

Parallel Computation and the Mind-Body Problem

PAUL THAGARD

*University of Michigan-Dearborn
Princeton University*

The position in the philosophy of mind called *functionalism* claims that mental states are to be understood in terms of their functional relationships to other mental states, not in terms of their material instantiation in any particular kind of hardware. But the argument that material instantiation is irrelevant to functional relationships is computationally naive. This paper uses recent work on parallel computation to argue that software and hardware are much more intertwined than the functionalists allow. Parallelism offers qualitative as well as quantitative advantages, leading to different styles of programming as well as increased speed. Hence hardware may well matter to the mental: only by further empirical investigations of the relation between the mind and brain and between artificial intelligence software and underlying hardware will we be able to achieve a defensible solution to the mind-body problem. The major disadvantage of parallel systems is the need to coordinate their subprocesses, but recent proposals that consciousness provides a serial control for parallel computation are implausible.

THE MIND-BODY PROBLEM

The currently dominant position in the philosophy of mind is *functionalism*, which says that mental states are to be understood in terms of their functional relationships to other mental states, not in terms of any particular material instantiation (Dennett, 1978; Fodor, 1968, 1981a, 1981b; Putnam, 1975). The rejection of a direct mind-matter link distinguishes functionalism from the *mind-body identity theory*, according to which types of mental states such as thoughts are identical to types of states in the brain. The primacy of functionalism over the identity theory is based on a simple but powerful argument, the *argument from multiple instantiation*. Mental states cannot, in general, be brain states, the argument runs, since we should allow that mental states such as thoughts may occur in material in-

I am grateful to Arthur Burks, Paul Churchland, Douglas Hofstadter, Philip Johnson-Laird, and Stephen White for helpful suggestions.

Correspondence and requests for reprints should be sent to the author at the Cognitive Science Laboratory, 221 Nassau St., Princeton, NJ 08542.

stantiations quite different from the human brain. For example, we should allow the possibility that the same mental states that occur in human brains can occur also in silicon-based digital computers, or in extra-terrestrial beings whose thoughts derive from substances and architectures vastly different from those of humans.

Functionalism has not been without its critics. The most persistent criticism has been that functionalism neglects the special character of conscious experience: feelings or "qualia" (Block, 1978). From a very different quarter has come the criticism that functionalism errs in supposing that there is an established set of psychological notions—thoughts, feelings, and so on—to be functionally characterized. Instead, we should expect that advances in the neurosciences will lead us to a very different set of categories for describing mental states, eliminating the old ones (Churchland, 1981, 1984; Churchland, 1983). This position is called *eliminative materialism* in distinction from the *reductive* materialism of the identity theory: The traditional categories for mental states will be replaced, not reduced to neurophysiological categories.

This paper attacks functionalism head-on, contending that the argument from multiple instantiation is computationally naive. In computational terms, functionalism is the claim that only software matters to the mental. The argument from multiple instantiation says that we can ignore hardware in characterizing the mental, since the same software can run on any number of different kinds of hardware: It is the functional performance of the software which is crucial. However, current research in computer science, including artificial intelligence, is heavily concerned with developing parallel computer architectures as an alternative to the standard serial von Neumann architectures. By discussing the various reasons for preferring parallel architectures over serial ones, I shall attempt to show that even from a computational perspective hardware does matter. Parallelism leads to more than just improvements in speed of processing; its adoption makes possible qualitatively different kinds of algorithms for intelligent operations. Hence functionalism errs in abstracting from hardware, and understanding of mental states may well require attention to various kinds of material instantiation.

The result will be a position that emphasizes the need for constructing a new set of categories for understanding the relation of mind and matter. But these categories should not simply come from neuroscience as the Churchlands seem to presume. Artificial intelligence has a long way to go, but undeniable progress has been made in such areas as problem solving, machine learning, and game playing. We must seek a set of categories broad enough to embrace both human and artificial intelligence, but much further scientific investigation on topics such as parallel computation will be needed before we can say with any reliability what those categories are.

PARALLEL COMPUTATION

Although the serial von Neumann architecture has been dominant in the almost 40-year history of the stored program computer, there has long been interest in parallel architectures. Von Neumann himself explored an abstract kind of highly parallel computer called a "cellular automaton" (Burks, 1970), and numerous suggestions have been made and sometimes implemented for parallel computers. Recent super-computers such as the CRAY-1 and the CDC CYBER 205 use parallelism to speed number-crunching activities (Hockney & Jesshope, 1981).

