts together determine the next action, which is carried out by the read/write head, and determine also the next state of the finite state machine. What Turing showed was that some such device, performing a sequence of simple computations governed by the symbols on the tape, could compute the answer to any sufficiently well-specified problem (see Box 1.3).

We thus confront a quite marvelous confluence of ideas. Turing's work clearly suggested the notion of a physical machine whose syntax-following properties would enable it to solve any well-specified problem. Set alongside the earlier work on logics and formal systems, this amounted to nothing less than

> . . . the emergence of a new level of analysis, independent of physics yet mechanistic in spirit . . . a science of structure and function divorced from material substance. (Pylyshyn, 1986, p. 68)

Thus was classical cognitive science conceived. The vision finally became flesh, however, only because of a third (and final) innovation: the actual construction of general purpose electronic computing machinery and the development of flexible, high-level programming techniques. The bedrock machinery (the digital computer) was designed by John von Neumann in the 1940s and with its advent all the pieces seemed to fall finally into place. For it was now clear that once realized in the physical medium of an electronic computer, a formal system could run *on its own*, without a human being sitting there deciding how and when to apply the rules to initiate the legal transformations. The well-programmed electronic computer, as John Haugeland nicely points out, is really just an automatic ("self-moving") formal system:

> It is like a chess set that sits there and plays chess by itself, without any intervention from the players, or an automatic formal system that writes out its own proofs and theorems without any help from the mathematician. (Haugeland, 1981a, p. 10; also Haugeland, 1997, pp. 11–12)

Of course, the machine needs a program. And programs were, in those days (but see Chapter 4), written by good old-fashioned human beings. But once the program was in place, and the power on, the machine took care of the rest. The transitions between legal syntactic states (states that also, under interpretation, *meant* something) no longer required a human operator. The physical world suddenly included clear, nonevolved, nonorganic examples of what Daniel Dennett would later dub "syntactic engines"—quasiautonomous systems whose sheer physical make-

BOX 1.3

A TURING MACHINE

To make the idea of Turing machine computation concrete, let us borrow an example from Kim (1996, pp. 80–85). Suppose the goal is to get a Turing machine to add positive numbers. Express the numbers to be added as a sequence of the symbols "#" (marking the beginning and end of numbers) "1" and "+." So the sum 3 + 2 is encoded on the tape as shown in Figure 1.1. A neat program for adding the numbers (where "Λ A" indicates the initial location and initial state of the read/write head) is as follows:

Instruction 1: If read-write head is in machine state A and encounters a "1," it moves one square to the right, and the head stays in state A.

Instruction 2: If the head is in state A and encounters a "+," it replaces it with a "1," stays in state A, and moves one square to the right.

Instruction 3: If the head is in state A and it encounters a "#," move one square left and go into machine state B.

Instruction 4: If the head is in machine state B and encounters a "1," delete it, replace with a "#," and halt.

You should be able to see how this works. Basically, the machine starts "pointed" at the leftmost "1." It scans right seeking a "+," which it replaces with a "1." It continues scanning right until the "#" indicates the end of the sum, at which point it moves one square left, deletes a single "1," and replaces it with a "#." The tape now displays the answer to the addition problem in the same notation used to encode the question, as shown in Figure 1.2.

Similar set-ups (try to imagine how they work) can do subtraction, multiplication, and more (see Kim, 1996, pp. 83–85). But Turing's most strik-

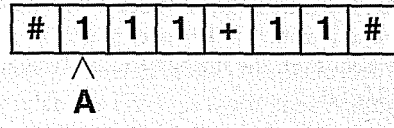

Figure 1.1 (After Kim, 1996, p. 81.)

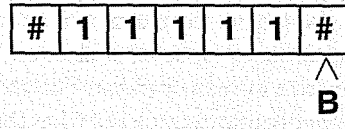

Figure 1.2 (After Kim, 1996, p. 81.)

ing achievement in this area was to show that you could then define a special kind of Turing machine (the aptly-named universal Turing machine) able to imitate any other Turing machine. The symbols on the tape, in this universal case, encode a description of the behavior of the other machine. The universal Turing machine uses this description to mimic the input–output function of any other such device and hence is itself capable of carrying out *any* sufficiently well-specified computation. (For detailed accounts see Franklin, 1995; Haugeland, 1985; Turing, 1936, 1950.)

The Turing machine affords a fine example of a simple case in which syntax-driven operations support a semantics-respecting (meaning-respecting) process. Notice also that you could *build* a simple Turing machine out of many different materials. It is the formal (syntactic) organization that matters for its semantic success.

up ensured (under interpretation) some kind of ongoing reason-respecting behavior. No wonder the early researchers were jubilant! Newell and Simon nicely capture the mood:

> It is not my aim to surprise or shock you. . . . But the simplest way I can summarize is to say that there are now in the world machines that think, that learn and that create. Moreover, their ability to do these things is going to increase rapidly until—in a visible future—the range of problems they can handle will be co-extensive with the range to which the human mind has been applied. (Newell and Simon, 1958, p. 6, quoted in Dreyfus and Dreyfus, 1990, p. 312)

This jubilant mood deepened as advanced programming techniques[2] brought forth impressive problem-solving displays, while the broader theoretical and philosophical implications (see Box 1.4) of these early successes could hardly have been more striking. The once-mysterious realm of mindware (represented, admittedly, by just two of its many denizens: truth preservation and abstract problem solving) looked ripe for conquest and understanding. Mind was not ghostly stuff, but the operation of a formal, computational system implemented in the meatware of the brain.

Such is the heart of the matter. Mindware, it was claimed, is to the neural meat machine as software is to the computer. The brain may be the standard (local, earthly, biological) implementation—but cognition is a program-level thing. Mind

[2]For example, list-processing languages, as pioneered in Newell and Simon's Logic Theorist program in 1956 and perfected in McCarthy's LISP around 1960, encouraged the use of more complex "recursive programming" strategies in which symbols point to data structures that contain symbols pointing to further data structures and so on. They also made full use of the fact that the same electronic memory could store both program and data, a feature that allowed programs to be modified and operated on in the same ways as data. LISP even boasted a universal function, EVAL, that made it as powerful, modulo finite memory limitations, as a Universal Turing Machine.

Box 1.4

MACHINE FUNCTIONALISM

The leading philosophical offspring of the developments in artificial intelligence went by the name of machine functionalism, and it was offered as an answer to one of the deepest questions ever asked by humankind, viz. what is the essence (the deep nature) of the mental? What fundamental facts make it the case that some parts of the physical world have mental lives (thoughts, beliefs, feelings, and all the rest) and others do not? Substance dualists, recall, thought that the answer lay in the presence or absence of a special kind of mental *stuff*. Reacting against this idea (and against so-called philosophical behaviorism—see Appendix I). Mind–brain identity theorists, such as Smart (1959) (and again, see Appendix I), claimed that mental states *just are* processes going on in the brain. This bald identity claim, however, threatened to make the link between mental states and specific, material brain states a little too intimate. A key worry (e.g., Putnam, 1960, 1967) was that if it was really essential to being in a certain mental state that one be in a specific brain state, it would seem to follow that creatures lacking brains built just like ours (say, Martians or silicon-based robots) could not be in those very same mental states. But surely, the intuition went, creatures with very different brains from ours could, at least in principle, share, e.g., the belief that it is raining. Where, then, should we look for the commonality that could unite the robot, the Martian, and the Bostonian? The work in logic and formal systems, Turing machines, and electronic computation now suggested an answer: look not to the specific physical story (of neurons and wetware), nor to the surface behavior, but to the inner organization, that is to say, to the golden web: to the abstract, formal organization of the system. It is this organization—depicted by the machine functionalists as a web of links between possible inputs, inner computational states, and outputs (actions, speech)—that fixes the shape and contents of a mental life. The building materials do not matter: the web of transitions could be realized in flesh, silicon, or cream cheese (Putnam, 1975, p. 291). To be in such and such a mental state is simply to be a physical device, of whatever composition, that satisfies a specific formal description. Mindware, in humans, happens to run on a meat machine. But the very same mindware (as picked out by the web of legal state transitions) might run in some silicon device, or in the alien organic matter of a Martian.

is thus ghostly enough to float fairly free of the gory neuroscientific details. But it is not so ghostly as to escape the nets of more abstract (formal, computational) scientific investigation. This is an appealing story. But is it correct? Let's worry.

1.2 Discussion

(A brief note of reassurance: many of the topics treated below recur again and again in subsequent chapters. At this point, we lack much of the detailed background needed to really do them justice. But it is time to test the waters.)

A. WHY TREAT THOUGHT AS COMPUTATION?

Why treat thought as computation? The principal reason (apart from the fact that it seems to work!) is that thinkers are physical devices whose behavior patterns are reason respecting. Thinkers act in ways that are usefully understood as sensitively guided by reasons, ideas, and beliefs. Electronic computing devices show us one way in which this strange "dual profile" (of physical substance and reason-respecting behavior) can actually come about.

The notion of reason-respecting behavior, however, bears immediate amplification. A nice example of this kind of behavior is given by Zenon Pylyshyn. Pylyshyn (1986) describes the case of the pedestrian who witnesses a car crash, runs to a telephone, and punches out 911. We could, as Pylyshyn notes, try to explain this behavior by telling a purely physical story (maybe involving specific neurons, or even quantum events, whatever). But such a story, Pylyshyn argues, will not help us understand the behavior in its *reason-guided* aspects. For example, suppose we ask: what would happen if the phone was dead, or if it was a dial phone instead of a touch-tone phone, or if the accident occurred in England instead of the United States? The neural story underlying the behavioral response will differ widely if the agent dials 999 (the emergency code in England) and not 911, or must run to find a working phone. Yet common sense psychological talk makes sense of all these options at a stroke by depicting the agent as seeing a crash and *wanting to get help.* What we need, Pylyshyn powerfully suggests, is a scientific story that remains in touch with this more abstract and reason-involving characterization. And the simplest way to provide one is to imagine that the agent's brain contains states ("symbols") that represent the event *as* a car crash and that the computational state-transitions occurring inside the system (realized as physical events in the brain) then lead to new sets of states (more symbols) whose proper interpretation is, e.g., "seek help," "find a telephone," and so on. The interpretations thus glue inner states to sensible real-world behaviors. Cognizers, it is claimed, "instantiate . . . representation physically as cognitive codes and . . . their behavior is a causal consequence of operations carried out on those codes" (Pylyshyn, 1986, p. xiii).

The same argument can be found in, e.g., Fodor (1987), couched as a point about content-determined transitions in trains of thought, as when the thought "it

raining" leads to the thought "let's go indoors." This, for Fodor (but see Chaprs 4 onward), is the essence of human rationality. How is such rationality meanically possible? A good empirical hypothesis, Fodor suggests, is that there are eural symbols (inner states apt for interpretation) that mean, e.g., "it is raining" id whose physical properties lead in context to the generation of other symbols iat mean "let's go indoors." If that is how the brain works then the brain is ineed a computer in exactly the sense displayed earlier. And if such were the case, ien the mystery concerning reason-guided (content-determined) transitions in iought is resolved:

> If the mind is a sort of computer, we begin to see how . . . there could be nonarbitrary content-relations among causally related thoughts. (Fodor, 1987, p. 19)

Such arguments aim to show that the mind *must* be understood as a kind of omputer implemented in the wetware of the brain, on pain of failing empirically o account for rational transitions among thoughts. Reason-guided action, it seems, nakes good scientific sense if we imagine a neural economy organized as a syntaxlriven engine that tracks the shape of semantic space (see, e.g., Fodor, 1987, pp. 19–20).

B. IS SOFTWARE AN AUTONOMOUS LEVEL IN NATURE?

The mindware/software equation is as beguiling as it is, at times, distortive. One immediate concern is that all this emphasis on algorithms, symbols, and programs tends to promote a somewhat misleading vision of *crisp level distinctions in nature.* The impact of the theoretical independence of algorithms from hardware is an artifact of the long-term neglect of issues concerning real-world action taking and the time course of computations. For an algorithm or program as such is just a sequence of steps with no inbuilt relation to real-world timing. Such timing depends crucially on the particular way in which the algorithm is implemented on a real device. Given this basic fact, the theoretical independence of algorithm from hardware is unlikely to have made much of an impact on Nature. We must expect to find biological computational strategies closely tailored to getting useful real-time results from available, slow, wetware components. In practice, it is thus unlikely that we will be able to fully appreciate the formal organization of natural systems without some quite detailed reference to the nature of the neural hardware that provides the supporting implementation. In general, attention to the nature of real biological hardware looks likely to provide both important clues about and constraints on the kinds of computational strategy used by real brains. This topic is explored in more depth in Chapters 4 through 6.

