THE BEHAVIORAL AND BRAIN SCIENCES (1980), 3 111-169
Printed in the United States of America

Computation and cognition: issues in the foundations of cognitive science

Zenon W. Pylyshyn
**Center for Advanced Study in the Behavioral Sciences, Stanford, Calif. 94305*

Abstract: The computational view of mind rests on certain intuitions regarding the fundamental similarity between computation and cognition. We examine some of these intuitions and suggest that they derive from the fact that computers and human organisms are both physical systems whose behavior is correctly described as being governed by rules acting on symbolic representations. Some of the implications of this view are discussed. It is suggested that a fundamental hypothesis of this approach (the "proprietary vocabulary hypothesis") is that there is a natural domain of human functioning (roughly what we intuitively associate with perceiving, reasoning, and acting) that can be addressed exclusively in terms of a formal symbolic or algorithmic vocabulary or level of analysis.

Much of the paper elaborates various conditions that need to be met if a literal view of mental activity as computation is to serve as the basis for explanatory theories. The coherence of such a view depends on there being a principled distinction between functions whose explanation requires that we posit internal representations and those that we can appropriately describe as merely instantiating causal physical or biological laws. In this paper the distinction is empirically grounded in a methodological criterion called the "cognitive impenetrability condition." Functions are said to be cognitively impenetrable if they cannot be influenced by such purely cognitive factors as goals, beliefs, inferences, tacit knowledge, and so on. Such a criterion makes it possible to empirically separate the fixed capacities of mind (called its "functional architecture") from the particular representations and algorithms used on specific occasions. In order for computational theories to avoid being ad hoc, they must deal effectively with the "degrees of freedom" problem by constraining the extent to which they can be arbitrarily adjusted post hoc to fit some particular set of observations. This in turn requires that the fixed architectural function and the algorithms be independently validated. It is argued that the architectural assumptions implicit in many contemporary models run afoul of the cognitive impenetrability condition, since the required fixed functions are demonstrably sensitive to tacit knowledge and goals. The paper concludes with some tactical suggestions for the development of computational cognitive theories.

Keywords: cognitive science; artificial intelligence; computational models; computer simulation; cognition; mental representation; mental process; imagery; philosophical foundations; functionalism; philosophy of mind

1. Introduction and summary

The view that cognition can be understood as computation is ubiquitous in modern cognitive theorizing, even among those who do not use computer programs to express models of cognitive processes. One of the basic assumptions behind this approach, sometimes referred to as "information processing psychology," is that cognitive processes can be understood in terms of formal operations carried out on symbol structures. It thus represents a formalist approach to theoretical explanation. In practice, tokens of symbol structures may be depicted as expressions written in some lexicographic notation (as is usual in linguistics or mathematics), or they may be physically instantiated in a computer as a data structure or an executable program.

The "information processing" idiom has been with us for about two decades and represents a substantial intellectual commitment among students of cognition. The fields that share this view (notably, segments of linguistics, philosophy of mind, psychology, artificial intelligence, cultural anthropology, and others) have been increasingly looking toward some convergence as the "cognitive sciences." Several journals devoted to that topic now exist (including, to some extent, *BBS*), and a Cognitive Science Society has just been formed. There remains, however, considerable uncertainty regarding precisely what constitutes the core of the approach and what constraints it imposes on theory construction.

In this essay I shall present what I consider some of the crucial characteristics of the computational view of mind and defend them as appropriate for the task of explaining cognition. As in the early stages of many scientific endeavors, the core of the approach is implicit in scientists' intuitions about what are to count as relevant phenomena and as legitimate explanations of the underlying processes. Yet as we tease out the central assumptions, we will find room for refinement: not everything that is intuitively cognitive will remain so as the theory develops, nor will all processes turn out to be appropriate for explaining cognitive phenomena.

We begin with an informal discussion of the position that certain types of human behavior are determined by *representations* (beliefs, tacit knowledge, goals, and so on). This, we suggest, is precisely what recommends the view that mental activity is computational. Then we present one of the main empirical claims of the approach – namely, that there is a natural domain of inquiry that can be addressed at a privileged *algorithmic* level of analysis, or with a proprietary vocabulary.

The remainder of the paper elaborates various requirements for constructing adequate explanatory theories on this basis. First, however, we need to analyse the notions of

©1980 Cambridge University Press 0140-525X/80/010111-59$4.00/0

"cognitive phenomenon" and "cognitive process" in the light of the evolving understanding of cognition as computation. Hence we take a detour in our discussion (sections 5 and 6) to examine the fundamental distinction between 1) behavior governed by rules and representations, and 2) behavior that is merely the result of the causal structure of the underlying biological system. A primary goal of the paper is to defend this as a principled distinction, and to propose a necessary (though not sufficient) empirical criterion for it. This methodological criterion (called the *cognitive impenetrability condition*) supports the maxim that to be explanatory, a theoretical account should make a *principled* appeal to a computational model. Exhibiting an algorithm and a representative sample of its behavior is not enough. One must, in addition, separate and independently justify two sources of its performance: its fixed (cognitively impenetrable) functional capacities (or its "functional architecture"), and the "effectiveness principles" underlying its capacity to perform as claimed.

2. The cognitive vocabulary and folk psychology

Most people implicitly hold a sophisticated and highly successful cognitive theory; that is, they can systematize, make sense of, and correctly predict an enormous range of human behavior. Although textbook authors are fond of pointing out the errors in folk psychology, it nonetheless far surpasses any current scientific psychology in scope and general accuracy. What is significant about this is the mentalistic vocabulary, or level of description, of folk psychology, and its corresponding taxonomy of things, behaviors, events and so on.

Whatever the shortcomings of folk psychology (and there are plenty, to be sure), the level of abstractness of its concepts (relative to those of physics), and particularly its appeal to the way situations are *represented in the mind* (i.e., its appeal to what human agents think, believe, infer, want, and so on, as opposed to the way they actually are), seems precisely suited to capturing just the kinds of generalizations that concern cognitive psychology – e.g., the nature of our intellectual abilities, the mechanisms underlying intellectual performances, and the causal and rational antecedents of our actions.

It seems overwhelmingly likely that explanations of cognitive phenomena will have to appeal to briefs, intentions, and the like, because it appears that certain regularities in human behavior can only be captured in such terms and at that level of abstraction. For example, when a person perceives danger, he will generally set about to remove himself from the source of that danger. Now, this generalization has an unlimited variety of instances. Thus, generally, if a person *knows* how to get out of a building, and *believes* the building to be on fire, then generally he will set himself the *goal* of being out of the building, and use his knowledge to determine a series of actions to satisfy this goal. The point is that even so simple a regularity could not be captured without descriptions that use the italicized mentalistic terms (or very similar ones), because there is an infinite variety of specific ways of "knowing how to get out of the building," of coming to "believe that the building is on fire," and of satisfying the goal of being out of the building. For each combination of these, *an entirely different causal chain* would result if the situation were described in physical or strictly behavioral terms. Consequently the psychologically relevant generalization would be lost in the diversity of possible causal connections. This generalization can only be stated in terms of the agent's *internal representation* of the situation (i.e. in mentalistic terms). For a different example, the laws of color mixture are properly stated over *perceived color*, or what are called "metameric" equivalence classes of colors, rather than over physically specifiable properties of light, since they hold regardless of how the particular color is produced by the environment – e.g., whether, say, the perceived yellow is produced by radiation of (roughly) 580 nm, a mixture of approximately equal energy of radiations of 530 nm and 650 nm, by a mixture of any complex radiations that metamerically match the latter two wavelengths, or even by direct electrical or chemical stimulation of the visual system. We might never be able to specify all the possible physical stimuli that produce a particular perceived color, and yet the laws of color mixture shall hold if stated over perceived color. Hochberg (1968) presents a variety of such examples, showing that the regularities of perception must be stated over *perceived* properties – i.e., over internal representations.