More relevant to the understanding of mind, some researchers in artificial intelligence have been turning to parallel architectures. A brief description of some of these efforts will set the stage for subsequent arguments about the quantitative and qualitative benefits of parallelism. Current commercial applications, the supercomputers, are primarily concerned with the quantitative benefits—speed. Artificial intelligence researchers are also very concerned with speed, but in addition emphasize that parallelism can lead to different *styles* of programming. Examples of parallel architectures for artificial intelligence are the NETL and THISTLE systems of Fahlman (1979) and Fahlman, Hinton, and Sejnowski (1983), the Boltzmann machines of Hinton and Sejnowski (1983) and Fahlman et al. (1983), and the connectionist machines of Hillis (1981) and Feldman and Ballard (1982). Japan's ambitious project for a fifth generation computer also looks to parallelism to produce the speed necessary for intelligent operations (McCorduck & Feigenbaum, 1983). Models of parallel distributed information processing are also being applied to psychological phenomena (Rumelhart, Hinton, & McClelland, in press). Although it is much too soon to endorse any of these projects as a key to parallel artificial intelligence, they are useful concrete illustrations of how parallel architectures can differ from serial ones.

Fahlman (1979) proposed that the speed of information retrieval from semantic networks could be greatly increased using a parallel architecture. In a semantic network, concepts are represented by nodes with links between them. For example, we know that Burger King is a kind of restaurant, which is a kind of business, which is a kind of organization, which is a kind of thing. Traditionally, these links are represented by propositions or slots in frames. (For an account of the epistemology of frame-based semantic networks, see Thagard, 1984.) Fahlman proposes instead that the nodes be independent processors and that links between them are actual wires. Communication in the system takes place by sending markers through the system of nodes along wires. Because the wires are independent of each other, much communication can take place in parallel. Suppose, for example, you want to find whether there are any Canadian violinists. You simultaneously propagate markers from the nodes of Canadian and violinists, and if any nodes

end up marked by both propagations then you know there is some intersection. On a serial machine, set intersection takes time proportional to the size of the smallest of the sets being intersected, but parallelism allows the comparison to occur in one operation once the markers have been propagated through the system.

In Fahlman's NETL, nodes and markers handle only relatively simple boolean operations. In contrast, Hillis (1981) has designed a message-passing machine which consists of more complex nodes (which he calls "cells") that have several registers and are capable of arithmetic operations. What the cells can communicate to neighbors is thus much more complex than in NETL: messages rather than mere markers. (In contrast, the similarly termed "connectionist" models of Feldman [Feldman & Ballard, 1982] are closer to marker passing than message passing machines.) Hewitt has been investigating the semantics of message passing in order to develop software systems for parallel processing (Hewitt & deJong, 1983).

A third kind of parallel operation is found in the "Boltzman" machines of Hinton and Sejnowski (1983). These are pattern-recognition devices consisting of numerous binary units operating in parallel. Objects in the environment are recognized by patterns of activity in the units, with different units given weights which are adjusted in response to success or failure in recognition tasks. This machine is called Boltzmann because it uses an algorithm for recognition of objects that is based on a thermodynamic model.

Thus at various research centers work proceeds apace on parallel architectures for artificial intelligence. I shall now argue that this line of work has metaphysical as well as scientific significance.

SPEED AND THE MIND-BODY PROBLEM

The speedup possible from parallel computation is well documented (see Gottlieb & Schwartz, 1982, p. 28, for a summary of the speed gains for a wide variety of processes). Parallelism is not a panacea. It is no help with functions that require exponentially increasing time to compute (Cook, 1983); if the time to compute a function of n steps is proportional to 2^n , parallelism cannot ward off the explosion that occurs for large n . But many processes using functions which increase in polynomial time can be significantly sped up using parallelism.

Functionalists may deny, however, that the importance of parallel computation counts against their position. Parallelism, runs the reply, is irrelevant since mere speed of processing is incidental to the functional arrangements of states. After all, the theory of computation tells us that we can simulate any parallel machine on any serial machine equivalent to a universal Turing machine, which operates serially with only a tape and the ability to write 0s and 1s on the tape. Conceptually, therefore, parallelism adds nothing.

Perhaps, but metaphysics is not a purely conceptual enterprise. From a mathematical perspective, a Turing machine is a perfectly suitable model for the operation of a computer, but it is a seriously defective model for understanding intelligence. Mathematicians have no need to concern themselves with the patent inefficiencies of Turing machines, since they have no concern to build or use them. But intelligence should not be viewed as a purely abstract notion like computational complexity. Intelligent systems must operate in real time. If human beings had thought processes that operated orders of magnitude more slowly, it is obvious that our species would never have evolved. The external environment puts definite constraints on admissible speed of operation: One has only so much time to flee a charging tiger or to find the next water hole. Artificial intelligence is also subject to real-time constraints. An expert system that requires months to do a computation diagnosing a patient's illness would be useless in most cases. A natural-language understander must react with sufficient speed to interact with a speaker.

Thus the slow-down resulting from serial simulation of parallel processes can matter, given environmental constraints. Of course, in environments different from ours, there may be more or less severe time constraints. It is easy to imagine high-pressure worlds in which only organisms that react much faster than we do could survive, such as ones that are continually pelted by meteors to be dodged. At the other extreme, we can imagine languid environments in which lack of external threat gives plenty of time for reflection. But any real intelligent system comes into existence in an environment in which it must perform. If it is a natural system, it is constrained by Darwinian selection; if it is an artificial system, it is constrained by the needs of its designers. Either way, an excessively slow system will not survive for long. Intelligence, then, should be viewed as relative to the environment in which behavior must occur.