Furthermore, the claim that mindware is software is—to say the least—merely schematic. For the space of possible types of explanatory story, all broadly computational (but see Box 1.5), is very large indeed. The comments by Fodor and by

Box 1.5

What Is Computation?

It is perhaps worth mentioning that the foundational notion of computation is itself still surprisingly ill understood. What do we really mean by calling some phenomenon "computational" in the first place? There is no current consensus at least (in the cognitive scientific community) concerning the answer to this question. It is mostly a case of "we know one when we see one." Nonetheless, there is a reasonable consensus concerning what I'll dub the "basic profile," which is well expressed by the following statement:

> we count something as a computer because, and only when, its inputs and outputs can be usefully and systematically interpreted as representing the ordered pairs of some function that interests us. (Churchland and Sejnowski, 1992, p. 65)

Thus consider a pocket calculator. This physical device computes, on this account, because first, there is a reliable and systematic way of interpreting various states of the device (the marks and numerals on the screen and keyboard) as representing other things (numbers). And second, because the device is set up so that under that interpretation, its physical state changes mirror semantic (meaningful) transitions in the arithmetical domain. Its physical structure thus forces it to respect mathematical constraints so that inputs such as "4 × 3" lead to outputs such as "12" and so on.

A truly robust notion of the conditions under which some actual phenomenon counts as computational would require, however, some rather more *objective* criterion for determining when an encountered (nondesigned) physical process is actually implementing a computation—some criterion that does not place our interpretive activities and interests so firmly at center stage.

The best such account I know of is due to Dave Chalmers (1996, Chapter 9). Chalmers' goal is to give an "objective criterion for implementing a computation" (p. 319). Intuitively, a physical device 'implements' an abstract, formal computational specification just in case the physical device is set up to undergo state changes that march in step with those detailed in the specification. In this sense a specific word-processing program might, for example, constitute a formal specification that can (appropriately configured) be made to run on various kinds of physical device (MACS, PCs, etc.).

Chalmers' proposal, in essence, is that a physical device implements an abstract formal description (a specification of states and state-transition relations) just in case "the causal structure of the system mirrors the formal

structure of the computation" (1996, p. 317). The notion of mirroring is then cashed out in terms of a fairly fine-grained mapping of states and state changes in the physical device onto the elements and transitions present in the abstract specification. Chalmer's allows that every physical system will implement some computational description. But the appeal to fine-grained mappings is meant to ensure that you cannot interpret *every* physical system as implementing *every* computational description. So although the claim that the brain implements *some* computational description is fairly trivial, the claim that it implements a *specific* computational description is not. And it is the brain's implementation of a specific computational description that is meant to explain mental properties.

The computational profile of most familiar devices is, of course, the result of the deliberate imposition of a mapping, via some process of intelligent design. But the account is not intrinsically so restricted. Thus suppose some creature has evolved organic inner states that represent matters of adaptive importance such as the size, number, and speed of approach of predators. If that evolutionary process results in a physical system whose causal state transitions, under that interpretation, make semantic sense (e.g., if fewer than two predators detected cause a "stand and fight" inner token leading to aggressive output behavior, whereas three or more yield a "run and hide" response), then Nature has, on this account, evolved a small computer. The brain, if the conjectures scouted earlier prove correct, is just such a natural computer, incorporating inner states that represent external events (such as the presence of predators) and exploiting state-transition routines that make sensible use of the information thus encoded.

Pylyshyn do, it is true, suggest a rather specific kind of computational story (one pursued in detail in the next chapter). But the bare explanatory schema, in which semantic patterns emerge from an underlying syntactic, computational organization, covers a staggeringly wide range of cases. The range includes, for example, standard artificial intelligence (A.I.) approaches involving symbols and rules, "connectionist" approaches that mimic something of the behavior of neural assemblies (see Chapter 4), and even Heath Robinsonesque devices involving liquids, pulleys, and analog computations. Taken very liberally, the commitment to understanding mind as the operation of a syntactic engine can amount to little more than a bare assertion of physicalism—the denial of spirit-stuff.[3]

To make matters worse, a variety of different computational stories may be told about one and the same physical device. Depending on the grain of analysis

[3]Given our notion of computation (see Box 1.5), the claim is just a little stronger, since it also requires the presence of systematically interpretable inner states, i.e., internal representations.

used, a single device may be depicted as carrying out a complex parallel search or as serially transforming an input x into an output y. Clearly, what grain we choose will be determined by what questions we hope to answer. Seeing the transition as involving a nested episode of parallel search may help explain specific error profiles or why certain problems take longer to solve than others, yet treating the process as a simple unstructured transformation of x to y may be the best choice for understanding the larger scale organization of the system. There will thus be a constant interaction between our choice of explanatory targets and our choice of grain and level of computational description. In general, there seems little reason to expect a single type or level of description to do all the work we require. Explaining the relative speed at which we solve different problems, and the kinds of interference effects we experience when trying to solve several problems at once (e.g., remembering two closely similar telephone numbers), may well require explanations that involve very specific details about how inner representations are stored and structured, whereas merely accounting for, e.g., the bare facts about rational transitions between content-related thoughts may require only a coarser grained computational gloss. [It is for precisely this reason that connectionists (see Chapter 4) describe themselves as exploring the microstructure of cognition.] The explanatory aspirations of psychology and cognitive science, it seems clear, are sufficiently wide and various as to require the provision of explanations at a variety of different levels of grain and type.

In sum, the image of mindware as software gains its most fundamental appeal from the need to accommodate reason-guided transitions in a world of merely physical flux. At the most schematic level, this equation of mindware and software is useful and revealing. But we should not be misled into believing either (1) that "software" names a single, clearly understood level of neural organization or (2) that the equation of mindware and software provides any deep warrant for cognitive science to ignore facts about the biological brain.

C. MIMICKING, MODELING, AND BEHAVIOR

Computer programs, it often seems, offer only shallow and brittle simulacrums of the kind of understanding that humans (and other animals) manage to display. Are these just teething troubles, or do the repeated shortfalls indicate some fundamental problem with the computational approach itself? The worry is a good one. There are, alas, all too many ways in which a given computer program may merely mimic, but not illuminate, various aspects of our mental life. There is, for example, a symbolic A.I. program that does a very fine job of mimicking the verbal responses of a paranoid schizophrenic. The program ("PARRY," Colby, 1975; Boden, 1977, Chapter 5) uses tricks such as scanning input sentences for key words (such as "mother") and responding with canned, defensive outbursts. It is capable, at times, of fooling experienced psychoanalysts. But no one would claim that

it is a useful psychological model of paranoid schizophrenia, still less that it is (when up and running on a computer) a paranoid schizophrenic itself!

Or consider a chess computer such as Deep Blue. Deep Blue, although capable of outstanding play, relies heavily on the brute-force technique of using its superfast computing resources to examine all potential outcomes for up to seven moves ahead. This strategy differs markedly from that of human grandmasters, who seem to rely much more on stored knowledge and skilled pattern recognition (see Chapter 4). Yet, viewed from a certain height, Deep Blue is not a bad simulation of human chess competence. Deep Blue and the human grandmaster are, after all, more likely to agree on a particular move (as a response to a given board state) than are the human grandmaster and the human novice! At the level of gross input–output profiles, the human grandmaster and Deep Blue are thus clearly similar (not identical, as the difference in underlying strategy—brute force versus pattern recognition—sometimes shines through). Yet once again, it is hard to avoid the impression that all that the machine is achieving is top-level mimicking: that there is something amiss with the underlying strategy that either renders it unfit as a substrate for a real intelligence, or else reveals it as a kind of intelligence very alien to our own.

This last caveat is important. For we must be careful to distinguish the question of whether such and such a program constitutes a good model of *human* intelligence from the question of whether the program (when up and running) displays some kind of *real, but perhaps nonhuman* form of intelligence and understanding. PARRY and Deep Blue, one feels, fail on both counts. Clearly, neither constitutes a faithful psychological model of the inner states that underlie human performance. And something about the basic style of these two computational solutions (canned sentences activated by key words, and brute-force look-ahead) even makes us uneasy with the (otherwise charitable) thought that they might nonetheless display real, albeit alien, kinds of intelligence and awareness.

How, though, are we to decide what kinds of computational substructure *might* be appropriate? Lacking, as we must, first-person knowledge of what (if anything) it is like to be PARRY or Deep Blue, we have only a few options. We could insist that all real thinkers must solve problems using exactly the same kinds of computational strategy as human brains (too anthropocentric, surely). We could hope, optimistically, for some future scientific understanding of the *fundamentals* of cognition that will allow us to recognize (on broad theoretical grounds) the shape of alternative, but genuine, ways in which various computational organizations might support cognition. Or we could look to the gross behavior of the systems in question, insisting, for example, on a broad and flexible range of responses to a multiplicity of environmental demands and situations. Deep Blue and PARRY would then fail to make the grade not merely because their inner organizations looked alien to us (an ethically dangerous move) but because the behavioral repertoire they support is too limited. Deep Blue cannot recognize a mate (well, only a check-

mate!), nor cook an omelette. PARRY cannot decide to become a hermit or take up the harmonica, and so on.

This move to behavior is not without its own problems and dangers, as we will see in Chapter 3. But it should now be clearer why some influential theorists (especially Turing, 1950) argued that a sufficient degree of behavioral success should be allowed to settle the issue and to establish once and for all that a candidate system is a genuine thinker (albeit one whose inner workings may differ greatly from our own). Turing proposed a test (now known as the Turing Test) that involved a human interrogator trying to spot (from verbal responses) whether a hidden conversant was a human or a machine. Any system capable of fooling the interrogator in ongoing, open-ended conversation, Turing proposed, should be counted as an intelligent agent. Sustained, top-level verbal behavior, if this is right, is a sufficient test for the presence of real intelligence. The Turing Test invites consideration of a wealth of issues that we cannot dwell on here (several surface in Chapter 3). It may be, for example, that Turing's original restriction to a verbal test leaves too much scope for "tricks and cheats" and that a better test would focus more heavily on real-world activity (see Harnad, 1994).

It thus remains unclear whether we should allow that surface behaviors (however complex) are sufficient to distinguish (beyond all theoretical doubt) real thinking from mere mimicry. Practically speaking, however, it seems less morally dangerous to allow behavioral profiles to lead the way (imagine that it is discovered that you and you alone have a mutant brain that uses brute-force, Deep Blue-like strategies where others use quite different techniques: has science discovered that *you* are not a conscious, thinking, reasoning being after all?).

D. CONSCIOUSNESS, INFORMATION, AND PIZZA

"If one had to describe the deepest motivation for materialism, one might say that it is simply a terror of consciousness" (Searle, 1992, p. 55). Oh dear. If I had my way, I would give in to the terror and just not mention consciousness at all. But it is worth a word or two now (and see Appendix II) for two reasons. One is because it is all too easy to see the facts about conscious experience (the "second aspect of the problem of mindfulness" described in the Introduction) as constituting a knock-down refutation of the strongest version of the computationalist hypothesis. The other is because consideration of these issues helps to highlight important differences between informational and "merely physical" phenomena. So here goes.

How could a device made of silicon be conscious? How could it feel pain, joy, fear, pleasure, and foreboding? It certainly seems unlikely that such exotic capacities should flourish in such an unusual (silicon) setting. But a moment's reflection should convince you that it is equally amazing that such capacities should show up in, of all things, meat (for a sustained reflection on this theme, see the skit in Section 1.3). It is true, of course, that the only known cases of conscious

awareness on this planet *are* cases of consciousness in carbon-based organic life forms. But this fact is rendered somewhat less impressive once we realize that all earthly life forms share a common chemical ancestry and lines of descent. In any case, the question, at least as far as the central thesis of the present chapter is concerned, is not whether our local carbon-based organic structure is crucial to all possible versions of conscious awareness (though it sounds anthropocentric in the extreme to believe that it is), but whether meeting a certain abstract computational specification is enough to *guarantee* such conscious awareness. Thus even the philosopher John Searle, who is famous for his attacks on the equation of mind-ware with software, allows that "consciousness might have been evolved in systems that are not carbon-based, but use some other sort of chemistry altogether" (Searle, 1992, p. 91). What is at issue, it is worth repeating, is not whether other kinds of stuff and substance might support conscious awareness but whether the fact that a system exhibits a certain computational profile is enough (is "sufficient") to ensure that it has thoughts, feelings, and conscious experiences. For it is crucial to the strongest version of the computationalist hypothesis that where our mental life is concerned, *the stuff doesn't matter.* That is to say, mental states depend solely on the program-level, computational profile of the system. If conscious awareness were to turn out to depend much more closely than this on the nature of the actual physical stuff out of which the system is built, then this global thesis would be either false or (depending on the details) severely compromised.