Similarly, when a particular event can be given more than one interpretation, then what determines behavioral regularities is not its physical properties, or even some abstract function thereof, but rather each agent's particular *interpretation*. The classical illustrations are ambiguous stimuli, such as the Necker cube, the duck-rabbit or the profiles-vase illusions, or ambiguous sentences. Clearly, what people do (e.g., when asked what they see or hear) depends upon which reading of the ambiguity they take. But all physical events are intrinsically ambiguous, in the sense that they are subject to various interpretations; so psychological regularities will always have to be stated relative to particular readings of stimuli (i.e., how they are internally represented).

Finally, if we include *goals* as well as beliefs among the types of representations, it becomes possible to give an account of a wide range of additional regularities. Behavior that is goal-directed is characterized by such properties as equifinality (i.e., its termination can only be characterised in terms of the terminating state, rather than in terms of the path by which the system arrived at that state – see the discussion in Newell and Simon 1972).

It is no accident that the systematicity of human behavior is captured in a vocabulary that refers to internal representations, for these are the terms in which we conceptualize and plan our actions in the first place. For example, as I write this paper, I produce certain movements of my fingers and hand. But I do that under control of certain higher level goals, or, as some people prefer to put it, I execute the behaviors under a certain "intended interpretation." I *intend* to make certain *statements* by my behavior. I do not intend to make marks on paper, although clearly I am doing that too. Although this hierarchical aspect of the behavioral description is part of my conceptualization of it, rather than an intrinsic part of the behavior itself, yet it is critical to how the behavior must be treated theoretically if the theory is to capture the systematicity of my actions. A theory that took my behavior to be an instance of finger-movement could not account for why, when my typewriter broke, I proceeded to make quite different movements using pencil and paper. This is an instance of the "equifinality" property associated with goal-directed behavior.

Of course, to say that folk psychology has nonfortuitously settled on some of the appropriate terms for describing our cognitive activity is not to say that it is good scientific theory. It may be that this set of terms needs to be augmented or pruned, that many of the beliefs expressed in folk psychology are either false or empirically empty, and that many of its explanations are either incomplete or circular. But its most serious shortcoming, from the point of view of the scientific enterprise, is that the collection of loose generalizations that makes up this informal body of knowledge is not tied together into an explicit system. The way in which sets of generalizations are tied together in developed sciences is through a theory that shows how the generalizations are derivable in some appropriate idealization from a smaller set of deeper universal principles (or axioms). The categories of the deeper principles are typically quite different from those found in the broader generalization (e.g. pressure and temperature are reduced to aspects of kinetic energy in molecular theory.)

3. Representation and computation

There are many characteristics that recommend computation as the appropriate form in which to cast models of cognitive processes. I have discussed some of them in Pylyshyn (1978a). For example, the hierarchical character of programs and the abstraction represented by the "information processing" level of analysis make it the ideal vehicle for expressing *functional* models of all kinds. These were the aspects of computation that led Miller, Galanter, and Pribram (1960) to propose the basic iterative loop (TOTE, or Test-Operate-Test-Exit) as the fundamental building block for psychological theory – to replace the reflex arc and even the cybernetic energy feedback loop.

What I wish to focus on here is what I take to be the most fundamental reason why cognition ought to be viewed as computation. That reason rests on the fact that computation is the only worked-out view of *process* that is both compatible with a materialist view of how a process is realized and that attributes the behavior of the process to the operation of rules *upon representations. In other words, what makes it possible* to view computation and cognition as processes of fundamentally the same type is the fact that both are physically realized and both are governed by rules and representations [see Chomsky, this issue]. Furthermore, they both exhibit the same sort of dual character with respect to providing explanations of how they work – and for the same reason.

As a physical device, the operation of a computer can be described in terms of the causal structure of its physical properties. The states of a computer, viewed as a physical device, are individuated in terms of the identity of physical descriptions, and its state transitions are therefore connected by physical laws. By abstracting over these physical properties, it is possible to give a *functional* description of the device. This is a description of the systematic relations that hold over certain (typically very complex) classes of physical properties – such as the ones that correspond to computationally relevant states of the device. While the transitions over states defined in this way are no longer instances of physical laws (i.e., there is *no law* relating state *n* and state *m* of an IBM machine, even though it is wired up so that state *m* always follows state *n*), they are nonetheless reducible to some complex function of various physical laws and of the physical properties of states *n* and *m*. Such a functional description of the device might, for example, be summarized as a finite state transition diagram of the sort familiar in automata theory. We shall see below, however, that this is not an adequate functional description from the point of view of understanding the device as a computer.

On the other hand, if we wish to explain the computation that the device is carrying out, or the regularities exhibited by some particular *programmed* computer, we must refer to objects in a domain that is the *intended interpretation* or the subject matter of the computations, such as, for example, the abstract domain of numbers. Thus, in order to explain why the machine prints out the symbol "5" when it is provided with the expression "(PLUS 2 3)," we must refer to the meaning of the symbols in the expression and in the printout. These meanings are the referents of the symbols in the domain of numbers. The explanation of why the particular symbol "5" is printed out then follows from these semantic definitions (i.e., it prints out "5" because that symbol represents the number five, "PLUS" represents the addition operator applied to the referents of the other two symbols, etc, and five is indeed the sum of two and three). In other words, from the definition of the symbols (numerals) as representations of numbers, and from the definition of the "PLUS" as representing a certain abstract mathematical operation, it follows that some state of the machine after reading the expression will correspond to a state that represents the value of the function and (because of a further *definition of the implicit printout function) causes the* printout of the appropriate answer.

This is true of computation generally. We explain why the machine does something by referring to certain interpretations of its symbols in some intended domain. This is, of course, precisely what we do in describing how (and why) people do what they do. In explaining why a chess player moves some piece onto a certain square, we refer to the type of piece it is in terms of its role in chess, to the player's immediate goal, and to the rules of chess. As I suggested earlier, this way of describing the situation is not merely a way of speaking or an informal shorthand reference to a more precise functional description. Furthermore, it is not, like the functional description referred to earlier, an abstraction over a set of physical properties. There is a fundamental difference between a description of a computer's operation cast in terms of its states (i.e., equivalence classes of physical descriptions) and one cast in terms of what it is *about*, such as in the illustrative example above. The fundamental difference is *that the former refers to intrinsic properties of the device,* while the latter refers to aspects of some entirely different domain, such as chess. The former can be viewed as a syntactic description, while the latter is semantic, since it refers to the represented domain.