Hence we cannot ignore the speed advantage of parallel over serial systems, since the time advantage of the former over the latter may in many environments make all the difference for the naturally or artificially determined survival of the system. This argument will be unsatisfying to philosophers who seek conceptual illumination in thought experiments. Imagine, for example the Chinese Mind of Block (1978). We are asked to consider an experiment in which the billion people of China pass cards among each other, functionally simulating a computer. Abstractly, we can certainly consider this an intelligent system, but like the Turing machine it is not one which could function in any environment we know. In the first place, the logistics of setting such a machine up are obviously unsurmountable, and in the second place it would be too slow to perform any activity we could recognize as intelligent. Churchland and Churchland (1981) show that the Chinese Turing machine has too few connections even to simulate an earth-worm! My point in mentioning this example is that reflections on mind and

computation will likely go astray if based on examples that ignore real-world computational constraints.

A metaphysical theory of the relation of intelligence and matter need not cover all conceivable intelligences, any more than a physical theory must account for all conceivable motions. Newtonian mechanics, for example, is not expected to explain the behavior of logically possible worlds in which the force of gravitation is negative or in which mass spontaneously appears and disappears. A metaphysical theory is necessarily more general and abstract than a physical one, but shares the need to account for what is actual or realizable, not merely conceivable.

Functionalism assumes a sharp distinction between hardware and software, but real-world constraints necessitate loosening that distinction. Computer designers do not always produce a general-purpose machine and expect arbitrary software to follow it. In current technology, hardware blends into software and software into hardware. To improve efficiency, it is sometimes desirable to incorporate what would normally be thought of as the software into hardware. For example, the chess-playing program Belle is very fast because much of its searching of board positions is hard-wired, and LISP machines are becoming popular because their architecture permits much faster running of programs written in LISP than is possible on comparable general purpose machines (Deering, 1984). At the other end, microprogramming is a technique that makes hardware more flexible by allowing the programming of certain functions that might easily have been wired in. Burks (1981) has proposed that computers be programmed to adjust their architectures to different kinds of problems. Thus computer design gives a very different perspective from the functionalist view of fixed hardware and multiply instantiated software. Design requires an ongoing tradeoff between the speed of hard-wiring and the flexibility of programming.

My argument against functionalism is subject to the following objection. What I have been discussing is not really parallel architectures, but parallel *algorithms*. The functionalist can grant that computation may need parallel algorithms, but point out that these algorithms can be run on different kinds of architectures. Once again, this is true in principle but false in practice. For example, it is obvious that the algorithms devised to operate NETL, the connection machine, and the Boltzmann machine are enormously different, reflecting the very different architectures of the underlying machines. Which parallel algorithms are implemented will depend very much on our view of the hardware on which they must run.

Because of environmental constraints, speed matters to any computational system. In a given environment, only some combinations of hardware and software will yield acceptable performance. We should thus think of an intelligent system as a three-part complex:

< hardware, software, environment > .

But this is not the whole story, since speed is not the only factor affecting the relation between hardware and software.

THE QUALITATIVE IMPORTANCE OF PARALLELISM

According to Wirth (1976) and other theorists, a program should be understood as consisting of data structures and algorithms for manipulating those structures. The structures and algorithms are interdependent: The algorithms must work with the data in the form given to them. In languages like PASCAL, data structures are conceptually distinct from the procedures which use them, whereas in LISP procedures are themselves data structures, namely lists. In both cases, however, it is possible to specify algorithms without noting the kinds of structures on which they operate. Philosophers tend to assume the ubiquity of only one kind of data structure—the proposition, and only one kind of algorithm—logical reasoning. But computer science offers a wealth of structures in which data can be stored: arrays, tables, records, frames, and so on. Our view of the nature of thinking can be broadened considerably by allowing for the possibility of non-propositional data structures and non-logistic inference mechanisms (Thagard, 1984).

Programming is often a matter of *style*. Any programmer knows that some programming tasks are much easier to do in some languages than others. You could conceivably write AI programs in PASCAL, or even assembly language, but it is much easier to design and write such symbol-manipulating programs in LISP. Thus qualitatively it is much easier to produce programs in languages which provide facilities for the appropriate kinds of data structures and algorithms. Some programming theorists even urge a kind of computational Whorf hypothesis, claiming that using a particular programming language can have a substantial effect on how problems are conceived. (Dijkstra jokes that BASIC and FORTRAN cause permanent brain damage.)

These features of programming point to a general argument for the qualitative importance of hardware. Some programming tasks are much more naturally done using particular kinds of data structures and algorithms found in particular programming languages. And, as we saw in the last section, great gains in efficiency and ease of use can be achieved by tailoring hardware for particular programming functions. Hence in contrast to the in-principle compatibility of any program with any hardware, we find in practice that a good fit of software and hardware is indispensable.