Matters are complicated by the fact that the term "conscious awareness" is something of a weasel word, covering a variety of different phenomena. Some use it to mean the high-level capacity to reflect on the contents of one's own thoughts. Others have no more in mind that the distinction between being awake and being asleep! But the relevant sense for the present discussion (see Block, 1997; Chalmers, 1996) is the one in which to be conscious is to be a subject of experience—to feel the toothache, to taste the bananas, to smell the croissant, and so on. To experience some x is thus to do more than just register, recognize, or respond to x. Electronic detectors can register the presence of semtex and other plastic explosives. But, I hope, they have no experiences of so doing. A sniffer dog, however, may be a different kettle of fish. Perhaps the dog, like us, is a subject of experience; a haven of what philosophers call "qualia"—the qualitative sensations that make life rich, interesting, or intolerable. Some theorists (notably John Searle) believe that computational accounts fall down at precisely this point, and that as far as we can tell it is the implementation, not the program, that explains the presence of such qualitative awareness. Searle's direct attack on computationalism is treated in the next chapter. For now, let us just look at two popular, but flawed, reasons for endorsing such a skeptical conclusion.

The first is the observation that "simulation is not the same as instantation." A rainstorm, simulated in a computational medium, does not make anything actually wet. Likewise, it may seem obvious that a simulation, in a computational

medium, of the brain states involved in a bout of black depression will not add one single iota (thank heaven) to the sum of real sadness in the world.

The second worry (related to, but not identical to the first) is that many feelings and emotions look to have a clear chemical or hormonal basis and hence (hence?) may be resistant to reproduction in any merely electronic medium. Sure, a silicon-based agent can play chess and stack crates, but can it get drunk, get an adrenaline high, experience the effects of ecstasy and acid, and so on?

The (genuine) intuitive appeal of these considerations notwithstanding, they by no means constitute the knock-down arguments they may at first appear. For everything here depends on what *kind* of phenomenon consciousness turns out to be. Thus suppose the skeptic argues as follows: "even if you get the overall inner computational profile just right, and the system behaves just like you and I, it will still be lacking the inner baths of chemicals, hormones, and neurotransmitters, etc. that flood our brains and bodies. Maybe without these all is darkness within—it just looks like the "agent" has feelings, emotions, etc., but really it is just [what Haugeland (1981a) terms] a "hollow shell." This possibility is vividly expressed in John Searle's example of the person who, hoping to cure a degenerative brain disease, allows parts of her brain to be gradually replaced by silicon chips. The chips preserve the input–output functions of the real brain components. One logical possibility here, Searle suggests, is that "as the silicon is progressively implanted into your dwindling brain, you find that the area of your conscious experience is shrinking, but that this shows no effect on your external behavior" (Searle, 1992, p. 66). In this scenario (which is merely one of several that Searle considers), your actions and words continue to be generated as usual. Your loved ones are glad that the operation is a success! But from the inside, you experience a growing darkness until, one day, nothing is left. There is no consciousness there. You are a zombie.

The imaginary case is problematic, to say the least. It is not even clear that we here confront a genuine logical possibility. [For detailed discussion see Chalmers (1996) and Dennett (1991a)—just look up zombies in the indexes!] Certainly the alternative scenario in which you *continue* your conscious mental life with no ill effects from the silicon surgery strikes many cognitive scientists (myself included) as the more plausible outcome. But the "shrinking consciousness" nightmare does help to focus our attention on the right question. The question is, just *what is* the role of all the hormones, chemicals, and organic matter that build normal human brains? There are two very different possibilities here and, so far, no one knows which is correct. One is that the chemicals, etc. affect our conscious experiences *only by affecting* the way information flows and is processed in the brain. If that were the case, the same kinds of modulation may be achieved in other media by other means. Simplistically, if some chemical's effect is, e.g., to speed up the processing in some areas, slow it down in others, and allow more information leakage between adjacent sites, then perhaps the same effect may be achieved in a purely electronic medium, by some series of modulations and modifications of current

flow. Mind-altering "drugs," for silicon-based thinkers, may thus take the form of black-market software packages—packages that temporary induce a new pattern of flow and functionality in the old hardware.

There remains, however, a second possibility: perhaps the experienced nature of our mental life is not (or is not just) a function of the flow of information. Perhaps it is to some degree a direct effect of some still-to-be-discovered physical cause or even a kind of basic property of some types of matter (for extended discussion of these and other possibilities, see Chalmers, 1996). If this were true, then getting the information-processing profile exactly right would still fail to guarantee the presence of conscious experience.

The frog at the bottom of the beer glass is thus revealed. The bedrock, unsolved problem is whether conscious awareness is an *informational* phenomenon. Consider the difference. A lunch order is certainly an informational phenomenon. You can phone it, fax it, E-mail it—whatever the medium, it is the same lunch order. But no one ever faxes you your lunch. There is, of course, the infamous Internet Pizza Server. You specify size, consistency, and toppings and await the on-screen arrival of the feast. But as James Gleick recently commented, "By the time a heavily engineered software engine delivers the final product, you begin to suspect that they've actually forgotten the difference between a pizza and *a picture* of a pizza" (Gleick, 1995, p. 44). This, indeed, is Searle's accusation in a nutshell. Searle believes that the conscious mind, like pizza, just *ain't an informational phenomenon.* The stuff, like the topping, really counts. This could be the case, notice, even if many of the *other* central characteristics of mindware reward an understanding that is indeed more informational than physical. Fodor's focus on reason-guided state-transitions, for example, is especially well designed to focus attention away from qualitative experience and onto capacities (such as deciding to stay indoors when it is raining) that can be visibly guaranteed once a suitable formal, functional profile is fixed.

We are now eyeball to eyeball with the frog. To the extent that mind is an informational phenomenon, we may be confident that a good enough computational simulation will yield an actual instance of mindfulness. A good simulation of a calculator is an instance of a calculator. It adds, subtracts, does all the things we expect a calculator to do. Maybe it even follows the same hidden procedures as the original calculator, in which case we have what Pylyshyn (1986) terms "strong equivalence"—equivalence at the level of an underlying program. If a phenomenon is informational, strong equivalence is surely sufficient[4] to guarantee that we confront not just a model (simulation) of something, but a new exemplar (in-

[4]Sufficient, but probably not necessary. x is sufficient for y if when x obtains, y always follows. Being a banana is thus a sufficient condition for being a fruit. x is necessary for y if, should x fail to obtain, y cannot be the case. Being a banana is thus not a necessary condition for being a fruit—being an apple will do just as well.

stantiation) of that very thing. For noninformational phenomena, such as "being a pizza," the rules are different, and the flesh comes into its own. Is consciousness like calculation, or is it more like pizza? The jury is still out.

1.3 A Diversion

[This is extracted from a story by Terry Bisson called "Alien/Nation" first published in *Omni* (1991). Reproduced by kind permission of the author.]

"They're made out of meat."

"Meat?"

"Meat. They're made out of meat."

"Meat?"

"There's no doubt about it. We picked several from different parts of the planet, took them aboard our recon vessels, probed them all the way through. They're completely meat."

"That's impossible. What about the radio signals? The messages to the stars."

"They use the radio waves to talk, but the signals don't come from them. The signals come from machines."

"So who made the machines? That's who we want to contact."

"They made the machines. That's what I'm trying to tell you. Meat made the machines."

"That's ridiculous. How can meat make a machine? You're asking me to believe in sentient meat."

"I'm not asking you, I'm telling you. These creatures are the only sentient race in the sector and they're made out of meat."

"Maybe they're like the Orfolei. You know, a carbon-based intelligence that goes through a meat stage."

"Nope. They're born meat and they die meat. We studied them for several of their life spans, which didn't take too long. Do you have any idea of the life span of meat?"

"Spare me. Okay, maybe they're only part meat. You know, like the Weddilei. A meat head with an electron plasma brain inside."

"Nope. We thought of that, since they do have meat heads like the Weddilei. But I told you, we probed them. They're meat all the way through."

"No brain?"

"Oh, there is a brain all right. It's just that the brain is made out of meat!"

"So . . . what does the thinking?"

"You're not understanding, are you? The brain does the thinking. The meat."

"Thinking meat! You're asking me to believe in thinking meat!"

"Yes, thinking meat! Conscious meat! Loving meat. Dreaming meat. The meat is the whole deal! Are you getting the picture?"

"Omigod. You're serious then. They're made out of meat."

"Finally, Yes. They are indeed made out of meat. And they've been trying to get in touch with us for almost a hundred of their years."

"So what does the meat have in mind?"

"First it wants to talk to us. Then I imagine it wants to explore the universe, contact other sentients, swap ideas and information. The usual."

"We're supposed to talk to meat?"

"That's the idea. That's the message they're sending out by radio. Hello. Anyone out there? Anyone home? That sort of thing."

"They actually do talk, then. They use words, ideas, concepts?"

"Oh, yes. Except they do it with meat."

"I thought you just told me they used radio."

"They do, but what do you think is on the radio? Meat sounds. You know how when you slap or flap meat it makes a noise? They talk by flapping their meat at each other. They can even sing by squirting air through their meat."

"Omigod. Singing meat. This is altogether too much. So what do you advise?"

"Officially or unofficially?"

"Both."

"Officially, we are required to contact, welcome, and log in any and all sentient races or multi beings in the quadrant, without prejudice, fear, or favor. Unofficially, I advise that we erase the records and forget the whole thing."

"I was hoping you would say that."

"It seems harsh, but there is a limit. Do we really want to make contact with meat?"

"I agree one hundred percent. What's there to say?" 'Hello, meat. How's it going?' But will this work? How many planets are we dealing with here?"

"Just one. They can travel to other planets in special meat containers, but they can't live on them. And being meat, they only travel through C space. Which limits them to the speed of light and makes the possibility of their ever making contact pretty slim. Infinitesimal, in fact." "So we just pretend there's no one home in the universe."

"That's it."

"Cruel. But you said it yourself, who wants to meet meat? And the ones who have been aboard our vessels, the ones you have probed? You're sure they won't remember?"

"They'll be considered crackpots if they do. We went into their heads and smoothed out their meat so that we're just a dream to them."

"A dream to meat! How strangely appropriate, that we should be meat's dream."

"And we can mark this sector unoccupied."

"Good. Agreed, officially and unofficially. Case closed. Any others? Anyone interesting on that side of the galaxy?"

"Yes, a rather shy but sweet hydrogen core cluster intelligence in a class nine star in G445 zone. Was in contact two galactic rotations ago, wants to be friendly again."

"They always come around."

"And why not? Imagine how unbearably, how unutterably cold the universe would be if one were all alone."

1.4 Suggested Readings

For an up-to-date, and indeed somewhat sympathetic, account of the *varieties of dualism*, see D. Chalmers, *The Conscious Mind* (New York: Oxford University Press, 1996, Chapter 4).

For *general philosophical background* (identity theory, behaviorism, machine functionalism) a good place to start is Appendix I of this text and then P. M. Churchland, *Matter & Consciousness* (Cambridge, MA: MIT Press, 1984, and subsequent expanded editions). Another excellent resource is D. Braddon-Mitchell and F. Jackson, *Philosophy of Mind and Cognition* (Oxford, England: Blackwell, 1996, Chapters 1, 2, 3, 5, 6, and 7).

For the *broad notion of a computational view of mind*, try the Introductions to J. Haugeland, *Mind Design*, 1st ed. (Cambridge, MA: MIT Press, 1981) and *Mind Design II* (Cambridge, MA: MIT Press, 1997). The former ("Semantic engines: An introduction to mind design") is especially good on the syntax/semantics distinction, and the latter ("What is mind design?") adds useful discussion of recent developments.

For more on *Turing machines*, see J. Kim, "Mind as a computer," [Chapter 4 of his excellent book, *Philosophy of Mind* (Boulder, CO: Westview Press, 1996)]. Chapters 1–3 cover *dualism, behaviorism, and identity theory* and are also highly recommended. Chapter 4 focuses on the advent of *machine functionalism* and includes detailed discussion of the antireductionist themes that surface as the "structure not stuff" claim discussed in our text.