This dual nature of mental functioning (referred to traditionally as the functional or causal, and the intentional) has been a source of profound philosophical puzzlement for a long time (e.g. Putnam 1978). The puzzle arises because, while we believe that people do things because of their goals and beliefs, we nonetheless also assume, for the sake of unity of science and to avoid the extravagance of dualism, that this process is actually carried out by causal sequences of events that can respond only to the intrinsic physical properties of the brain. But how can the process depend both on properties of brain tissue and on some other quite different domain, such as chess or mathematics? The parallel question can of course equally be asked of computers: How can the state transitions in our example depend both on physical laws and on the abstract *properties of numbers? The simple answer is that this* happens because both numbers and rules relating numbers are *represented* in the machine as symbolic expressions and programs, and that it is the physical realization of these representations that determines the machine's behavior. More precisely, the abstract numbers and rules (e.g. Peano's axioms) are first expressed in terms of syntactic operations over symbolic expressions or some notation for the number system, and then these expressions are "interpreted" by the built-in functional properties of the physical device. Of course, the machine does not interpret the symbols as numbers, but only as formal patterns that cause the machine to function in some particular way.

Because a computational process has no access to the actual represented domain itself (e.g., a computer has no way of distinguishing whether a symbol represents a number or letter or someone's name), it is mandatory, if the rules are to continue to be semantically interpretable (say as rules of arithmetic), that all relevant semantic distinctions be mirrored by syntactic distinctions – i.e., by features intrinsic to the representation itself. Such features must in turn be reflected in functional differences in the operation of the device. That is what we mean when we say that a device *represents* something. Simply put, all and only syntactically encoded aspects of the represented domain can affect the way a process behaves. This rather obvious assertion is the cornerstone of the formalist approach to understanding the notion of *process.* Haugeland (1978) has made the same point, though in a slightly different way. It is also implicit in Newell's (1979) "physical symbol system" hypothesis. Many of the consequences of this characteristic of computation and of this way of looking at cognition are far-reaching, however, and not widely acknowledged [for a discussion, see Fodor, this issue].

By separating the semantic and syntactic aspects of cognition, we reduce the problem of accounting for meaningful action to the problem of specifying a mechanism that operates upon meaningless symbol tokens and in doing so carries out the meaningful process being modelled (e.g. arithmetic). This, in turn, represents an important breakthrough because, while one can see how the formal process can be realized by causally connected sequences of events (as in a computer), no one has the slightest notion of how carrying out semantically interpreted rules could even be viewed as compatible with natural law [c.f. Fodor, this issue]. That is, as far as we can see there could be no natural law which says, for example, that when (and only when) a device is in a state that represents the number *five*, and applies the operation that represents the successor function, it will go into the state that represents the number *six*. Whatever the functional states that are being invoked, there is nothing to prevent anyone from consistently and correctly interpreting the same states and the same function as representing something quite different – say, producing the name of the next person on a list, or the location of an adjacent cell in a matrix, or any other consistent interpretation. Indeed, the very same physical state recurs in computers under circumstances in which very different processes, operating in quite different domains of interpretation, are being executed. In other words, the machine's functioning is completely independent of how its states are interpreted (though it is far from clear whether this would still remain so if it were wired up through transducers to a natural environment). For that reason we can never specify the behavior of a computer uniquivocally in terms of a semantically interpreted rule such as the one cited above (which referred to the domain of numbers). This is what people like Fodor (1978), Searle (1979), or Dreyfus (1979) mean when they say that a computer does not know what it is doing.

The formalist view requires that we take the syntactic properties of representations quite literally. It is literally true of a computer that it contains, in some functionally discernable form (which could even conceivably be a typewritten form, if someone wanted to go through the trouble of arranging the hardware that way), what could be referred to as a code or an inscription of a symbolic expression, whose formal features mirror (in the sense of bearing a one-to-one correspondence with) semantic characteristics of some represented domain, and which causes the machine to behave in a certain way. Because of the requirement that the syntactic structure of representations reflect all relevant semantic distinctions, the state transition diagram description of an automaton is an inadequate means of expressing the functional properties of a computational system. Individual states must be shown as factored into component parts rather than as being distinct atomic or holistic entities. Some functionally distinguishable aspects of the states much correspond to individual terms of the representing symbol structure, while other aspects must correspond to such additional properties as the control state of the device (which determines which operation will be carried out next) and the system's relation to the representation (e.g., whether the representation corresponds to a belief or a goal). Components of the syntactic expressions must be functionally factorable, otherwise we could not account for such regularities as that several distinct representational states may be followed by the same subsequent state, or that rules tend to be invoked in certain systematic ways in relation to one another. Indeed, one could not represent individual *rules* in the state transition notation, since individual rules affect a large (and in principle unbounded) set of different states. For example, the rule that specifies that one remove oneself from danger must be potentially evokable by every belief state a part of whose representational content corresponds to danger. By representing this regularity once for each such state, one misses the generalization corresponding to that one rule. In addition, of course, the fact that there are an unbounded number of possible thoughts and representational states makes it mandatory that the symbolic encoding of these thoughts or states be combinatoric – i.e., that they have a recursive syntactic structure. I mention all this only because there have been some who have proposed that we not view the content of states in terms of some articulation of what they represent (e.g. Davidson 1970) – i.e., that we avoid postulating an internal syntax for representations, or a "mentalese."

The syntactic, representation-governed nature of computation thus lends itself to describing cognitive processes in such a way that their relation to causal laws is bridgeable, at least in principle. But beyond that, the exact nature of the device that instantiates the process is no more a direct concern to the task of discovering and explaining cognitive regularities than it is in computation – though in both cases specifying the fixed functional architecture of the underlying system is an essential component of understanding the process itself. Given that computation and cognition can be viewed in these common abstract terms, there is no reason why computation ought to be treated as merely a metaphor for cognition, as opposed to a hypothesis about the literal nature of cognition. In spite of the widespread use of computational terminology (e.g., terms like "storage," "process," "operation"), much of this usage has had at least some metaphorical content. There has been a reluctance to take computation as a *literal* description of mental activity, as opposed to being a mere heuristic metaphor. In my view this failure to take computation literally has licensed a wide range of activity under the rubric of "information processing theory," some of it representing a significant departure from what I see as the core ideas of a computational theory of mind.

Taking a certain characterization literally carries with it far-reaching consequences. The history of science contains numerous examples of the qualitative changes that can come about when a community accepts a certain characterization as applying literally to phenomena. The outstanding example of this is the case of geometry. Our current scientific conception of physical space is a projection of Euclidean geometry onto the observations of mechanics. But plane geometry was well known and widely used by the Egyptians in surveying. Later it was developed into an exquisitely elegant system by the Greeks. Yet for the Egyptians it was a way of calculating – like an abacus – while for the Greeks it was a demonstration of the perfect Platonic order. It was not until two millenia later that Galileo began the conceptual transformation that eventually resulted in the view, that is so commonplace today that virtually no vestige remains of the Aristotelian ideas of natural motions and natural places. Everyone imagines space to be that empty, infinitely extended, isotropic, three-dimensional receptacle, whose existence and properties are quite independent of the earth or any other objects. Such a strange idea was literally unthinkable before the seventeenth century. In fact, not even Galileo completely accepted it. For him a straight line was still bound to the earth's surface. It was not until Newton that the task of "geometrization of the world" was completed (to borrow a phrase from Butterfield 1957).