In the remainder of this section, I shall try to illustrate this general lesson with specific cases concerning parallel computation. Parallel architectures offer not merely speed, but different *kinds* of programs which have the potential of being more reliable, more flexible, and more easily produced than programs for serial computers.

Reliability

Parallelism offers much more natural ways of providing for system reliability than are found in serial machines. Compare, for example, the effects of removing part of the memory of a digital computer with the effect of removing a similarly small part of the human brain. The brain's memory and processing capacity seem to be distributed over large areas, so that remaining parts can compensate for what has been removed. In contrast, removal of storage for part of a serial computer program will result eventually in a total bomb of the program. Parallel machines such as that of Hillis (1981, p. 9) can operate much more like the human brain. A system with a few faulty cells can continue to function, since algorithms do not depend on a cell existing at a specific address. The neighbors of a cell can identify it as defective and effectively ignore it, with performance continuing with only a slight degradation.

We can of course contrive reliability with serial computers. In the early days of computers when failure of vacuum tubes was frequent, *two* computers were sometimes used in tandem, providing checking and backup for each other. But it is clearly more efficient to avoid this total duplication of resources and build some degree of reliability into each system.

Flexibility

Most philosophers and computer scientists abhor inconsistency (Psychologists, in contrast, often enjoy it.) Popper (1965) and others have argued that an inconsistent system is worthless, since any proposition follows logically from a contradiction. Quine (1960) has urged a "principle of charity" which requires that we always interpret the utterances of others in such a way as to avoid finding them in violation of the rules of logic. Artificial intelligence researchers who see logic as the paradigm for knowledge representation (Nilsson, 1983) are similarly appalled by the havoc that inconsistency can wreak in an elegant system.

In contrast, Minsky (1974) has argued that consistency is not a paramount virtue; a sufficiently flexible system can function despite contradictions (cf. Thagard, 1984). Thagard and Nisbett (1983) contend that it is sometimes legitimate to attribute irrationality to humans, if there is an empirically supported account of what they are doing instead of following the laws of logic. Consistency, then, need not be a defining characteristic of an intelligent processing system.

This is especially clear from the perspective of parallel computation. Unlike a serial machine a parallel machine does not need detailed coordination of its components. It does not matter if the information in one of the cells of the connection machine contradicts that in other cells, although at some point a real conflict—one that causes processing problems—may arise.

To make this point concrete, consider scientific communities as highly parallel systems (Kornfeld & Hewitt, 1981). Whereas individuals are gener-

ally expected to maintain consistency and coherence in their beliefs, a community can be expected to have sharply competing views. Proponents of different theories fight it out in the journals and other public forms. Arguably, this kind of competition is better suited to the goals of scientific research than a more monolithic approach would be, since it is difficult to predict from what quarters good new ideas will come. Scientific communities require some degree of coordination to function, but they can clearly accommodate some differences in doctrine and even in method.

Why not allow the same flexibility in an individual? Rather than imposing uniformity, different parts of a processing system can pursue different strategies for attacking problems. Pursuing multiple hypotheses in parallel is clearly an effective strategy in scientific communities, and may well also be useful in a processing system. The alternative is to fix on a canonical set of ideas too soon, or to undergo repeated Popperian oscillations and reject well-developed sets of ideas.

Parallelism lends itself to audacity. With multiple hypotheses a system can afford to maintain daring but improbable hypotheses which stand little chance of being true, but which may lead to great payoffs in the unlikely event they work out (Holland, 1983). Proceeding serially, a system must tend more to look for hypotheses that are only optimal in a limited local context.

The flexibility of parallelism is evident in a LISP program called PI, for "processes of induction" (Thagard & Holyoak, 1985; Holland, Holyoak, Nisbett, & Thagard, 1986). PI simulates problem solving and inductive inference: In the context of attempts to solve problems, various kinds of induction, including generalization, specialization, abduction, and concept formation, are triggered. PI simulates parallelism by allowing the firing of any number of production rules at a single timestep, so that no strict priority of rules need be maintained. Spreading activation of concepts and the different kinds of learning also occur in parallel. The result is that the system need not concentrate on only one possible solution to a problem at a time, but can simultaneously be considering different tacks. PI simulates the discovery of the wave theory of sound by the Roman architect Vitruvius, but is also able at the same time to discover and explore the consequences of a particle theory of sound.

Thus a parallel architecture more naturally gives rise to mechanisms of rational deliberation which admit flexibility in considering multiple hypotheses. Various parts of the system can work out solutions without constantly checking on what other parts of the system are doing. Sometimes, such as when external action is required, at least a partial unification must occur. How control is established is an open question. Below I shall consider whether consciousness has this function.

Another way in which parallelism can encourage flexibility is through the emergence, rather than the explicit programming, of important structures.