For *philosophical accounts of machine functionalism, and critiques*, see H. Putnam, "The nature of mental states." In H. Putnam (ed.), *Mind, Language & Reality: Philosophical Papers*, Vol. 2 (Cambridge, England: Cambridge University Press, 1975) (a classic and very readable account of machine functionalism) and N. Block, "Introduction: What is functionalism?" and "Troubles with functionalism." Both in his *Readings in Philosophy of Psychology*, Vol. 1 (Cambridge, MA: Harvard University Press, 1980). (Clean and critical expositions that nicely reflect the flavor of the original debates.)

J. Searle, "The critique of cognitive reason," Chapter 9 of his book, *The Rediscovery of the Mind* (Cambridge, MA: MIT Press, 1992) is a characteristically direct *critique of the basic computationalistic claims* and assumptions.

A useful, *up-to-date introduction to the empirical issues* is S. Franklin, *Artificial Minds* (Cambridge, MA: MIT Press, 1995), and an excellent general *collection of papers* may be found in J. Haugeland, *Mind Design II* (Cambridge, MA: MIT Press, 1997).

2 SYMBOL SYSTEMS

2.1 Sketches
2.2 Discussion
A. The Chinese Room
B. Everyday Coping
C. Real Brains and the Bag of Tricks
2.3 Suggested Readings

2.1 Sketches

> The study of logic and computers has revealed to us that intelligence resides in physical-symbol systems. This is computer science's most basic law of qualitative structure. (Newell and Simon, 1976, p. 108)

The equation of mindware with software (Chapter 1) found clear expression and concrete computational substance in a flurry of work on *physical-symbol systems.* A physical-symbol system, as defined by Newell and Simon (1976, pp. 85–88) is a physical device that contains a set of interpretable and combinable items (symbols) and a set of processes that can operate on the items (copying, conjoining, creating, and destroying them according to instructions). To ensure that the symbols have meanings and are not just empty syntactic shells, the device must be located in a wider web of real-world items and events. Relative to this wider web, a symbolic expression will be said to pick out (or designate) an object if "given the expression, the system can either affect the object itself or behave in ways depending on the object" (Newell and Simon, 1976, p. 86). Given this specification, Newell and Simon make a bold claim:

> *The Physical Symbol System Hypothesis.* A physical symbol system has the necessary and sufficient means for general intelligent action. (Newell and Simon, 1976, p. 87)

The claim, in less formal language, is that a symbol cruncher of the kind just sketched possesses all that matters for thought and intelligence. Any such machine "of sufficient size" can (it is argued) always be programmed so as to support intelligent behavior, hence being a physical-symbol system is *sufficient* for intelligence. And nothing can be intelligent unless it is an instance of a physical-symbol

system (PSS), so being a PSS is also a *necessary* condition for "general intelligent behavior." As Newell and Simon are quick to stress, we thus confront a strong *empirical* hypothesis. The notion of a PSS is meant to delimit a class of actual and potential systems and the claim is that all cases of general intelligent action will, as a matter of scientific fact, turn out to be produced by members of that class.

So just what *is* that class? The question is, unfortunately, more difficult than it at first appears. Clearly, we are being told that intelligent behavior depends on (and only on) processes that are broadly computational in the sense described in Chapter 1. That is to say, they involve inner states that can be organized so as to preserve semantic sense. Moreover, there is a commitment to the existence of inner *symbols* that are not just any old inner states capable of systematic interpretation, but that are in addition capable of participating in processes of copying, conjoining, and other familiar types of internal manipulation. It is this kind of inner economy, in which symbols exist as stable entities that are moved, copied, conjoined, and manipulated, that has *in practice* most clearly characterized work in the PSS paradigm and that differentiates it from the bare notion of mindware as software

Nonetheless, it is important to be clear about what this commitment to inner symbols actually involves. It is a commitment to the existence of a computational symbol-manipulating regime *at the level of description most appropriate to understanding the device as a cognitive (reasoning, thinking) engine.* This claim is thus fully compatible with the discovery that the brain is at bottom some other kind of device. What matters is not the computational profile at the hardware level, but the one "higher up" at the level of what is sometimes called a "virtual machine." (This is like saying: "don't worry about the form of the machine code—look at the elements and operations provided by some higher level language.") It is at this higher, virtual level that the system must provide the set of symbols and symbol-manipulating capacities associated with classical computation (copying, reading and amending symbol strings, comparing currently generated symbol strings to target sequences, and so on). In some cases these symbols will be systematically interpretable in ways that line up with our intuitive ideas about the elements of the task domain. For example, a program for reasoning about the behavior of liquids may use procedures defined over symbols for items such as "liquid," "flow," "edge," "viscous," and so on (see, e.g., Hayes 1979, 1985). Or a chess-playing program may use procedures applied to symbols for rook, king, checkmate, etc., whereas a sentence parser might use symbols for noun, verb, subject, and so on. These kinds of symbols reflect our own ideas about the task domain (chess, liquids, whatever). Systems whose computational operations are defined over this type of familiar symbolic elements may be termed *semantically transparent systems* (Clark, 1989, p. 17). The great advantage of semantically transparent systems, it should be clear, is that they make it immediately obvious *why* the physical device is able to respect specific semantic regularities. It is obvious that getting such symbols to behave ap-

Box 2.1

THE RESTAURANT SCRIPT

Schank's (1975) program could, for example, infer that someone who eats and enjoys a restaurant meal will probably have left a tip. It does so by referring to a background knowledge base encoding the "script" for a stereotypic restaurant visit. The script uses symbols for standard events and a special symbolic code for action types. In the extract below, "PTrans" stands for the change of location of an object and "Atrans" signifies the transfer of a relationship, e.g., my money becomes the waitresses' money in Scene 4. Here, then, is the script:

Script: Restaurant
Roles: customer; waitress; chef; cashier.
Reason: to get food so as to go down in hunger and up in pleasure.

Scene 1: ENTERING
PTRANS: go into restaurant
MBUILD: find table
PTRANS: go to table
MOVE: sit down
Scene 2: ORDERING
ATRANS: receive menu
ATTEND: look at it
MBUILD: decide on order
MTRANS: tell order to waitress
Scene 3: EATING
ATRANS: receive food
INGEST: eat food
Scene 4: EXITING
MTRANS: ask for check
ATRANS: give tip to waitress
PTRANS: go to cashier
MTRANS: give money to cashier
PTRANS: go out of restaurant
(Schank, 1975, p. 131, quoted in Dreyfus, 1997, pp. 167–168)

Basically, then, the program compares the details it is given in a short story to the fuller scenario laid out in the appropriate scripts, and calls on this knowledge (all accessed and deployed according to form-based syntactic matching procedures) to help answer questions that go beyond the specific details given in the story.

propriately will yield good reasoning about chess (or whatever), since many of the reason-respecting transitions are then visibly encoded in the system.

To get the flavor of the PSS hypothesis in action, consider first a program from Schank (1975). The goal of the program was story understanding: given a short text, it was meant to be able to answer some questions requiring a modicum of "common sense." To this end, Schank's program deployed so-called *scripts*, which used a symbolic event description language to encode background information about certain kinds of situations. For example, there was a script that laid out the typical sequence of actions involved in a visit to a restaurant (see Box 2.1). Now suppose you input a short story: "Jack goes into the restaurant, orders a hamburger, sits down. Later, he leaves after tipping the waiters." You can then ask: "Did Jack eat the hamburger?" and the computer, courtesy of the background information available in the script, can reply by guessing that he did.

Or consider SOAR (see Box 2.2). SOAR is a large-scale, on-going project that aims to apply the basic tenets of the PSS approach so as to implement general intelligence by computational means. It is, in many ways, the contemporary successor to the pioneering work on general problem solving (Newell, Shaw, and Simon, 1959) that helped set the agenda for the first three decades of work in artificial intelligence. SOAR is a symbol-processing architecture in which all long-term knowledge is stored using a uniform format known as a production memory. In a production memory, knowledge is encoded in the form of condition–action structures ("productions") whose contents are of the form: "If such and such is the case, then do so and so."[1] When it confronts a specific problem, SOAR accesses this general memory store until all relevant productions have been executed. This results in the transfer, into a temporary buffer or "working memory," of all the stuff that SOAR "knows" that looks like it might be relevant to the problem at hand. This body of knowledge will include a mixture of knowledge of facts, knowledge about actions that can be taken, and knowledge about what actions are desirable. A decision procedure then selects one action to perform on the basis of retrieved information concerning relative desirability ("preferences"). Naturally, SOAR is able to coordinate a sequence of such operations so as to achieve a specified goal. SOAR can work toward a distant goal by creating and attempting to resolve subgoals that reduce the distance between its current state and an overall solution. Such problem solving is conducted within so-called problem spaces populated by sets of states (representing situations) and operations (actions that can be applied to the states so as to yield further states). It is part of SOAR's job, given a goal, to select a problem space in which to pursue the goal, and to create a state that represents the ini-

[1]SOAR's productions differ from standard production-system structures insofar as SOAR incorporates a decision level (see text) that takes over some of the work traditionally done by the productions themselves. See Rosenbloom et al. (1992, pp. 294–295) for details.

Box 2.2

SOAR Points

The SOAR architecture has been used to solve problems in a wide variety of domains including computer configuration, algorithm design, medical diagnosis, and job-shop scheduling, as well as for less knowledge-intensive tasks such as playing tic-tac-toe. A simple demonstration, outlined in Rosenbloom et al. (1992, pp. 301–308), is the use of SOAR to do multicolumn subtraction. Here SOAR learns an effective procedure by searching in a subtraction problem space whose structure is provided in advance. The space contains the necessary "primitive acts" for a multicolumn "borrowing" procedure, in the form of operations such as the following:

> **Write-difference:** If the difference between the top digit and the bottom digit of the current column is known, then write the difference as an answer to the current column.
>
> **Borrow-into:** If the result of adding 10 to the top digit of the current column is known, and the digit to the left of it has a scratch mark on it, then replace the top digit with the result. (From Rosenbloom et al., 1992, p. 303, Figure 4)

The problem space contains a variety of such operators and a test procedure of the form "if each column has an answer, then succeed." SOAR then searches for a way to select and sequence these operations so as to succeed at the task. The search is constrained by productions associated with each operator that specify preferences concerning its use. SOAR is able to search the space of possible operator applications so as to discover a working procedure that makes use of the chunking maneuver to learn integrated, larger scale sequences that simplify future subtraction tasks.

A note in closing. SOAR, as a helpful referee reminds me, is at heart a universal programming system that can support pretty well *any* functional profile you like, so long as it is equipped with the right specialized sets of productions. The worries I raise in the text are thus not worries about what the bedrock programming system could *possibly* do, so much as worries about the particular configurations and strategies pursued in actual SOAR-based research (e.g., as exemplified in Rosenbloom et al., 1992). These configurations and strategies do indeed reflect the various practical commitments of the physical symbol system hypothesis as outlined earlier, and it is these commitments (rather than the bedrock programming system) that are critically examined in the text.

tial situation (the problem). An operator is then applied to that state, yielding a new state, and so on until (with luck) a solution is discovered. All these decisions (problem-space selection, state generation, operator selection) are based on the knowledge retrieved from the long-term production memory. In addition, the basic SOAR architecture exploits a single, uniform *learning mechanism*, known as "chunking," in which a successful sequence of subgoal generations can be stored away as a single unit. If SOAR later encounters a problem that looks similar to the earlier one, it can retrieve the unit and carry out the chunked sequence of moves without needing to search at each substage for the next move.

The actual practice of PSS-inspired artificial intelligence thus displays three key commitments. The first is the use of a symbolic code as a means of storing all of the system's long-term knowledge. The second is the depiction of intelligence as the ability to successfully search a symbolic problem-space. A physical symbol system "exercises its intelligence in problem-solving by search—that is, by generating and progressively modifying symbol structures until it reaches a solution structure" (Newell and Simon, 1976, p. 96). The third is that intelligence resides at, or close to, the level of deliberative thought. This is, if you like, the theoretical motivation for the development of semantically transparent systems—ones that directly encode and exploit the kinds of information that a human agent might consciously access when trying to solve a problem. Rosenbloom et al. (1992, pp. 290–291) thus describe SOAR as targeting the "cognitive band" in which contentful thoughts seem to flow in a serial sequence and in which most significant events occur in a time frame of 10 milliseconds to 10 seconds. This restriction effectively ensures that the computational story will at the same time function as a *knowledge-level*[2] story—a story that shows, rather directly, how knowledge and goals (beliefs and desires) can be encoded and processed in ways that lead to semantically sensible choices and actions. This is, of course, just the kind of story that Fodor (Chapter 1) insists we must provide so as to answer the question, "How is rationality mechanically possible?" (Fodor, 1986, p. 20).