The transformation that led to the reification of geometry –

to accepting the formal axioms of Euclid as a literal description of physical space – profoundly affected the course of science. Accepting a system as a literal account of reality enables scientists to see that certain further observations are possible and others are not. It goes beyond merely asserting that certain thinks happen "as if" some unseen events were taking place. In addition, however, it imposes severe restrictions on a theory-builder, because he is no longer free to appeal to the existence of unspecified similarities between his theoretical account and the phenomena he is addressing – as he is when speaking metaphorically. It is this latter degree of freedom that weakens the explanatory power of computation when it is used metaphorically to describe certain mental functions. If we view computation more abstractly as a symbolic process that transforms formal expressions that are in turn interpreted in terms of some domain of representation (such as the numbers), we see that the view that mental processes are computational can be just as literal as the view that what IBM computers do is properly viewed as computation.

Below I shall consider what is entailed by the view that mental activity can be viewed literally as the execution of algorithms. In particular I shall suggest that this imposes certain constraints upon the theory construction enterprise. If we are to view cognition as literally computation, it then becomes relevant to inquire how one can go about developing explanatory theories of such cognitive processes in selected domains and also to consider the scope and limit of such a view. In other words, if computation is to be taken seriously as a literal account of mental activity, it becomes both relevant and important to inquire into the sense in which computational systems can be used to provide *explanations* of mental processes. Before turning to the general issue of what is entailed by a literal view of cognition as computation, I shall conclude this discussion of the formalist nature of this approach with one or two examples, suggesting that even proponents of this view sometimes fail to respect some of its fundamental constraints. One still sees the occasional lapse when the psychological significance or the psychological principles implicated in some particular computer model are being described. Thus, semantic properties are occasionally attributed to representations (for an animated discussion, see Fodor 1978), and even more commonly, principles of processing are stated in terms of the semantics of the representations.

For example, in arguing for the indeterminacy of forms of representation, Anderson (1978) proposed a construction by means of which a system could be made to mimic the behavior of another system while using a form of representation different from that used by the mimicked system. While I have already argued (Pylyshyn 1979b) that much is wrong with Anderson's case, one flaw that I did not discuss is that the mimicking model is required (in order to be constructable as in Anderson's "existence proof") to have access to the semantics of the original representations (i.e., to the actual objects being referred to) in the system it is mimicking – which, as we have seen, is not possible according to the computational view. This occurs in Anderson's construction, because in order to decide what representation to generate next, the mimicking model must first determine what the mimicked model would have done. It does this by first determining what representation the original system would have had at that point and what new representation it would have transformed it into. Then it is in a position to infer what representation it should generate in its mimicking process. But the only way the mimicking system could be assured of finding out what representation the target model would have had (and the way actually invoked in the proposed construction) is to allow it to find out which stimuli correspond to its current representation, and then to compute what the target model would have encoded them as in its own encoding scheme. This step is, furthermore, one that must be carried out "on line" each time an operation is to be mimicked in the mimicking model – it cannot be done once and for all in advance by the constructor itself, except in the uninteresting case where there are only a finite number of stimuli to be discriminated (not merely a finite number of codes). But finding out which real stimuli generate a certain code in a model is precisely to determine the semantic extension of a representation – something that a formal system clearly cannot do.

Similarly, in describing the principles of operation of a model, it is common to characterize them in terms of the represented domain. For example, when it is claimed that "mental rotation" of images proceeds by small angular steps (e.g. Kosslyn and Shwartz 1977; Anderson 1978), or that "mental scanning" of images proceeds by passing through adjacent places, one is appealing to properties of the represented domain. Phrases like "small angular steps" and "adjacent places" can only refer to properties of the stimulus or scene being represented. They are semantically interpreted notions. The representation itself does not literally *have* small angles or adjacent places (unless one wishes to claim that it is laid out spatially in the brain – a logical possibility that most people shun); it only *represents* such properties. Of course, it is possible to arrange for a process to transform a representation of a stimulus into a representation of the same stimulus in a slightly different orientation, or to focus on a part of the representation that corresponds to a place in the stimulus that is adjacent to another specified place, but in neither case is this choice of transformation principled [see Kosslyn et al.: "On the demystification of mental imagery," *BBS* 2(4) 1979]. In other words, there is nothing about the representation itself (e.g. its syntax or format) that requires one class of transformation rather than another – as is true in the situation being represented, where to physically go from one point to another, one *must* pass through intermediate adjacent points or else violate universal laws of (classical) physics. Thus the existence of such a constraint on permissible transformations in the imaginal case is merely stipulated. The choice of the transformation in that case is like the choice of a value for a free empirical parameter – i.e., it is completely *ad hoc* – in spite of the appeal to what sounds like a general principle. However, because the principle as stated applies only to the semantic domain, one is still owed a corresponding principle that applies to the representational or syntactic domain – i.e., which applies in virtue of some intrinsic property of the representation itself. This issue is discussed in greater detail in Pylyshyn (1979c).

4. Algorithmic models and the proprietary vocabulary hypothesis

We begin our discussion of how a literal computational view can provide an explanatory account of cognition by examining what sorts of questions a computational model might be expected to address. One of the discoveries that led to the development of computer science as a discipline is that it is possible to study formal algorithmic processes without regard to how such processes are *physically* instantiated in an actual device. What this means is that there is an interesting and reasonably well-defined set of questions concerning computational processes (e.g., what is the fastest or the least memory-consuming way of realizing a certain class of functions?), the answers to which can be given without regard to the material or hardware properties of the device on which these processes are to be executed. This is not to suggest, of course, that issues of material realization are uninteresting, or even that they are irrelevant to certain other questions about computation. It simply implies that in studying computation it is possible, and in certain respects essential, to factor apart the nature of the symbolic process from properties of the physical device in which it is realized. It should be noted that the finding that there is a natural domain of questions concerning the behavior of a system, which is independent of the details of the physical structure of the device – providing only that the device is known to have a certain general functional character – is a substantive one. Prior to this century there was no reason to believe that there was even a coherent notion of *process* (or of mechanism) apart from that of physical process. The new notion initially grew out of certain developments in the study of the foundations of mathematics, especially with the work of Alan Turing (1936).

Because of the parallels between computation and cognition noted earlier, the corresponding view has also been applied to the case of mental processes. In fact, such a view has a tradition in philosophy that, in various forms, predates computation – though it has received a much more precise formulation in recent years in the form known as the representational theory of mind, one of the most developed versions of which can be found in Fodor (1975). One way of stating this view is to put it in terms of the existence of a privileged or a proprietary vocabulary in which the structure of mental processes can be couched – *viz.* the vocabulary needed to exhibit algorithms.

More precisely, the privileged vocabulary claim asserts that there is a natural and reasonably well-defined domain of questions that can be answered solely by examining 1) a *canonical description* of an algorithm (or a program in some suitable language – where the latter remains to be specified), and 2) a system of formal symbols (data structures, expressions), together with what Haugeland (1978) calls a "regular scheme of interpretation" for interpreting these symbols as expressing the representational content of mental states (i.e., as expressing what the beliefs, goals, thoughts, and the like are about, or what they represent). Notice that a number of issues have been left unresolved in the above formulation. For example, the notion of a canonical description of an algorithm is left open. We shall return to this question in section 8. Also, we have not said anything about the scheme for interpreting the symbols – for example, whether there is any indeterminacy in the choice of such a scheme or whether it can be uniquely constrained by empirical considerations (such as those arising from the necessity of causally relating representations to the environment through transducers). This question will not be raised here, although it is a widely debated issue on which a considerable literature exists (e.g. Putnam 1978).