Rumelhart et al. (in press) describes how schemas can be understood as emergent from much simpler connections in parallel-distributed processing systems. One result is that schemas need not offer monolithic characterizations of kinds of things, but may be constructed as situations demand. A system would not store a rigid, unified schema for restaurant, for example, but would have a set of expectations about what is likely to happen in a restaurant emerge from the parallel activity of simpler structures.

Producibility

No processing system is created from scratch. The human mind is the product of millions of years of evolution, and design of a modern computer also has to build on ideas which already exist. My claim in this section is that parallel systems might be more “producible” in some contexts than serial ones.

Biological evolution has proceeded without any overall design, with progressively more complex information processing systems being built on top of existing ones. The current human mind-brain is a consequence of the whole evolutionary chain of mammalian development. Consider, for example, the interactions of cognition and emotion, a currently active research area in psychology (Bower, 1981; Zajonc, 1980). There is increasing evidence that fundamental operations of human intelligence such as memory retrieval are tied in with the operations of emotions, and hence with physical operations which we presumably share with our less cognitively developed ancestors. We might be deluded that cognition can be discussed independently of anything material, but it is clear that the understanding of emotion can not ignore physical elements such as glands and hormones. If human cognition is so intimately intertwined with emotion, then the understanding of cognition must similarly require reference to the physical constitution of humans.

If artificial intelligence were easier to devise, producibility might not be an issue for computers. In the early days of artificial intelligence, there was much hope that programmers could directly enter into computers enough information to make them intelligent, but it is increasingly clear that this kind of spoonfeeding has limitations. Expert systems are proliferating, but each is restricted to a very narrow domain. To be intelligent, computers must have some of the flexibility and learning capacity that people do. Thus parallel computation, if it brings the benefits described above, might allow intelligent machines to be produced by human designers who cannot see the whole, incredibly complex picture. Parallelism would allow greater subdivision of design tasks without the necessity of worrying about all the interactions that might occur.

It might be objected that you do not need parallelism to have the kind of modularity I have been advocating. Modern programming languages such

as ADA allow for modular development, yet are designed for use with serial computers. However, programming in such language requires careful attention to all the interfaces between the modules. In a parallel system such as Hillis's connection machine or Holland's classifiers, no such special interfaces between the independent sections need be produced.

Thus both human and artificially intelligent systems must have a history. The genesis of intelligence, both in the species and in the individual, must be taken into account. I therefore propose to expand the earlier characterization of intelligence, to encompass:

< hardware, software, environment, history > .

Here, "history" is understood to include both the specific learning experience of the system and the evolution or design history of its species. In a given historical context, parallelism can be a boon for both the individual and the species of intelligent system.

Despite the many advantages so far discussed, parallel computation has many problems, the major one being how to coordinate the operations of the independent units. I shall now examine the proposal that consciousness plays an important role in directing parallel computation in humans.

CONSCIOUSNESS

The problem of consciousness has been the bane of functionalism. It has frequently been objected that analyzing mental states in terms of their functional relations neglects the crucial *experiential* quality of our mental states. We are aware of our thoughts, perceptions, and feelings in a way in which a functionally equivalent robot might not be.

But what is consciousness for? Discussion of mental operations in terms of parallelism suggests an important role for consciousness. The most striking difference between conscious experience and the processes so far discussed is the serial character of consciousness. Whereas many different processes can occur at the same time in a parallel system, conscious experiences come serially, even if continuously. This makes consciousness a prime candidate for solution of the major problem of parallelism mentioned above: How can we control and coordinate the independent operations of the parallel subsystems? Johnson-Laird (1983a, 1983b) and Burks (1984) have proposed to understand consciousness as the central control system which operates on top of the many parallel sub-systems in the human brain. Although this view has much initial plausibility, I shall argue against it, proposing that consciousness may have no essential computational function, or at best a function very different from executive control.

I shall not be advocating the position, primarily held by behaviorists, that consciousness is an illusion. Humans have conscious experience, and so

might lower animals and sophisticated robots. But this consciousness may not play an essential part in the information processing of the organism or computer.

Burks (1984) states: "Viewed from the perspective of computer architecture, human consciousness is a particular kind of computer control system, a relatively simple real-time control which, when the system is awake, directs short-term activities and plans longer-term activities" (p. 16). Similarly, according to Johnson-Laird (1983a): "The brain is a parallel computer that is organized hierarchically. Its operating system corresponds to consciousness and it receives only the results of the computations of the rest of the system" (p. 584).

This view seems plausible since parallel systems such as the brain do seem to need some kind of control system, and serial consciousness appears to fit the bill. Parallelism has to be limited somehow: To act, an organism has to select one action from numerous alternatives which its subsystems might be considering. Funneling the alternatives through consciousness which selects from among them seems like a natural way to handle the problem. But there are numerous reasons for questioning whether consciousness actually has the specified function.