So there it is. Intelligence resides at, or close to,[3] the level of deliberative thought. It consists in the retrieval of symbolically stored information and its use in processes of search. Such processes involve the generation, composition, and transformation of symbolic structures until the specified conditions for a solution are met. And it works, kind of. What could be wrong with that?

[2]For much more on the ideas of a "cognitive band" and a "knowledge level," see Newell (1990).

[3]The full story, as told in Newell (1990), recognizes four levels of cognitive activity as together constituting the "cognitive band." Only the topmost of these four levels (the "unit task" level) actually coincides with the consciously reportable steps of human problem solving. But all four levels involve operations on encoded knowledge, elementary choices, retrieval of distal information, and so on. In this respect, all four sublevels involve recognizably semantic or knowledge-involving operations.

2.2 Discussion

A. THE CHINESE ROOM

The most famous worry about symbol-crunching[4] artificial intelligence is predicated upon John Searle's (1980) "Chinese Room" thought experiment. Searle asks us to imagine a monolingual English speaker, placed in a large room, and confronted with a pile of papers covered with apparently unintelligible shapes and squiggles. The squiggles are, in fact, Chinese ideograms, but to the person in the room, they are just shapes on a page: just syntactic shells devoid of appreciable meaning. A new batch of squiggles then arrives, along with a set of instructions, in English, telling the person how to manipulate the apparently meaningless squiggles according to certain rules. The upshot of these manipulations, unbeknownst to the person in the room, is the creation of an intelligent response, in Chinese, to questions (also in Chinese) encoded in the incoming batch of papers.

The scenario, though strained and unlikely, cannot be ruled out. We saw, in Chapter 1, that any well-specified, intelligent behavior can be performed by a well-programmed computing device. What Searle has done is, in effect, to (1) replace the operating system and central processing unit of a computer (or the read-write head and finite state machine of a Turing machine) with a human agent and book of instructions, and (2) replace the real-world knowledge encoded in the computer's general memory (or the Turing machine's tape) with knowledge encoded (in Chinese) in the pile of papers. Under such circumstances, if the agent follows the rules, then (assuming, as we must, that the program is correct) the output will indeed be a sensible response in Chinese. The agent is "taking care of the syntax." And just as Haugeland (Chapter 1) said, the semantics is taking care of itself!

But says Searle, this is surely an illusion. It may seem like the overall system (the agent in the room) understands Chinese. But there is no real understanding at all. It seems to converse in Chinese, but no Chinese is actually understood! The monolingual agent is just doing syntactic matching. And the room and papers surely do not understand anything at all. Real understanding, Searle concludes, depends on more than just getting the formal operations right. Real understanding requires, Searle suggests, certain actual (though still largely unknown) physical properties, instantiated in biological brains. Stuff counts. Symbol manipulation alone is not enough.

Searle's argument has spawned a thousand attempts at rebuttal and refutation. A popular response is to insist that despite our intuitions, the room plus papers plus agent really does constitute a system that understands Chinese, has conscious experiences, and all the rest. And certainly, nothing that Searle (or anyone else)

[4]In fact, Searle (1992) extends his thought-experiment so as to (try to) cast doubt on connectionist approaches (see Chapter 4) also. Given my diagnosis (see the text) of the grain of truth in Searle's critique, this extension will not succeed. For a similar response, see Churchland and Churchland (1990).

says can rule that out as an empirical possibility. Appeals to intuition ("it doesn't *look* much like a system that really understands Chinese") are practically useless at the edges of scientific understanding.

It is also possible, however, that Searle is right, but for all the wrong reasons. For the Chinese room was initially envisioned as a weird and souped-up version of the story-understanding program mentioned earlier (see Box 2.1, and Schank and Abelson, 1977). As such, we were to imagine an inner computational economy in which semantically transparent symbols were being manipulated, in a stepwise, serial fashion, in ways specified by a further set of symbolic instructions. In short, we were to envision a fairly coarse-grained approach in which the system's stored knowledge, as encoded in the Chinese squiggles, might include general knowledge (about what happens when, for example, someone visits a restaurant) in a chunky, language-like format such as the following:

Script: Restaurant

Scene 1: ENTERING

PTRANS: go into restaurant
MBUILD: find table
PTRANS: go to table
MOVE: sit down

Extracted from Schank (1975, p. 131)

(Recall that symbols such as PTRANS form part of a special event description language devised by Schank, and are defined elsewhere in the program. PTRANS, for example, signifies the transfer of physical location of an object.)

Much of the *intuitive* appeal of Searle's argument, I believe, comes not from its logical structure but from a certain discomfort with the idea that a simulation *pitched at that kind of level* could actually amount to an instantiation of understanding, as opposed to a kind of superficial structural echo. Considered as a fully general logical argument, Searle's case is flimsy indeed. He aims to convince us that no amount of syntactic, formal organization can yield real understanding. But the only evidence [beyond the question-begging assertion that syntax is not sufficient for semantics—see, e.g., Churchland and Churchland (1990) for a nice discussion] is the reader's intuitive agreement, perhaps based on quite superficial features of the example.

Yet for all that the original thought experiment strikes a nerve. But the nerve is not (as Searle believes) the unbridgeability of the gap between syntax and semantics. It rather (concerns) the need for a finer grained specification of the relevant computational and syntactic structure. For it is plausible to suppose that if we seek to genuinely instantiate (not just roughly simulate) mental states in a computer, we will need to do more than just run a program that manipulates relatively high-level (semantically transparent) symbolic structures.

To begin to fix this idea (whose full expression must however wait until Chapter 4), we may introduce a contrast between functionalism and what I once termed (Clark, 1989) *microfunctionalism*. The functionalist, you will recall (Chapter 1), identifies being in a mental state with being in an abstract functional state, where a functional state is just some pattern of inputs, outputs, and internal state transitions taken to be characteristic of being in the mental state in question. But at what level of description should the functional story be told?

Consider a second famous thought experiment, this time due to Ned Block (1980, pp. 276–278) Block imagines that we somehow get the whole population of China to implement the functional profile of a given mental state by having them passing around letters or other formal symbols. But such an instantiation of the formal symbol-trading structure, Block fears, surely will not actually possess the target mental properties. At any rate, it will not be a thinking, feeling being in its own right. There will be no qualia, no raw feelings, no pains and pleasures for the country as a whole. The various individuals will have their own mental states, of course. But no new ones will come into being courtesy of the larger functional organization created by passing around slips of paper alone. From such considerations, Block concludes that functional identity cannot guarantee full-blooded (qualia-involving) mental identity. But once again, it all depends on our (unreliable) intuitions. Why shouldn't the Chinese room, or Block's Chinese population, actually have real, and qualitatively rich, mental states? Our discomfort, I suggest, flows not from the bedrock idea that the right formal structure could guarantee the presence of such states so much as from a nagging suspicion that the formal structures that will be implemented will prove too shallow, too much like the restaurant script structure rehearsed earlier. Now imagine instead a much finer grained formal description, a kind of "microfunctionalism" that fixes the fine detail of the internal state-transitions as, for example, a web of complex mathematical relations between simple processing units. Once we imagine such a finer grained formal specification, intuitions begin to shift. Perhaps once these microformal properties are in place, qualitative mental states will always emerge just as they do in real brains? It is somewhat harder to imagine just how these more microstructural features are to replicated by the manipulations of slips of paper, beer cans (another of Searle's favorites), or the population of China. But if these unlikely substrates *were* thus delicately organized, it does not strike me as crazy to suppose that real mental events might ensue. Or rather, it seems no *more* unlikely than the fact that they also ensue in a well-organized mush of tissue and synapses!

We will encounter, in Chapter 4, a somewhat different kind of computational model that pitches its descriptions of the formal structure of mind at just such a fine-grained level. These "connectionist" (or "neural network") approaches trade semantic transparency (the use of formal symbols to stand directly for familiar concepts, objects, events, and properties) against fineness of grain. They posit formal descriptions pitched at a level far distant from daily talk. They do not restrict their attention to the level of Newell's "cognitive band" or to operations that (in real

brains) take over 100 milliseconds to occur. They do, however, preserve the guiding vision of attending to the (micro)syntax and letting the semantics take care of itself.

B. EVERYDAY COPING

Here is a very different kind of criticism of the program of symbol-crunching A.I. Symbolic A.I., it has been suggested, is congenitally unable to come to grips with fast, fluent, everyday activity. It cannot do so because such activity is not, and could not be, supported by any set of symbolically coded rules, facts, or propositions. Instead, our everyday skills, which amount to a kind of expert engagement with the practical world, are said to depend on a foundation of "holistic similarity recognition" and bodily, lived experience. Such, in essence, is the criticism developed in a sequence of works by the philosopher Hubert Dreyfus (see, e.g., Dreyfus, 1972, 1992; Dreyfus and Dreyfus, 1986) and partially inspired by the ideas of Martin Heidegger (1927:1961).

Dreyfus' central concern is with the apparently bottomless richness of the understanding that we bring to our daily lives. Recall, for example, the simple restaurant script whose structure was displayed earlier. The point of such a script is to capture a stereotypical course of events (go into a restaurant, order food, eat it, leave tip) so as to provide some background knowledge for use in problem-solving behavior. But human minds seem able to respond sensibly to an apparently infinite set of potential variations on such a situation. What will the symbolic A.I. program do if it confronts a Martian in the kitchen, or a Harley-Davidson ridden into the restaurant?

Classical artificial intelligence has only two real responses to this problem of "depth of understanding." One is to add more and more (and more and more) knowledge in the form of explicitly coded information. [Doug Lenat's CYC project described in Lenat and Feigenbaum (1992) is an example of this strategy.] The other is to use powerful inference engines to press maximal effect from what the system already knows (the SOAR project discussed earlier displays something of this strategy). Both such strategies really amount to doing "more of the same," albeit with different emphases. Dreyfus' radical suggestion, by contrast, is that no amount of symbolically couched knowledge or inference can possibly reproduce the required "thickness" of understanding, since the thickness flows not from our knowledge of facts or our inferential capacities but from a kind of pattern-recognition ability honed by extensive bodily and real-world experience. The product of this experience is not a set of symbolic strings squirreled away in the brain but a kind of "knowing-how"—a knowing-how that cannot be reduced to any set, however extensive, of "knowing-thats" (see, e.g., Dreyfus, 1981, p. 198).

For example, we are asked to consider the contrast between the novice chess player (or car driver, or whatever) and the real expert. The novice, Dreyfus suggests, relies heavily on the conscious rehearsal of explicit symbol strings—rules like

"get your queen out early." The expert, by contrast, experiences "a compelling sense of the issue and the best move." Excellent chess players, we are told, can distinguish at a glance "roughly 50,000 types of position," and can, if necessary, choose moves at a speed that effectively precludes conscious analysis of the situation. The resultant flexibility of expert competence contrasts strongly with the oft-remarked "brittleness" of classical A.I. programs that rely on symbolically coded knowledge and make wild errors when faced with new or unexpected situations. Expert know-how, Dreyfus and Dreyfus (1986, p. 28) suggest, may be more fruitfully modeled using the alternative, pattern-recognition-based technologies (see Chapter 4) of connectionism and artificial neural networks. Since such expertise pervades the bulk of our daily lives (we are all, or most of us, "experts" at making tea and coffee, avoiding traffic accidents, engaging in social interactions, cooking dinner, making sandwiches, riding bicycles, and so on), the criticism that such activity lies outside the scope of symbolic A.I. is damning indeed. Is Dreyfus right? It is hard to fault the observation that symbolic A.I. seems to yield limited and brittle systems whose common sense understanding leaves plenty to be desired. In exactly this vein, for example, a skeptical computer scientist, commenting on the SOAR project, once offered the following "friendly challenge":

> Give us "Agent-Soar" [a system capable of] operating continuously, selectively perceiving a complex unpredictable environment, noticing situations of interest. Show us how it integrates concurrent tasks and coordinates their interacting needs . . . show us how it modifies its knowledge based on experience and makes the best use of dynamic but limited resources under real-time constraints. (Hayes-Roth, 1994, p. 96)

It is only fair to note, however, that much the same challenge could be raised regarding the connectionist research program presented in Chapter 4. My own view, then, is that the "argument from fluent everyday coping" actually points to much that is wrong with *both* connectionist *and* symbol-processing artificial intelligence. This point is not lost on Dreyfus and Dreyfus (1990) who note that human beings may be even more "holistic" than neural nets, and wonder whether we need to consider a larger "unit of analysis" comprising brain, body, and cultural environment [a "whole organism geared into a whole cultural world" (p. 331)]. Such issues will return to haunt us in the closing chapters. For now, we may simply conclude that everyday coping poses extremely difficult problems for any staunchly symbolic approach and that any move away from reliance on explicit, coarse-grained symbol structures and toward fast, flexible pattern-recognition-based models is probably a step in the right direction.