A crucial aspect of the assumption that there exists a fixed formal vocabulary for addressing a significant set of psychological questions is the view that such questions can be answered without appealing to the material embodiment of the algorithm, and without positing additional special analytical functions or relations which themselves are not to be explained in terms of algorithms. In fact, as I shall argue presently, one can take the position that this proprietary vocabulary or level of description *defines* the notion of *cognitive phenomenon in the appropriate technical sense* required for explanatory theories. It should, however, be emphasized that such a definition is presented in the spirit of a broad empirical hypothesis. This hypothesis would be falsified if it turned out that this way of decomposing the domain of cognitive psychology did not lead to any progress or any significant insights into human thought and human rationality. Thus when I spoke of this level as specifying a domain of questions, it was assumed that such a domain is a natural one that (at least to a first approximation) can be identified independently of particular algorithms – though, like the notion of a well-formed sentence, it will itself evolve as the larger theoretical system develops. The tacit assumption, of course, has been that this domain coincides with what we pretheoretically think of as the domain of cognition – i.e., with the phenomena associated with language, thought, perception, problem-solving, planning, commonsense reasoning, memory, and so on. However, there may also turn out to be significant departures from the pretheoretical boundaries. After all, we cannot be sure in advance that our commonsense taxonomy will be coextensive with the scope of a particular class of theory – or that we have pretheoretically carved Nature exactly at her joints.

One of the main points of this article will be that the proprietary level hypothesis, together with certain intrinsic requirements on the use of algorithmic models to provide theoretical explanations, leads to certain far-reaching consequences. For the next two sections, however, I shall make a somewhat extensive digression to establish a number of background points. Discussions of cognition and computation frequently founder on certain preconceived views regarding what a cognitive phenomenon is and what a cognitive process is. Thus, for example, cognitive science is sometimes criticized for allegedly not being able to account for certain types of phenomena (e.g. emotions, consciousness); at other times, attempts are made to provide explanations of certain observations associated with thinking (e.g. reaction time, effect of practice) in terms of *ad hoc* mixtures of biological and computational mechanisms. In many of these cases what is happening is that inappropriate or unclear ideas concerning the notion of cognitive phenomenon or cognitive process are being invoked, to the detriment of theoretical clarity. Consequently it is important to our story that we first try to clarify these notions in the following digression.

5. *What is a cognitive phenomenon?*

Many things happen when people are engaged in thinking or problem-solving. For example, they may ask questions or report being puzzled, frustrated, or following false leads, they may get up and walk around or jot down notes or doodles; their attentiveness to environmental events may deteriorate; various physiological indicators such as skin resistance, peripheral blood flow, or skin temperature (as measured by GSR, plethysmograph, or thermometer) may change systematically, and finally, they may report what they believe is a solution. In addition, these various events may occur over time in certain systematic ways. Now, one might ask which, if any, of these observations could be explained by examining a canonical description of an algorithm, or, taking the position that algorithmic accountability defines the now technical notion "cognitive phenomenon," one might ask which of the above reports represents an observation of a cognitive phenomenon?

It seems clear that this question cannot be answered without bringing in other considerations. In particular, we must know how the observations are being interpreted in relation to the algorithm. For example, a cognitive model (of the kind I have been speaking about) would not account for people's ability to move their limbs in certain ways (e.g. in certain directions and at certain speeds) or for the spectral properties of the sounds they make, although it ought to be able to account in some general way for what people do (i.e. what intended actions they carry out) in moving their limbs, or for what they *say* in making those sounds. Thus, even when a phenomenon is clearly implicated in the cognitive activity, its relevance is restricted to the case in which it is appropriately described or interpreted (and, as we noted earlier, the appropriate interpretation will generally correspond to what the people themselves intended when they performed the action).

But what of the other observations associated with the problem-solving episode, especially the physiological indicators, task-independent performance measures (such as measures of distractability), and the temporal pattern associated with these observations? Here the case has typically not been as clear, and people's intuitions have differed. Thus the manner in which we interpret measures such as the galvanic skin response (GSR) and time-of-occurence in relation to the algorithm depends on various methodological assumptions, and these are just beginning to be articulated as the methodological foundations of cognitive science evolve.

For example, it is clear that a cognitive model will not have values for blood flow, skin resistance, or temperature among its symbolic terms, because the symbols in this kind of model must designate mental representations (i.e., they must designate mental structures that have representational content – such as thoughts, goals, and beliefs). The calculation of temperature and resistance in this case does not itself represent a cognitive process, and the values of these parameters are simply intrinsic physical magnitudes – they themselves do not represent anything (see the further discussion of this in section 6, below). However, such measures can be (and often are) taken as indices of certain aggregate properties of the process. For example, they might be taken as indices of what could be thought of as "processing load," where the latter is theoretically identified with, say, the size and complexity of the data structures on which the model is operating. Whether such an interpretation of these observations is warranted depends on the success of the ancilliary hypotheses in accounting for the relation among observations in the past. In other words, the justification of subsidiary methodological assumptions is itself an empirical question. There is no room here for *a priori* claims that a cognitive theory *must* account for these (or any other observations, and hence that certain phenomena are necessarily (or analytically) "cognitive."

Consider the following example, in which the developing methodology of cognitive science has led to a gradual shift in the way an important aspect of observed behavior is interpreted. The example concerns what is probably the most widely used dependent measure in cognitive psychology, namely, reaction time. This measure has sometimes been interpreted as just another response, to be accounted for by a cognitive model in the same way that the model accounts for such response properties as which button was pressed. Since Donders's (1969) pioneering work (carried out in the 1860s), it has also been widely interpreted as a more-or-less direct measure of the duration of mental processes [see Wasserman and Kong: "The Absolute Timing of Mental activities *BBS* 2(2) 1979]. I have argued (e.g. in commenting on the Wasserman and Kong paper) that neither of these interpretations is correct in general – that, in general, reaction time can neither be viewed as the computed output of a cognitive process itself, nor as a measure of mental duration.

If reaction time were thought of as simply another response, then it would be sufficient if our computational model simply calculated a predicted value for this reaction time, given the appropriate input. But clearly that would not suffice if the computation is to be viewed as modelling the *cognitive process*. Contemporary cognitive theorists would not view a system that generated pairs of outputs, interpreted as the response and the time taken, as being an adequate model of the underlying process, no matter how well these outputs fit the observed data.

The reason for this intuition is significant. It is related to what people understand by the concept of a cognitive process, and to the evolving concept of strong equivalence of processes, to which we will return in section 6.1. As the construction and validation of computational models has developed, there has evolved a general (though usually tacit) agreement that the kind of input-output equivalence that Turing proposed as a criterion for intelligence is insufficient as a condition on the validity of such models. It was obvious that two quite different algorithms could compute the same input-output function. It gradually became clear, however, that there were ways to distinguish empirically among processes that exhibited the same input-output function, providing certain additional methodological assumptions were made. In particular, these additional assumptions required that one view certain aspects of the organism's behavior not as a response *computed* by the cognitive process, but rather as an independent indicator of some property of the algorithm by which the response was computed. There are, in fact, independent reasons for wanting to distinguish between the kind of behavior that is directly attributable to the symbolic process, and other kinds of behavior that are not viewed as "outputs" of the cognitive process. We shall return to these below when we examine the complementary notion of a cognitive process – a notion that is as much in a state of evolution as that of a cognitive phenomenon.