First, despite the apparent convenience of having a serial control on top of a parallel system, it is not necessary that control be either serial or centralized. Such current computers as the CRAY-1 and CDC CYBER 205 do in fact have serial central processing systems governing their parallel operations, but many other proposed parallel computers do not. Von Neumann's cellular automata, for example, behave in ways determined solely by the transition functions and states of cells and their neighbors (Burks, 1970). There is no central control system at all, except perhaps for an underlying clock which synchronizes transitions in discrete time steps. Even this degree of coordination may be superfluous: Hewitt's APIARY model is avowedly asynchronous. The human brain appears to function extremely well without any general executive or clock.

Second, even if the brain does need a control system, there is no reason to say that it has anything to do with conscious experience. Burks distinguishes between "functional consciousness" and immediate experience, and argues that a system can have a control system corresponding to the former without having the latter. But what is consciousness without experience? What he calls functional consciousness sounds more like a control system independent of consciousness. It is not at all clear what the essential experiential feature of consciousness would contribute to the conceivably independent function. There are many control systems, such as those proposed by Fahlman (1979) and Holland (1983), that centralize control of parallel systems without anything approximating consciousness.

I have argued so far that parallel intelligence does not require a central control and that control can be established without consciousness. My third

argument is the most direct and important. There are reasons for believing that consciousness does not in fact provide a control for the operations of the underlying parallel systems. Notoriously, we cannot decide consciously what we want to think about. If I ask you to try consciously not to think about elephants, you will undoubtedly fail. Most of us have had the experience of feeling that we were dwelling excessively on some annoying issue, yet being unable to drive it from our minds. Subjects given post-hypnotic suggestions carry out the suggested actions perfectly well without having any conscious knowledge of why they are performing as they are. A wealth of experimental studies supports the view that people are often unaware of the causes of their behavior (Nisbett & Wilson, 1977). Churchland (1983) has compiled a long catalogue of kinds of intelligent activities that proceed without self-conscious awareness.

None of these phenomena is strictly incompatible with Johnson-Laird's view which assumes that the mind has a great deal of hierarchical organization. He says that consciousness controls only the top level, with lower level functions not directly accessible from consciousness. You cannot consciously control your heartbeat, although you can consciously do things that will affect your heartbeat. But if consciousness is not needed as a direct control for the vast range of unconscious phenomena, from heartbeats to inferences, why is it needed as a control at all? Perhaps at best consciousness directly controls only those functionings of the brain concerned with intentional action, where we have to make decisions for which there are conflicting reasons. Even here, the process by which one set of reasons dominates another, producing a favored course of action, is not wholly conscious: How a particular decision was reached is something that even the decider often must reconstruct.

These considerations undermine the claim that consciousness is a central control mechanism. What then is consciousness for? One radical answer is that it does not have to be for anything at all. Consider the difference between the old vacuum tube computers and modern ones composed of silicon chips. The old ones produced vast amounts of heat, and sometimes designers would monitor their existence by placing a radio on top of them and listening to the resulting hum—a certain continuous hum would indicate an infinite loop. In contrast, silicon machines produce much less heat and hum. Consciousness may be like the heat or the hum or the smell of the computer. It is a side effect of the particular kinds of hardware and software being used, but is not of any particular importance in understanding that hardware or software. We should expect different organisms with different kinds of hardware and software to have different types of consciousness, and it may well be that only certain kinds are capable of having consciousness. But because of their epiphenomenal character, the nature of the experiences may not play a role in the theory of intelligence of the system.

It would be premature, however, to dismiss consciousness as having no computational importance. Perhaps we should agree with Johnson-Laird that consciousness is important for resolving some conflicts, even if it is not a general control mechanism. Another suggestion (owed to Ziva Kunda) is that consciousness in humans is important for look-ahead. In planning, we often carry out a kind of simulation of what we expect different actions to lead to. Doing this consciously keeps us from confusing the simulation with what is actually occurring. Of course, other computational mechanisms could have the same function in other kinds of systems. Another suggestion (due to John Kihlstrom) is that the primary function of consciousness is pedagogical. By virtue of our awareness of what we do, we are able to teach others how to duplicate our performance. Consciousness may be irrelevant to the performance itself, but crucial for communicating about it with others. A final suggestion (due to Johnson-Laird) is that consciousness is important for social relations, because it makes possible a kind of empathy. Because I am aware of my own feelings, I can appreciate how others seem to be feeling, giving me much more understanding of their behavior. By studying natural systems and designing artificial ones, we may gain a much better understanding of the role of consciousness in intelligence. Thus, although we have reason to doubt that consciousness is the mind's central control mechanism, the conclusion that consciousness is epiphenomenal is premature.

FUNCTIONALISM AND MATERIALISM

With the importance of parallelism in mind, let us now return to the metaphysical question of the relation of mind and body. We must not look only at the relation between human minds and human bodies, since artificial intelligence offers the potential of different kinds of minds in different kinds of materials. We must accordingly seek a solution to a general "intelligence-matter" problem, which asks: What is the relation between intelligent systems and their underlying material instantiation? (I am ignoring dualist or idealist views that say there does not need to be any material instantiation.) Functionalists say that there does not have to be any special relation between an intelligent system and a material instantiation, since any intelligence could be duplicated on any material device powerful enough to compute: Any device that can simulate a Turing machine can do any computation you might want. We saw that this mathematical fact is irrelevant to understanding intelligent systems, which must be producible and operate in real time. The functionalist argument from material instantiation relies on the following principle:

- (F1) Any set of algorithms producing intelligent behavior run on one kind of hardware can be run on any hardware capable of computing.