C. REAL BRAINS AND THE BAG OF TRICKS

One of the guiding assumptions of classical symbol-crunching A.I. is, we saw, that the scientific study of mind and cognition may proceed without essential reference to matters of implementation. This assumption, clearly displayed in, e.g., the SOAR

team's decision to focus purely on the "cognitive band," is open to serious doubt. The situation is nicely summed-up by the cognitive scientist Donald Norman:

> Soar . . . espouses the software-independence approach to modeling. That is, psychological functions are assumed to be independent of hardware implementation, so it is safe to study the cognitive band without examination of the implementation methods of the neural band, without consideration of the physical body in which the organism is embedded, and without consideration of non-cognitive aspects of behavior. (Norman, 1992, p. 343)

The worries concerning the potential roles of the physical body (and the wider environment) will occupy us in later chapters. An immediate question, however, concerns the attempt to model psychological functions without reference to the details of neural implementation.

On the positive side, we can say this: it is probably true that at least some psychological states will be *multiply realizable.* That is to say, several different hardware and software organizations will be capable of supporting the same mental states. The point about multiple *hardware* realizability flows directly from the bedrock idea of mind as a formal system, and the consequent focus on structure not stuff. The point about multiple *software* realizability is trickier (and is further pursued in the next chapter). But there exist, for example, a variety of different procedures for sorting a set of numbers or letters into sequence (Quick-sort and BUBBLE-sort to name but two). Is it not similarly unlikely that there is just one algorithmic structure capable of supporting, e.g., the mental state of believing it is raining?

On the negative side, however, it is *equally* unlikely that we will discover a good model of the formal structure of human thought if we proceed in a neurophysiological vacuum. Consider, for example, the SOAR team's commitment to a single type of long-term memory (but see Box 2.2 for an important caveat). SOAR thus used relies on a uniform production memory to store all its long-term knowledge. Is this assumption legitimate? Donald Norman (among others) argues that it is not, since human memory seems to involve multiple psychologically and neurophysiologically distinct systems.[5] For example, the distinction between semantic memory (memory for facts, such as "dogs have four legs") and episodic memory (memory of specific experiences and events, such as the day the dog buried the tortoise). SOAR can, it is true, reproduce much of the surface behavior associated with each memory type (see Newell, 1990, Chapter 6). But this surface mimicry, as Norman points out, does little to counter the growing body of neuropsychological evidence in favor of the psychological realism of multiple memory systems. Much of the relevant evidence comes not from normal, daily behavior but from

[5]See, e.g., Tulving (1983). The debate over multiple memory types continues today. But for our purposes, it does not really matter what the final story is. The example serves merely to illustrate the potential for conflict between specific uses of SOAR and neuropsychological data.

studies of brain damage and brain abnormalities, for example, studies of amnesiacs whose episodic memory is much more severely impaired than their semantic memory.[6] There is also some neuroimaging work (using scanning techniques to plot blood flow in the brain) that suggests that different neural areas are active in different kinds of memory tasks. Such studies all combine to suggest real and psychologically significant differences between various memory systems.

The point about multiple memory systems may be carried a step further by considering the more general idea of multiple cognitive systems. Recent work in so-called evolutionary psychology (see, e.g., Tooby and Cosmides, 1992) challenges the ideas of uniformity and simplicity stressed by Rosenbloom et al. (1992, p. 293) and enshrined in their particular configuration of SOAR. Instead of a uniform learning procedure, single long-term memory, and a small set of inference engines, the evolutionary psychologists depict the mind as a kind of grab-bag of quite specialized knowledge-and-action stores, developed in a piecemeal fashion (over evolutionary time) to serve specific, adaptively important ends. They thus liken the mind to a Swiss army knife—a collection of surprisingly various specialized implements housed in a single shell. Such cognitive implements (sometimes called "modules") might include one for thinking about spatial relations, one for tool use, one for social understanding, and so on (see, e.g., the list in Tooby and Cosmides, 1992, p. 113). Evolutionary psychology presents a radical and as yet not fully worked-out vision. [For a balanced assessment see Mitchell (1999).] But the general image of human cognition as to some degree a "bag of tricks" rather than a neat, integrated system is winning support from a variety of quarters. It is gaining ground in work in real-world robotics, since special-purpose tricks are often the only way to generate adaptive behavior in real time (see Chapter 6). And it is gaining ground in some neuroscientific and neuropsychological research programs.[7] In a great many quarters, the idea that intelligent activity is mediated by the sequential, serial retrieval of symbol structures from some functionally homogeneous inner store is being abandoned in favor of a more neurologically realistic vision of multiple representational types and processes, operating in parallel and communicating in a wide range of different ways. Notice, then, the extreme distance that separates this image of cognition from the idea (Newell, 1990, p. 50) of a single sequence of cognitive actions drawing on a unified knowledge store. Serial retrieval of items from a homogeneous knowledge store may work well as a model of a few isolated fragments of human behavior (such as doing a crossword). But, to quote Marvin Minsky:

> Imagine yourself sipping a drink at a party while moving about and talking with friends. How many streams of processing are involved in shaping your hand to keep the cup

[6]Squire and Zola-Morgan (1988) and Tulving (1989).

[7]See, e.g., Churchland, Ramachandran, and Sejnowski (1994). See also Ballard (1991). This work is discussed at length in Clark (1997).

> level, while choosing where to place your feet? How many processes help choose the words that say what you mean while arranging those words into suitable strings . . . what about those other thoughts that clearly go on in parallel as one part of your mind keeps humming a tune while another sub-mind plans a path that escapes from this person and approaches that one. (Minsky, 1994, p. 101)

Minsky's alternative vision depicts mind as an assortment of subagencies, some of which deploy special-purpose routines and knowledge stores. The neuroscientist Michael Arbib offers a related vision of neural computation as essentially distributed with different brain regions supporting different kinds of "partial representations." Cognitive effects, Arbib suggests, arise from the complex interactions of a multitude of such concurrently active partial representations. The point, he says, is that "no single, central, logical representation of the world need link perception and action—the representation of the world is *the pattern of relationships between all its partial representations*" (Arbib, 1994, p. 29, original emphasis).

We should not, of course, mistake every criticism of a particular use of SOAR[8] for a criticism of classical, symbol-crunching A.I. per se. Perhaps one day there will be symbol-processing systems (perhaps even a version of SOAR—see Box 2.2) that take much more account of the parallel, distributed, fragmentary nature of real neural processing. Certainly there is nothing in the bedrock ideas of classical A.I. (see Chapter 1) that rules out either the use of parallel processing or of multiple, special-purpose tricks and strategies. There are even up-and-running programs that prove the point. What seems most at stake is the once-standard image of the actual nature of the symbol structures involved. For the contents of such multiple, "partial" representations are unlikely to be semantically transparent in the sense described earlier; they are unlikely to admit of easy interpretation in terms of our high-level understanding of some problem domain. Instead, we must attend to a panoply of harder to interpret, "partial," perhaps "subsymbolic" (see Chapter 4) states whose cumulative effect is to sculpt behavior in ways that respect the space of reasons and semantic sense. The spirit of this enterprise, it seems to me, is genuinely distinct from that of symbol system A.I. Instead of going straight for the jugular and directly recapitulating the space of thought and reasons using logical operations and a language-like inner code, the goal is to coax semantically sensible behavior from a seething mass of hard-to-manage parallel interactions between semantically opaque inner elements and resources.

2.3 Suggested Readings

On *classical A.I. and the physical symbol system hypothesis*, see A. Newell and H. Simon, "Computer science as empirical inquiry: Symbols and search." In J. Haugeland (ed.), *Mind Design II* (Cambridge, MA: MIT Press, 1997, pp. 81–110). (Nice original account of the Physical Symbol System Hypothesis from two of the early stars of classical artificial intelli-

[8]For replies to some of these criticisms, see Rosenbloom and Laird (1993)

gence.) For *the classical A.I. endeavor in modern dress*, see P. Rosenbloom, J. Laird, A. Newell, and R. McCarl, "A preliminary analysis of the SOAR architecture as a basis for general intelligence." In D. Kirsh (ed.), *Foundations of Artificial Intelligence* (Cambridge, MA: MIT Press, 1992, pp. 289–325).

For *important critiques of classical A.I.*, see J. Searle, "Minds, brains and programs." In J. Haugeland (ed.), *Mind Design II* (Cambridge, MA: MIT Press, 1997, pp. 183-204). (Crisp, provocative critique of classical AI using the infamous Chinese Room thought experiment.) H. Dreyfus, "Introduction" to his *What Computers Still Can't Do* (Cambridge, MA: MIT Press, 1992). (The "everyday coping" objections, and some intriguing comments on the connectionist alternative to classical A.I.) D. Dennett, "Cognitive wheels: The frame problem of AI." In M. Boden (ed.), *The Philosophy of Artificial Intelligence* (Oxford, England: Oxford University Press, 1990, pp. 147–170). (Another take on the problem of formalizing common-sense reasoning, written with Dennett's customary verve and dash.)

For *some recent retrospectives on classical A.I., its attractions and pitfalls*, see S. Franklin, *Artificial Minds* (Cambridge, MA: MIT Press, 1995, Chapters 4 and 5), and the various perspectives represented in the 11 reviews collected in Section 1 ("Symbolic models of mind") of W. Clancey, S. Smoliar, and M. Stefik (eds.), *Contemplating Minds* (Cambridge, MA: MIT Press, 1994, pp. 1–166). A *useful collection* is J. Haugeland's *Mind Design II* (Cambridge, MA: MIT Press, 1997), especially (in addition to the pieces by Searle and by Newell and Simon cited above) the introduction "What is mind design?" by J. Haugeland and the papers by Minsky ("A framework for representing knowledge") and Dreyfus ("From micro-worlds to knowledge representation: A.I. at an Impasse").

4 CONNECTIONISM

4.1 Sketches
4.2 Discussion
A. Connectionism and Mental Causation
B. Systematicity
C. Biological Reality?
4.3 Suggested Readings

4.1 Sketches

The computational view of mind currently comes in two basic varieties. The basic physical symbol system variety, already encountered in Chapter 2, stresses the role of symbolic atoms, (usually) serial processing, and expressive resources whose combinational forms closely parallel those of language and logic. The other main variety differs along all three of these dimensions and is known variously as connectionism, parallel distributed processing, and artificial neural networks.

These latter models, as the last name suggests, bear some (admittedly rather distant) relation to the architecture and workings of the biological brain. Like the brain, an artificial neural network is composed of many simple processors linked in parallel by a daunting mass of wiring and connectivity. In the brain, the "simple processors" are neurons (note the quotes: neurons are much more complex than connectionist units) and the connections are axons and synapses. In connectionist networks, the simple processors are called "units" and the connections consists in numerically weighted links between these units—links known, unimaginatively but with pinpoint accuracy, as connections. In both cases, the simple processing elements (neurons, units) are generally sensitive only to local influences. Each element takes inputs from a small group of "neighbors" and passes outputs to a small (sometimes overlapping) group of neighbors.

The differences between simple connectionist models and real neural architectures remain immense and we will review some of them later in this chapter. Nonetheless, something of a common flavor does prevail. The essence of the common flavor lies mostly in the use of large-scale parallelism combined with local computation, and in the (related) use of a means of coding known as distributed

representation. To illustrate these ideas, consider the now-classic example of NETtalk.