It has become customary in cognitive science to view reaction time in the same way that we view measures such as the GSR or plethysmograph records, or measures of distractability – namely as an index, or an observable *correlate*, of some aggregate property of the process. In particular, reaction time is frequently viewed as an index of what might be called "computational complexity," which is usually taken to correspond to such properties of the model as the number of operations carried out. A process that merely computed time as a parameter value would not account for reaction time viewed in this particular way, since the parameter would not express the computational complexity of the process.

Measures such as reaction time, when interpreted in this way, are extremely important to cognitive science, because they provide one possible criterion for assessing *strong equivalence* of processes. Thus, all processes that, for each input, produce 1) the same output, and 2) the same measure of computational complexity, as assessed by some independent means, are referred to as complexity-equivalent. The complexity-equivalence relation is a refinement of the input-output or the *weak equivalence* relation. However, it need not be identical to what we would call the relation of strong equivalence, since we must allow for the possibility that future methodologies or other theoretical considerations may refine the relation even further.

Nonetheless the complexity-equivalence relation remains a central one in cognitive science, and reaction time measures remain one of the primary methods of assessing complexity. Since complexity is a relative quality (e.g., it makes no sense to speak of the absolute complexity of a computation, only of its complexity in relation to other computations that utilize the same hypothetical operations), we are only concerned with measures of complexity up to a linear transform. Thus we would distinguish between two lookup processes if the complexity of one increased linearly with the number of items stored while the complexity of the other was independent of, or increased as the logarithm of, the number of items. However, we would not necessarily discriminate between two hypothetical processes if their complexity varied in the same way with the input, but one process took a constant number of steps more than the other or required a fixed amount of storage capacity more than the other, unless we had some independent calibration that enabled us to translate this difference into, say, absolute reaction-time predictions.

This idea of appealing to plausible methodological assumptions (which must in the long run themselves be justified by the success of the models to which they lead) in order to make finer discriminations among theoretical systems is exactly parallel to the situation that has developed in linguistics over the last two decades. Linguistic judgments made by competent speakers are not simply taken as part of the linguistic corpus. The methodological assumption is that these judgments provide an independent source of constraint on linguistic structures, defining the analogous relation of strong equivalence of grammars (cf. Chomsky 1975).

This view of the role of reaction-time measures takes it for granted that such measures cannot be interpreted as a direct observation of the duration of a mental process. A mental process does not possess the intrinsic property of duration, any more than it possesses the property of location, size, mass, electrical resistance, or concentration of sodium ions. Of course, the underlying brain events do have these properties, and we can, at least in principle, measure them. But one must be careful in drawing conclusions about properties of the cognitive process on the basis of such measurements.

Recall that we are examining the claim that there is a domain, roughly coextensive with the range of phenomena covered by our intuitive notion of cognition, that can be accounted for solely in terms of a canonical description of the symbol manipulation algorithm. Suppose we have some observations (duration being a case in point) of properties of the physical instantiation of the algorithm. The question is, can such evidence be treated as the measurement of a cognitive property? We have already seen how, with the aid of ancillary methodological hypotheses, it can be used as evidence in favor of one or another putative cognitive process or algorithm. We are now asking whether such measurements can be viewed as anything more than fallible correlates, whose validity in each case depends upon whether or not the ancillary hypothesis holds.

I have argued (e.g. in my commentary on the Wasserman and Kong paper, BBS 2(3) 1979, and in Pylyshyn 1979b) that the answer to this question must in general be *no*. There are many situations in which measurements of properties of the underlying physical events may tell us little about the algorithm. It may, instead, tell us either about the way in which the process is physically (i.e. neurophysiologically) instantiated, or it may tell us about the nature of the task itself. We will briefly consider these two cases since they reveal an important general point concerning the relations among cognitive phenomena, the task being carried out, the method the person is using, the fixed functional properties (or functional architecture) of the cognitive system, and the biological or physical properties of some particular (token) instantiation of that solution process.

Measurement such as reaction time are particularly unlikely to tell us much about the nature of the underlying biological mechanism in the case of what is often called a "higher-level cognitive function," in which the processing is not as closely tied to anatomical structures as it is, say, in certain parts of perception or motor coordination. While in many cases there would, no doubt, be a correspondence between the duration of some correlated physical measurements and such purely algorithmic properties as the number of steps carried out, which particular steps were carried out, whether parts of the algorithm were carried out serially or in parallel, and so on, this is not always the case. The explanation of some of the time differences (and every explanation of the absolute time taken) must appeal to some properties of the physical realization that are unique to that particular instantiation and therefore irrelevant to the algorithmic or process explanation in general. Such duration data may not make a valid discrimination among putative algorithms.

Using a computer as an example, we can readily see that some of the time differences might arise from the fact that a signal had farther to travel in some particular (token) occasion because of the way the machine was wired up and the way the algorithm was implemented in it; some might arise from variable delay physical effects, such as the distance that a moveable arm had to travel in making a disk access in that implementation; and some could even depend on physical properties of the noncomputational environment, as would be the case if real-time interrupts could occur. None of these properties bears on the nature of the algorithm, since they could be quite different for a different realization of the identical algorithm. Consequently, in this case, measuring such times would not help to distinguish different candidate algorithms. That is why time measurements alone cannot be taken as measurements of a property of the algorithmic process in the computer case. And for precisely the same reasons they cannot be taken literally as measurements of mental durations – only as indirect (as possibly false) indicators of such things as processing complexity, to be used judiciously along with other indirect sources of evidence in inferring underlying mental processes.

The other case in which the observations may tell us little about the cognitive process itself arises when the preliminary determinant of the behavior in question is what Newell and Simon (1972) call the "task demands" (which, incidentally, are not to be confused with such "experimental demand" factors as Kosslyn, Pinker, Smith, and Schwartz 1979 have tried to rule out in their defense of their interpretation of the "mental scanning" results). Consider, for example, the various observations associated with certain operations on mental images. A large number of these investigations (e.g. Shepard 1978; Kosslyn, in press) have proceeded by measuring the time it takes to imagine a certain mental action, such as rotating an image or scanning with one's attention between two points on a mental image. But what the analysis of these results has consistently failed to do is to distinguish two different tasks that could be subsumed under the same general instructions to the subject. The first is the task of simply using a certain form of representation to solve the problem. The second is to imagine actually seeing certain of the problem-solving events taking place. In the latter case one would expect various incidental properties of the real events to be duplicated in the imagining. For example, suppose you are asked to imagine two different events: call them E_1 and E_2. If you know that, in reality, E_1 takes, say, twice as long as E_2, and if you interpret the task as requiring you to imagine yourself *seeing* the events happening (as seems reasonable, given the ambiguity of the instructions in this respect), then surely the task requires, by its very definition, that you spend more time in the act of imagining E_1 than in the act of imagining E_2 (in fact, twice as long – if you are capable of the psychophysical task of generating a time interval corresponding to a particular known magnitude). The crucial difference between these two tasks is that quite different success criteria apply in the two cases; although "using an image" does not require, as a condition for having carried out the task, that one reproduce incidental characteristics of some imagined event such as its relative duration, the task of "imagining seeing it happen" does demand that this be done.