The principle fails to defeat materialists, who can maintain that parallelism and concerns about real environments and real-time computability show that there may in fact be a one-one relation between intelligent systems and their material instantiations. The failure of the argument from multiple instantiation does not, however, show that materialism is true. Eliminativist materialists such as the Churchlands endorse neurophysiology as the key to understanding of mind, but the possibility of artificial intelligence shows that neuroscience need not be the only source of ideas concerning the relation of intelligence and matter; the human brain may not be the only repository of natural or artificial intelligence.

Moreover, standard versions of materialism would be undermined if the following quasi-functional principle turns out to be true:

- (F2) For any set of algorithms capable of intelligent behavior and running on a kind of hardware, there is another quite different kind of hardware on which the algorithms can be run.

The truth of this principle would vindicate a weakened functionalist claim that intelligent behavior is at least somewhat independent of hardware. Could we then talk only of the algorithms rather than pay attention to the hardware? No, because the hardware may in fact place a very tight constraint on the algorithms and vice-versa, even if the constraint is loose enough to allow more than a simple one-one relation between software and hardware.

My conclusion is that we currently know too little about the human mind and brain and about the range of possibility of other kinds of intelligence to form a plausible solution to the intelligence-matter problem. Any answer offered at this point would be a generalization from one ill-understood instance, the brain. It may turn out that the same algorithms that operate in the human brain can operate *in real time* on parallel hardware quite different from the brain. This is an empirical issue, that can be answered only by future investigations. It may turn out on the other hand that intelligence can only be realized by the brain. Or we may find that it is possible to develop artificial intelligence, but only by building machines that compute algorithms very different from those computed by the brain. In either of these two cases, we would have to say that some form of materialism is true, since the functioning of thought is tied in a one-to-one fashion with the material substrate.

On the other hand, it may turn out that it is possible to develop artificial minds that compute algorithms very much like humans. This will be a partial vindication of the functionalist argument, since it will have turned out as a matter of fact that multiple instantiations are an empirical possibility, not just a conceptual one. Similarly, if extra-terrestrial beings arrive with manifest intelligence, then we would have another empirical source of information. We would want to ask the questions: Do they compute with algorithms like ours? Do they use hardware like ours? It may turn out that the func-

tionalist assumption that our algorithms are computable on very different hardware is borne out, but given current ignorance it is equally likely that the extra-terrestrials would have very different algorithms and hardware from us, with a fit between their algorithms and hardware that makes their kind of intelligence very different from ours. Thus the only way to resolve current disputes between functionalists and materialists is to continue scientific investigations of the nature of human thought and of the possibility of artificial thought. The metaphysical issue depends, just as it should, on the scientific one.

What about consciousness and emotions? Again this is an empirical question. It may turn out that the complex control structures needed to develop intelligence will require something like emotions, along with hardware corresponding to all the apparatus that produced emotions in human beings. Once again, there is no hope of purely conceptual or thought-experimental answers to the question. We have to see if development of intelligent computers will require features so similar to what seems to go on in humans that we must attribute emotions and consciousness to computers. Consciousness may prove to be a desirable feature to have in a smart system. We still do not know enough about either the human mind or artificial minds to determine the extent of the interdependency of mind and matter.

My non-dogmatic conclusion is that resolution of the mind-body problem will have to await additional understanding about the range of realistic possibilities of interaction of intelligence and matter. This understanding will require attention to the four factors mentioned above, <hardware, software, environment, history>, and to the correlative importance of parallel computation. The key question is: Given constraints provided by environments and historical developments, what kinds of hardware-software combinations can give rise to intelligence? Before it can be answered, however, much more knowledge must be gained in all the cognitive sciences concerning the properties of natural and artificial systems.

REFERENCES

- Block, N. (1978). Troubles with functionalism. In C.W. Savage (Ed.), *Perception and cognition*. Minneapolis: University of Minnesota Press.
- Bower, G. (1981). Mood and memory. *American Psychologist*, 36, 129-148.
- Burks, A. (1970). *Essays on cellular automata*. Urbana: University of Illinois Press.
- Burks, A. (1981). Programming and structure changes in parallel computers. In W. Handler (Ed.), *CONPAR 1981*. Berlin: Springer-Verlag.
- Burks, A. (1984). *An architectural theory of human consciousness. Logic of Computers* (Tech. Rep. No. 254). Ann Arbor, MI: University of Michigan.
- Churchland, P.S. (1983). Consciousness: The transmutation of a concept. *Pacific Philosophical Quarterly*, 64, 80-95.
- Churchland, P.S. & Churchland, P. (1981). Functionalism, qualia, and intentionality. *Philosophical Topics*, 12, 121-145.