NETtalk (Sejnowski and Rosenberg, 1986, 1987) is an artificial neural network, created in the mid-1980s, whose task was to take written input and turn it into coding for speech, i.e., to do grapheme-to-phoneme conversion. A successful classical program, called DECtalk, was already in existence and performed the same task courtesy of a large database of rules and exceptions, hand coded by a team of human programmers. NETtalk, by contrast, instead of being explicitly programmed, *learned* to solve the problem using a learning algorithm and a substantial corpus of example cases—actual instances of good text-to-phoneme pairings. The output of the network was then fed to a fairly standard speech synthesizer that took the phonetic coding and transformed it into real speech. During learning, the speech output could be heard to progress from initial babble to semirecognizable words and syllable structure, to (ultimately) a fair simulacrum of human speech. The network, it should be emphasized, was not intended as a model of language understanding but only of the text-to-speech transition—as such, there was no semantic database tied to the linguistic structures. Despite this lack of semantic depth, the network stands as an impressive demonstration of the power of the connectionist approach. Here, in briefest outline, is how it worked.

The system, as mentioned above, is comprised of a set of simple processing units. Each unit receives inputs from its neighbors (or from the world, in the case of so-called input units) and yields an output according to a simple mathematical function. Such functions are often nonlinear. This means that the numerical value of the output is not directly proportional to the sum of the inputs. It may be, for example, that a unit gives a proportional output for an intermediate range of total input values, but gives a constant output above and below that range, or that the unit will not "fire" until the inputs sum to a certain value and thereafter will give proportional outputs. The point, in any case, is that a unit becomes activated to whatever degree (if any) the inputs from its local neighbors dictate, and that it will pass on a signal accordingly. If unit *A* sends a signal to unit *B*, the strength of the signal arriving at *B* is a joint function of the level of activation of the "sender" unit and the numerical weighting assigned to the connection linking *A* to *B*. Such weights can be positive (excitatory) or negative (inhibitory). The signals arriving at the receiving units may thus vary, being determined by the product of the numerical weight on a specific connection and the output of the "sender" unit.

NETtalk (see Box 4.1) was a fairly large network, involving seven groups of input units, each group comprising some 29 individual units whose overall activation specified one letter. The total input to the system at each time step thus specified seven distinct letters, one of which (the fourth) was the target letter whose phonemic contribution was to be determined and given as output. The other six letters provided essential contextual information, since the phonemic impact of a

Box 4.1

THE NETTALK ARCHITECTURE

The specific architecture of NETtalk (see Figure 4.1) involved three layers of units (a typical "first-generation" layout, but by no means obligatory). The first layer comprised a set of "input" units, whose task was to encode the data to be processed (information about letter sequences). The second layer consisted of a group of so-called hidden units whose job was to effect a partial recoding of the input data. The third layer consisted of "output" units whose activation patterns determine the system's overall response to the original input. This response is specified as a vector of numerical activation values, one value for each output unit. The knowledge that the system uses to guide the input–output transitions is thus encoded to a large extent in the weights on the various interunit connections. An important feature of the connectionist approach lies in the use of a variety of potent (though by no mean omnipotent!) learning algorithms. These algorithms (see text and Box 4.2) adjust the weights on the interunit connections so as to gradually bring the overall performance into line with the target input–output function implicit in a body of training cases.

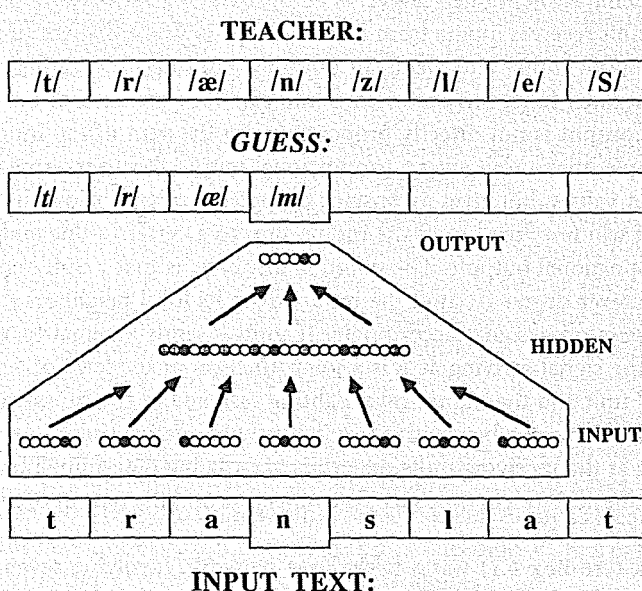

Figure 4.1 Schematic of NETtalk architecture showing only some units and connectivity. Each group of 29 input units represents a letter. The 7 groups of input units were transformed by 80 hidden units. These hidden units then projected to 26 output units, which represented 54 phonemes. There were a total of 18,629 weights in the network. (From Sejnowski and Rosenberg, 1987, by permission.)

Box 4.2

Gradient Descent Learning

The learning routine involves what is known as gradient descent (or hill climbing, since the image can be systematically inverted!). Imagine you are standing somewhere on the inner slopes of a giant pudding basin. Your task is to find the bottom—the point corresponding to the lowest error and hence the best available solution. But you are blindfolded and cannot see the bottom and cannot run directly to it. Instead, you take a single step and determine whether you went up or down. If you went up (a local error), you go back and try again in the opposite direction. If you went down, you stay where you are. By repeating this procedure of small steps and local feedback, you slowly snake toward the bottom and there you halt (since no further step can take you any lower). The local feedback, in the case of the neural network, is provided by the supervisory system that determines whether a slight increase or decrease in a given weight would improve performance (assuming the other weights remain fixed). This procedure, repeated weight by weight and layer by layer, effectively pushes the system down a slope of decreasing error. If the landscape is a nice pudding-basin shape with no nasty trenches or gorges, the point at which no further change can yield a lower error signal will correspond to a good solution to the problem.

given letter (in English) varies widely accordingly to the surrounding letters. The input units were connected to a layer of 80 hidden units, and these connected in turn to a set of 26 output units coding for phonemes. The total number of interunit links in the overall network summed to 18,829 weighted connections.

Given this large number of connections, it would be impractical (to say the least) to set about finding appropriate interunit connection weights by hand coding and trial and error! Fortunately, automatic procedures (learning algorithms) exist for tuning the weights. The most famous (but probably biologically least realistic) such procedure is the so-called back-propagation learning algorithm. In back-propagation learning, the network begins with a set of randomly selected connection weights (the layout, number of units, etc. being fixed by the designer). This network is then exposed to a large number of input patterns. For each input pattern, some (initially incorrect) output is produced. An automatic supervisory system monitors the output, compares it to the target (correct) output, and calculates small adjustments to the connection weights—adjustments that would cause slightly improved performance were the network to be reexposed to the very same input pattern. This procedure (see Box 4.2) is repeated again and again for a large

(and cycling) corpus of training cases. After sufficient such training, the network often (though not always) learns an assignment of weights that effectively solves the problem—one that reduces the error signal and yields the desired input–output profile.

Such learning algorithms can discover solutions that we had not imagined. Researcher bias is thus somewhat decreased. Moreover, and perhaps more importantly, the way the trained-up network *encodes* the problem-solving information is quite unlike the more traditional forms of symbol-string encoding characteristic of the work discussed in Chapter 2. The connectionist system's long-term knowledge base does not consist in a body of declarative statements written out in a formal notation based not on the structure of language or logic. Instead, the knowledge inheres in the set of connection weights and the unit architecture. Many of these weighted connections participate in a large number of the system's problem-solving activities. Occurrent knowledge—the information active during the processing of a specific input—may usefully be equated with the transient activation patterns occurring in the hidden unit layer. Such patterns often involve *distributed* and *superpositional* coding schemes. These are powerful features, so let's pause to unpack the jargon.

An item of information is here said to have a distributed representation if it is expressed by the simultaneous activity of a number of units. But what makes distributed representation computationally potent is not this simple fact alone, but the systematic use of the distributions to encode further information concerning subtle similarities and differences. A distributed pattern of activity can encode "microstructural" information such that variations in the overall pattern reflect variations in the content. For example, a certain pattern might represent the presence of a black cat in the visual field, whereas small variations in the pattern may carry information about the cat's orientation (facing ahead, side-on, etc.). Similarly, the activation pattern for a black panther may share some of the substructure of the cat activation pattern, whereas that for a white fox may share none. The notion of superpositional storage is precisely the notion of such partially overlapping use of distributed resources, in which the overlap is informationally significant in the kinds of way just outlined. The upshot is that semantically related items are represented by syntactically related (partially overlapping) patterns of activation. The public language words "cat" and "panther" display no such overlap (though *phrases* such as "black panther" and "black cat" do). Distributed superpositional coding may thus be thought of as a trick for forcing still more information into a system of encodings by exploiting even more highly structured syntactic vehicles than words. This trick yields a number of additional benefits, including economical use of representational resources, "free" generalization, and graceful degradation. Generalization occurs because a new input pattern, if it resembles an old one in some aspects, will yield a response rooted in that partial overlap. "Sensible" responses to new inputs are thus possible. "Graceful degradation," alluring as it sounds, is just

the ability to produce sensible responses given some systemic damage. This is possible because the overall system now acts as a kind of pattern completer—given a large enough fragment of a familiar pattern, it will recall the whole thing. Generalization, pattern completion, and damage tolerance are thus all reflections of the same powerful computational strategy: the use of distributed, superpositional storage schemes and partial cue-based recall.

Two further properties of such coding schemes demand our attention. The first is the capacity to develop and exploit what Paul Smolensky (1988) has termed "dimension shifted" representations. The second is the capacity to display fine-grained context sensitivity. Both properties are implied by the popular but opaque gloss on connectionism that depicts it as a "subsymbolic paradigm." The essential idea is that whereas basic physical symbol system approaches displayed a kind of semantic transparency (see Chapter 2) such that familiar words and ideas were rendered as simple inner symbols, connectionist approaches introduced a much greater distance between daily talk and the contents manipulated by the computational system. By describing connectionist representation schemes as dimension shifted and subsymbolic, Smolensky (and others) means to suggest that the features that the system uncovers are finer grained and more subtle than those picked out by single words in public language. The claim is that the contentful elements in a subsymbolic program do not directly recapitulate the concepts we use "to consciously conceptualize the task domain" (Smolensky, 1988, p. 5) and that "the units do not have the same semantics as words of natural language" (p. 6). The activation of a given unit (in a given context) thus signals a semantic fact: but it may be a fact that defies easy description using the words and phrases of daily language. The semantic structure represented by a large pattern of unit activity may be very rich and subtle indeed, and minor differences in such patterns may mark equally subtle differences in contextual nuance. Unit-level activation differences may, thus, reflect minute details of the visual tactile, functional, or even emotive dimensions of our responses to the same stimuli in varying real-world contexts. The pioneer connectionists McClelland and Kawamoto (1986) once described this capacity to represent "a huge palette of shades of meaning" as being "perhaps . . . the paramount reason why the distributed approach appeals to us" (p. 314).

This capacity to discover and exploit rich, subtle, and nonobvious schemes of distributed representation raises an immediate methodological difficulty: how to achieve, after training, some understanding of the knowledge and strategies that the network is actually using to drive its behavior? One clue, obviously, lies in the training data. But networks do not simply learn to repeat the training corpus. Instead they learn (as we saw) general strategies that enable them to group the training instances into property-sharing sets, to generalize to new and unseen cases, etc. Some kind of knowledge organization is thus at work. Yet it is impossible (for a net of any size or complexity) to simply read this organization off by, e.g., inspecting a trace of all the connection weights. All you see is numerical spaghetti!

The solution to this problem of "posttraining analysis" lies in the use of a variety of tools and techniques including statistical analysis and systematic interference. Systematic interference involves the deliberate damaging or destruction of groups of units, sets of weights, or interunit connections. Observation of the network's "postlesion" behavior can then provide useful clues to its normal operating strategies. It can also provide a further dimension (in addition to brute performance) along which to assess the "psychological reality" of a model, by comparing the way the network reacts to damage to the behavior patterns seen in humans suffering from various forms of local brain damage and abnormality (see, e.g., Patterson, Seidenberg, and McClelland, 1989; Hinton and Shallice, 1989). In practice, however, the most revealing forms of posttraining analysis have involved not artificial lesion studies but the use of statistical tools (see Box 4.3) to generate a picture of the way the network has learned to negotiate the problem space.

So far, then, we have concentrated our attention or what might be termed "first-generation" connectionist networks. It would be misleading to conclude, however, without offering at least a rough sketch of the shape of more recent developments.