This is, in fact, part of a very general difficulty that permeates the whole range of interpretations of imagery research findings that appeal to such things as "analogue media." I discuss this issue in detail elsewhere (Pylyshyn 1979c). For the present purposes we might simply note that a variety of determinants of observed phenomena, other than the structure of the cognitive process or the functional architecture of the underlying system, are possible. In order for observations such as, say, that reaction time varies linearly with the imagined distance, to have a bearing upon the cognitive theory itself (i.e., upon the structure of the algorithm or the functional architecture of the model), it would be necessary to further show that under certain prescribed conditions (such as when subjects report that their answers come from examining their image) the relation between distance and time is a *necessary* one (i.e., it is not determined by beliefs, tacit knowledge, or certain specific goals – such as to reproduce aspects of the event as it would have been perceived). To my knowledge this has not been demonstrated – and furthermore, some preliminary evidence from our own laboratory suggests that it is false in the case of mental scanning (see Pylyshyn 1979c). Clearly, in the case of a situation such as imagining a heavy stone being rolled as opposed to a light object being flipped over, any differences in the time taken would be attributed to the subject's knowledge of the referent situation – i.e. to the task demand. It is not clear how many of the reported time functions for mental operations fall into this category. I suspect that not only mental scanning, but also such findings as the relation between "size" of mental images and time to report details is explainable in this way, although perhaps some of the mental rotation results are not – but then it appears that the latter are not explainable as holistic analogue processes either (cf. Pylyshyn 1979a).

Whatever the ultimate verdict on the correct explanation of the observed reaction time functions, considerations such as these do illustrate that certain distinctions need to be made before we are free to interpret the observations. For example, if we distinguish, as I suggested, between the case of applying operations to visual images of objects and the case of *imagining yourself seeing* operations physically applied to real objects, then it would follow that in at least the second of these cases the reaction time functions should to some degree (depending on subjects' capacity to carry out the required psychophysical task of generating appropriate outputs) mirror the real time functions of the corresponding events. Furthermore, on this view it is also not at all surprising that we cannot imagine (in the sense of imagining ourselves seeing) such things as four-dimensional space. Certainly this fact tells us nothing about the nature of the representation: It is a matter of definition that we cannot *imagine ourselves seeing* what is in principle not seeable! The important point is that, to the extent that task-demand factors can be viewed as the primary determinants of observations, such observations cannot be taken as having a bearing on the structure of our cognitive model.

We thus have another instance of the principle that whether or not some measurement turns out to be an observation of a "cognitive phenomenon" by our definition depends on how we interpret it. I have no doubt that we will refine our notion of "cognitive phenomenon," just as we refined our notion of "physical phenomenon" in the course of the development of our theories. It should not come as a shock to us, therefore, if certain ways of viewing phenomena ultimately end up as irrelevant to cognitive theory, even though they might have been part of our pretheoretic notion of cognition. As Chomsky (1964) has reminded us, it is no less true of cognitive science than of other fields that we start off with the clear cases and work out, modifying our view of the domain of the theory as we find where the theory works.

For example, Haugeland (1978) considers it a serious shortcoming of the computational view that it appears unable to explain such things as skills and moods. There are two assumptions here, for which little evidence exists at present, one way or the other. The first is that no aspect of skills or moods can be accounted for computationally. The second is that, under the appropriate interpretation, these are phenomena that we know *a priori* to be cognitive. Consider the case of skills. The popular view is that, at least in the case of motor skills, competence is built up by repeated practice (given the inherent talent) without any intellectual intervention. In fact, it is often believed that thinking about such skills impedes their fluency (recall the story about the centipede that was asked which foot it moved first). But there is now reason to believe that the acquisition and exercise of such skills can be enhanced by purely cognitive means. For example, imagined or "mental" practice is known to improve certain motor skills. Furthermore, a careful functional analysis of a skill such as juggling, in terms of its intrinsic hierarchical structure, can lead to methods for verbally instructing novices, which cuts the learning time to a fraction of what it would otherwise take (Austin 1974). Clearly, cognitive processes are relevant to motor skills. Equally clearly, however, certain aspects of their execution are purely biological and physical. How this problem will naturally factor will depend on what the facts are: it is not a question of which we can issue a verdict in advance.

The same goes for moods, emotions, and the like. It is very likely that there will be certain aspects of these phenomena that will resist functional analysis, and therefore computational explanation (as there have been aspects that have historically resisted every kind of analysis). For instance, the conscious feeling accompanying moods may not be susceptible to such analysis, except insofar as it has a cognitive component (e.g., insofar as it leads to people noticing that they are experiencing a certain mood). It would not even be surprising if the pervasive effects that moods have on cognitive activity (which Haugeland mentions) turn out to be outside the scope of computational explanation. We know that there are global changes in various aspects of cognitive activity that accompany biological development and endocrine changes. Such changes seem more akin to variations in the underlying functional architecture (a concept to which we shall return shortly) than to changes in the algorithm. Variations in such inner environments do not connect with mental representations the way variations in the perceived environment do: We do not perceive our hormonal levels and thus come to have new beliefs. The relation in that case does not appear to be of the appropriate type: It is biological and causal, rather than rational or cognitive. On the other hand, we cannot rule out the possibility that a broader view of computation might make it possible to subsume even these facts into a computational theory. The real mystery of conscious experience is that the fact that it appears to elude both functionalist (i.e. computational) and naturalist (i.e. identity theory) accounts. A possible approach to this dilemma (and one on which I have speculated in Pylyshyn 1979d) might be to allow cognitive states to have access (via internal transduction) to biological states (e.g., imagine a computer with a register whose contents correspond to its internal temperature or line voltage). Neither providing for a computational system to appeal directly to arbitrary causal properties of its physical instantiation, nor allowing it to alter its own functional architecture presents any difficulties in principle. In fact, we routinely do change the functional architecture of computers by means of programs – for example, by a process called *compiling*. Whether such methods will allow us to capture aspects of such global effects as those mentioned above is a question for future theories to answer.

6. What is a cognitive process?

6.1 Cognitive simulation and strong equivalence. There is a systematic ambiguity in the use of the term "computer simulation" that occasionally causes confusion. Computers are said to simulate economic cycles, traffic flow, chemical reactions, and the motion of celestial bodies. They are also said to simulate aspects of human behavior as well as cognition. It is important, however, to understand the essential difference between the latter sense of simulation (as a psychological model) and the former. When we simulate, say, the motion of planets, the only empirical claim we make is that the coordinate values listed on the printout correspond to the ones that will actually be observed under the specified conditions. Which algorithm is used to compute these values is irrelevant to the veridicality of the simulation. In other words, for this purpose we do not distinguish among algorithms that compute the same input-output function. We have refered to such algorithms as being *weakly equivalent*.

The case of cognitive simulation, however, is quite different. As we have already noted, weak equivalence is not a sufficient condition for validity of a model. Intuitively, we require that the algorithm correspond in much greater detail to the one the person actually uses. The difficulty is, of course, that we cannot directly observe this algorithm, so we must discriminate among weakly equivalent processes on the basis of other considerations. We have already suggested how observations such as reaction time can be used as evidence for deciding among weakly equivalent algorithms. Such criteria help us to select a set of what I have referred to as complexity-equivalent processes. Other criteria, such as those that provide evidence for intermediate states and for factorable subcomponents of the process (cf. Pylyshyn 1978a), as well as endogenous criteria, such as minimizing redundancy through appropriate subroutinizing, and further methodological criteria yet to be developed, define a sense of equivalence I have referred to earlier as *strong evidence*.