- Churchland, P. (1981). Eliminative materialism and the propositional attitudes. *Journal of Philosophy*, 78, 67-90.
- Churchland, P. (1984). *Matter and consciousness*. Cambridge, MA: Bradford Books/MIT Press.
- Cook, S.A. (1983). An overview of computational complexity. *Communications of the ACM*, 26, 401-408.
- Deering, M. (1984). Hardware and software architectures for efficient AI. *Proceedings of the National Conference on Artificial Intelligence*. Palo Alto, CA: William Kaufmann.
- Dennett, D. (1978). *Brainstorms*. Montgometry, VT: Bradford Books.
- Fahlman, S. (1979). *NETL: A system for representing and using real world knowledge*. Cambridge, MA: MIT Press.
- Fahlman, S., Hinton, G., & Sejnowski, T. (1983). Massively parallel architectures for AI: NETL, THISTLE, and Boltzmann Machines. *Proceedings of the National Conference on Artificial Intelligence*. Los Altos, CA: William Kaufmann.
- Feldman, J., & Ballard, D. (1982). Connectionist models and their properties. *Cognitive Science*, 6, 205-254.
- Fodor, J. (1968). *Psychological explanation*. New York: Random House.
- Fodor, J. (1981a). The mind-body problem. *Scientific American*, 244, 114-123.
- Fodor, J. (1981b). *Representations*. Cambridge, MA: MIT Press.
- Gottlieb, A., & Schwartz, J. (1982). Networks and algorithms for very-large-scale parallel computation. *Computer*, 15, 27-36.
- Hewitt, C., & de Jong, P. (1983). Analyzing the roles of descriptions and actions in open systems. *Proceedings of the National Conference on Artificial Intelligence*. Los Altos, CA: William Kaufmann.
- Hillis, D. (1981). *The connection machine* (A.I. Memo No. 646). Cambridge, MA: MIT Artificial Intelligence Laboratory.
- Hinton, G.E. & Anderson, J. (Eds.). (1981). *Parallel models of associative memory*. Hillsdale, NJ: Erlbaum.
- Hinton, G., & Sejnowski, T. (1983). Analyzing cooperative computation. *Proceedings of the Fifth Annual Conference of the Cognitive Science Society*. Rochester, NY.
- Hockney, R., & Jesshope, C. (1981). *Parallel computers*. Bristol: Adam Hilger.
- Holland, J. (1983). Escaping brittleness. *Proceedings of the International Machine Learning Workshop*. Urbana: University of Illinois, Department of Computer Science.
- Holland, J., Holyoak, K., Nisbett, R., & Thagard, P. (1986). *Induction: Processes of inference, learning, and discovery*. Cambridge, MA: Bradford Books/MIT Press.
- Johnson-Laird, P. (1983a). A computational analysis of consciousness. *Cognition and Brain Theory*, 6, 499-508.
- Johnson-Laird, P. (1983b). *Mental models*. Cambridge, MA: Harvard University Press.
- Kornfeld, W., & Hewitt, C. (1981). The scientific community metaphor. *IEEE Transactions on Systems, Man, and Cybernetics*.
- McCorduck, P., & Feigenbaum, E. (1983). *The fifth generation*. Reading, MA: Addison-Wesley.
- Minsky, M. (1974). *A framework for representing knowledge* (A.I. Memo No. 306). Cambridge, MA: MIT Artificial Intelligence Laboratory.
- Nilsson, N. (1983). Artificial intelligence prepared for 2001. *The AI Magazine*, 4, 7-14.
- Nisbett, R., & Wilson, T.D. (1977). Telling more than we can know: Verbal reports on mental processes. *Psychological Review*, 84, 231-259.
- Popper, K. (1965). *Conjectures and refutations*. New York: Basic Books.
- Putnam, H. (1975). *Mind, language, and reality*. Cambridge: Cambridge University Press.
- Quine, W.V.O. (1960). *Word and object*. Cambridge, MA: MIT Press.
- Rumelhart, D., Hinton, G., & McClelland, J. (in press). *Parallel distributed processing: Explorations in the microstructure of cognition*. Cambridge, MA: Bradford Books/MIT Press.

- Thagard, P. (1984). Frames, knowledge, and inference. *Synthese*, 61, 233–259.
- Thagard, P., & Holyoak, K. (1985). Discovering the wave theory of sound: Induction in the context of problem solving. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*. Los Altos, CA: Morgan Kaufman.
- Thagard, P., & Nisbett, R. (1983). Rationality and charity. *Philosophy of Science*, 50, 250–267.
- Wirth, N. (1976). *Algorithms + data structures = programs*. Englewood Cliffs, NJ: Prentice-Hall.
- Zajonc, R. (1980). Feeling and thinking: Preferences need no inferences. *American Psychologist*, 35, 151–175.