Second-generation connectionism is marked by an increasing emphasis on temporal structure. First-generation networks, it is fair to say, displayed no real capacity to deal with time or order. Inputs that designated an ordered sequence of letters had to be rendered using special coding schemes that artificially disambiguated the various possible orderings. Nor were such networks geared to the production of output patterns extended in time (e.g., the sequence of commands needed to produce a running motion)[1] or to the recognition of temporally extended patterns such as the sequences of facial motion that distinguish a wry smile from a grimace. Instead, the networks displayed a kind of "snapshot reasoning" in which a frozen temporal instant (e.g., coding for a picture of a smiling person) yields a single output response (e.g., a judgment that the person is happy). Such networks could not identify an instance of pleasant surprise by perceiving the gradual transformation of puzzlement into pleasure (see e.g., Churchland, 1995).

To deal with such temporally extended data and phenomena, second-generation connectionist researchers have deployed so-called recurrent neural networks. These networks share much of the structure of a simple three-layer "snapshot" network, but incorporate an additional feedback loop. This loop (see Figure 4.3) recycles some aspect of the networks activity at time t_1 alongside the new inputs arriving at time t_2. Elman nets (see Elman, 1991b) recycle the hidden unit activation pattern from the previous time slice, whereas Jordan (1986) describes a net that recycles its previous output pattern. Either way, what is preserved is some kind of

[1]These issues are usefully discussed in Churchland and Sejnowski (1992, pp. 119–120). For a more radical discussion, see Port, Cummins, and McCauley (1995).

Box 4.3

Cluster Analysis

Cluster analysis is an example of an analytic technique addressing the crucial question "what kinds of representation has the network acquired?" A typical three-layer network, such as NETtalk, uses the hidden unit layer to partition the inputs so as to compress and dilate the input representation space in ways suited to the particular target function implied by the training data. Thus, in text-to-phoneme conversion, we want the rather different written inputs "sale" and "sail" to yield the same phonetic output. The hidden units should thus compress these two input patterns into some common intermediate form. Inputs such as "shape" and "sail" should receive different, though not unrelated, codings, whereas "pint" and "hint," despite substantial written overlap, involve widely variant phonemic response and should be dilated—pulled further apart. To perform these tricks of pulling together and pushing apart, NETtalk developed 79 different patterns of hidden unit activity. Cluster analysis then involved taking each such pattern and matching it with its nearest neighbor (e.g., the four-unit activation pattern 1010 is nearer to 1110 than to 0101, since the second differs from the first in only one place whereas the third differs in four). The most closely matched pairs are then rendered (by a process of vector averaging) as new single patterns and the comparison process is repeated. The procedure continues until the final two clusters are generated, representing the grossest division of the hidden unit space learned by the system. The result is an unlabeled, hierarchical tree of hidden unit activity. The next task is to label the nodes. This is done as follows. For each of the original 79 activation patterns, the analyst retains a trace of the input pattern that prompted that specific response. She then looks at the pairs (or pairs of pairs, etc.) of inputs that the network "chose" to associate with these similar hidden unit activation patterns so as to discern what those inputs had in common that made it *useful* for the network to group them together. The result, in the case of NETtalk, is a branching hierarchical tree (see Figure 4.2) whose grossest division is into the familiar groupings of vowels and consonants and whose subdivisions include groupings of different ways of sounding certain input letters such as *i*, *o* etc. In fact, nearly all the phonetic groupings learned by NETtalk turned out to correspond closely to divisions in existing phonetic theory. One further feature discussed in Section 4.2 is that various versions of NETtalk (maintaining the same architecture and learning routine and training data but beginning with different assignments of random weights) exhibited, after training,

very different sets of interunit weightings. Yet these superficially different solutions yield almost identical cluster-analytic profiles. Such nets use different numerical schemes to encode what is essentially the *same* solution to the text-to-phoneme conversion problem.

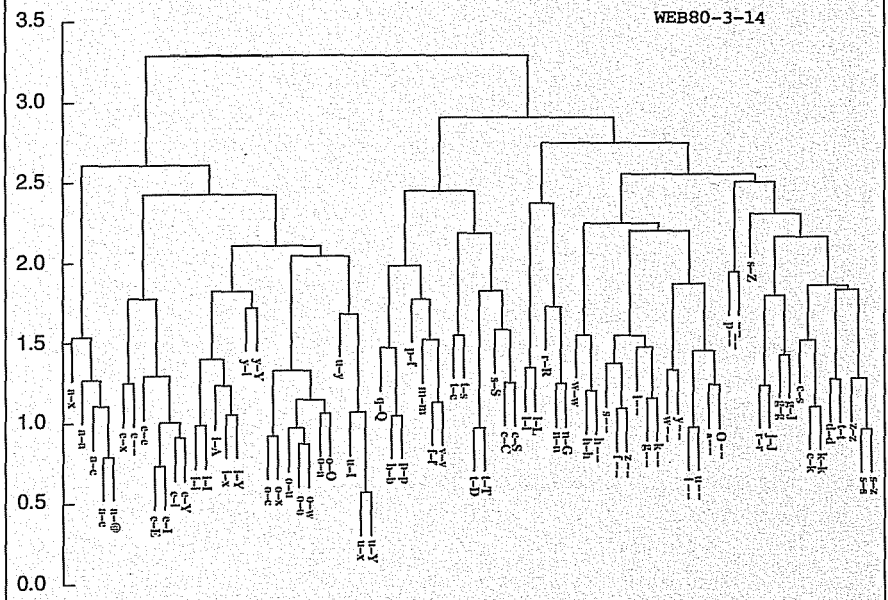

Figure 4.2 Hierarchical cluster analysis of the average activity levels on the hidden units for each letter-to-sound correspondence (*l–p* for letter *l* and phoneme *p*). The closest branches correspond to the most nearly similar activation vectors of the hidden units. (From Sejnowski and Rosenberg, 1987. Parallel networks that learn to pronounce English text. *Complex Systems*, 1, 145–168. Reproduced by kind permission of T. Sejnowski.)

on-going trace of the network's last activity. Such traces act as a kind of short-term memory enabling the network to generate new responses that depend both on the current input and on the previous activity of the network. Such a set-up also allows output activity to continue in the complete absence of new inputs, since the network can continue to recycle its previous states and respond to them.

For example, Elman (1991b) describes a simple recurrent network whose goal is to categorize words according to lexical role (noun, verb, etc.). The network was exposed to grammatically proper sequences of words (such as "the boy broke the window"). Its immediate task was to predict the next word in the on-going sequence. Such a task, it should be clear, has no unique solution insofar as many continuations will be perfectly acceptable grammatically. Nonetheless, there are

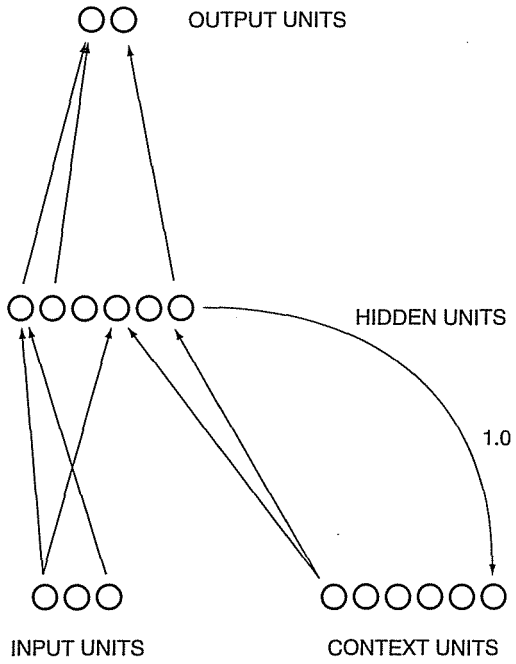

Figure 4.3 A three-layer recurrent network. The context units are activated, one by one, by the corresponding hidden units. For simplicity, not all the activation is shown. (After Elman, 1991b, with permission.)

whole classes of words that cannot be allowed to follow. For example, the input sequence "the boy who" cannot be followed by "cat" or "tree." These constraints on acceptable successor words reflect grammatical role and the training regime thus provides data germane to the larger goal of learning about lexical categories.

Elman's network proved fairly adept at its task. It "discovered" categories such as verb and noun and also evolved groupings for animate and inanimate objects, foods, and breakable objects—properties that were good clues to grammatical role in the training corpus used. To determine exactly what the network learned, Elman used another kind of posttraining analysis (one better suited to the special case of recurrent nets) called "principal component analysis" (PCA). The details are given in Clark (1993, pp. 60–67) and need not detain us here. It is worth noting, however, that whereas cluster analysis can make it seem as if a network has merely learned a set of static distributed symbols and is thus little more than a novel implementation of the classical approach, principal component analysis reveals the role of even deeper dynamics in promoting successful behavior. The key idea is that whereas cluster analysis stresses relations of similarity and difference between static states ("snapshots"), PCA reflects in addition the ways in which being in one state (in a recurrent network) can promote or impede movement into

future states. Standard cluster analysis would not reveal these constraints on processing trajectories. Yet the grammatical knowledge acquired by the recurrent net inheres quite profoundly in such temporally rich information-processing detail.[2]

The more such temporal dynamics matter, the further we move (I contend) from the guiding image of the basic physical symbol system hypothesis. For at the heart of that image lies the notion of essentially static symbol structures that retain stable meanings while being manipulated by some kind of central processor. Such a picture, however, does not usefully describe the operation of even the simple recurrent networks previously discussed. For the hidden unit activation patterns (the nearest analogue to static symbols) do not function as fixed representations of word-role. This is because each such pattern reflects something of the prior context,[3] so that, in a sense, "every occurrence of a lexical item has a separate internal representation" (Elman, 1991b, p. 353). Elman's model thus uses so-called dynamic representations. Unlike the classical image in which the linguistic agent, on hearing a word, retrieves a kind of general-purpose lexical representation, Elman is suggesting a dynamic picture in which

> There is no separate stage of lexical retrieval. There are no representations of words in isolation. The representations of words (the internal states following input of a word) always reflect the input taken together with the prior state . . . the representations are not propositional and their information content changes constantly over time . . . words serve as guideposts which help establish mental states that support (desired) behavior. (Elman, 1991b, p. 378)

Elman thus invites us to see beyond the classical image of static symbols that persist as stored syntactic items and that are "retrieved" and "manipulated" during processing. Instead, we confront an image of a fluid inner economy in which representations are constructed on the spot and in light of the prevailing context and in which much of the information-processing power resides in the way current states constrain the future temporal unfolding of the system.

Third-generation connectionism continues this flight from the (static) inner symbol by laying even greater stress on a much wider range of dynamic and time-involving properties. For this reason it is sometimes known as "dynamical connectionism." Dynamical connectionism (see Wheeler, 1994, p. 38; Port and van Gelder, 1995, pp. 32–34) introduces a number of new and more neurobiologically realistic features to the basic units and weights paradigm, including special purpose units (units whose activation function is tailored to a task or domain), more complex connectivity (multiple recurrent pathways and special purpose wiring), computationally salient time delays in the processing cycles, continuous-time processing, analog signaling, and the deliberate use of noise. Artificial neural networks

[2]See Elman (1991b, p. 106).

[3]Even the first word in a sentence incorporates a kind of "null" context that is reflected in the network state.

exhibiting such nonstandard features support "far richer intrinsic dynamics than those produced by mainstream connectionist systems" (Wheeler, 1994, p. 38). We shall have more to say about the potential role of such richer and temporally loaded dynamics in future chapters. For the moment, it will suffice to note that second- and third-generation connectionist research is becoming progressively more and more dynamic: it is paying more heed to the temporal dimension and it is exploiting a wider variety of types of units and connectivity. In so doing, it is moving ever further from the old notion of intelligence as the manipulation of static, atemporal, spatially localizable inner symbols.

The connectionist movement, it is fair to conclude, is the leading expression of "inner symbol flight." The static, chunky, user-friendly, semantically transparent (see Chapter 2) inner symbols of yore are being replaced by subtler, often highly distributed and increasingly dynamic (time-involving) inner states. This is, I believe, a basically laudable transition. Connectionist models profit from (increasing) contact with real neuroscientific theorizing. And they exhibit a profile of strengths (motor control, pattern recognition) and weaknesses (planning and sequential logical derivation) that seems reassuringly familiar and evolutionarily plausible. They look to avoid, in large measure, the uncomfortable back-projection of our experiences with text and words onto the more basic biological canvass of the brain. But the new landscape brings new challenges, problems, and uncertainties. Time to meet the bugbears.