There is an important reason why strong equivalence is relevant to cognitive models, though not to other uses of computer simulation, such as those mentioned earlier. The reason was hinted at when we spoke of modelling the algorithm that the person "actually uses." The idea is that the appropriate way to functionally characterize the mental activities that determine a person's behavior is to provide an initial representational state – interpreted as representing beliefs, tacit knowledge, goals and desires, and so on – and a sequence of operations that transform this initial state, through a series of intermediate states, into the commands that are finally sent to the output transducers. All the intermediate states, on this view, are also representations which, in the model, take the form of expressions or data structures. Each of these has psychological significance: it must be interpretable as a mental representation. Thus all intermediate states of the model constitute claims about the cognitive process.

Contrast this with the case of simulating planetary motion. Clearly in this case no empirical claims whatsoever are being made about the intermediate states of the computation itself: only the outputs are interpreted as referring to the modelling domain. The ambiguity we spoke of earlier arises because of the use of the phrase "computer simulation" to refer to two quite different activities. In the case of planetary motion, chemical processes, traffic flow, and so on, what we call simulation is only a way of computing a series of values of a function, while in the psychological case, simulation refers to the execution of a hypothesized algorithm in a simulated (in the first sense) mechanism. In the computer industry the technical term for this kind of mimicry is *emulation:* one computer is sometimes made to emulate the functional capacities or functional architecture of another, so that programs originally written for the first can be run directly on the second. Thus, modelling cognitive processes must proceed in two phases. The first is to emulate the functional architecture of the mind, and the second is to *execute* the hypothetical cognitive algorithm on it (not to *simulate* the behavior of the cognitive process, for the algorithm represents a literal proposal for the structure of this process).

Notice that this discussion is relevant to the distinction made earlier, between observations directly attributable to the cognitive algorithm and ones that must be attributed to physical properties of the device, where we included GSR, plethysmograph, thermometer, and timing records in the latter category. Although, clearly, we could in principle compute values of these paramenters, and this computation itself would be some sort of process, yet it would not be a *cognitive* process. In computing these values we would, at best, be simulating the function of the biological mechanism, not literally executing *its* algorithm on an emulated architecture. The intermediate states of this process would not be viewed as having empirical cognitive referents. To the extent that the body could be said to "compute" GSR, it clearly does not do it symbolically. If it could be said to compute at all, it would be by analogue means (we shall have more to say below about the important question of when an organism or device symbolically computes a function, as opposed to merely instantiating or exhibiting that function).

Another closely related reason why strong equivalence is demanded of cognitive models, but not of models of physical processes, is that the former are assumed to be governed by rules acting upon symbolic representations. While we do not assume that planets have a symbolic representation of their orbits (or of the laws governing their trajectory), we *do* claim that the appropriate explanation of cognitive processes must appeal to the organism's use of rules and explicit symbolic representations. The distinction between behavior being governed by symbolic representations and behavior being merely exhibited by a device in virtue of the causal structure of that device is one of the most fundamental distinctions in cognitive science. We shall therefore devote the next section to examining that distinction.

6.2 Representation-governed behavior. The question of whether we should explain the behavior of a certain organism or device by ascribing to it certain explicit symbolic *rules and representations,* or whether we should simply describe its *dispositions to respond* (i.e. its intrinsic input-output function) in any way we find convenient (including appealing to exactly the same rules, but without the assumption that they are represented anywhere other than in the theorist's notebook, and consequently without any concern for strong equivalence), is a central philosophical issue in the foundations of cognitive science. To some extent the issue is a conceptual one and relates to the question of whether psychology has a special (and less than objective) status among the sciences (for example, Putnam 1978 has argued that cognitive theories are necessarily "interest relative" rather than being empirically determinate – see, however, Chomsky, this issue). In addition to this more general question, however, there is also a straightforward empirical side to this issue, to which I now turn.

Elsewhere (Pylyshyn 1978b) I have sketched some general conditions under which it would be reasonable to speak of behavior as being governed by rules and representations. The general position is that whenever behavior is sufficiently plastic and stimulus-independent, we can at least assume that it is somehow mediated by internal functional states. Such states may be further viewed as representational, or epistemic, if certain other empirical conditions hold. For example, we would describe the behavior as being governed by representations and rules if the relation between environmental events and subsequent behavior, or the relations among functional states themselves, could be shown to be, among other things, a) arbitrary with respect to natural laws, b) informationally plastic, or c) functionally transparent. We elaborate briefly on these conditions below.

The relation between an event and the ensuing behavior would be said to be arbitrary if there were no *necessary* intrinsic connection between them. More precisely, a relation between an environmental event, viewed in a certain way, and a behavioral act, is said to be arbitrary if, *under that particular description of the event and the behavior,* the relation does not instantiate a natural law. For example, there can be no nomological law relating what someone says to you and what you will do, since the latter depends on such things as what you believe, what inferences you draw, what your goals are (which might include obeying totally arbitrary conventions), and, perhaps even more important, how you perceive the event (what you take it to be an instance of). Since all physical events are intrinsically ambiguous, in the sense that they can be seen in very many different ways, each of which could lead to very different behavior, the nonlawfulness of the link between these events and subsequent behavior seems clear. Systematic but nonlawlike relations among functional states are generally attributed to the operation of *rules* rather than natural laws.

The condition that I referred to as informational plasticity will play an important role in later discussion. For the present I introduce it to suggest that epistemic mediation is implicated whenever the relation between environmental events and behavior `can be radically, yet systematically, varied by a wide range of conditions that need have no more in common than that they provide certain information, or that they allow the organism to infer that a certain state of affairs holds – perhaps one which, if it were actually perceived, would also produce the same behavior. For example, seeing that the building you are in is on fire, smelling smoke coming in through the ventilation duct, or being told by telephone that the buildings is on fire, can all lead to similar behavior, and this behavior might be radically different if you believed yourself to be performing in a play at the time.

The third condition, that of transparency of relations among representations, is, in a sense, the converse of the second condition. Whereas informational plasticity reflects the susceptibility of the process between stimulus and response to cognitive influences, the transparency condition reflects the multiple availability of rules governing relations among representational states. Wherever quite different processes appear to use the same set of rules, we have a *prima facie* reason for believing that there is a single explicit representation of the rules, or at least a common shared subprocess, rather than independent identical multiple processes. The case can be made even stronger, however, if it is found that whenever the rules appear to change in the context of one process, the other processes also appear to change in a predictable way. In that case we would say (borrowing a term from computer science) that the rules were being used in an "interpretive" or transparent, rather than a "compiled" or opaque mode.

Such seems to be the case with grammatical rules, since we appear to have multiple access to these rules. We appeal to them to account for production, for comprehension, and for linguistic judgments. Since these three functions must be rather thoroughly coordinated (e.g., a rule first available only in comprehension soon becomes effective in the other two functions), it seems a reasonable view that they are explicitly represented and available as a symbolic code (as I have argued elsewhere; Pylyshyn 1976). In other words, there are cases, such as grammar, in which rules can be used not only within some specific function, but in some circumstances they can even be referred to or *mentioned* by other parts of the system. These cases argue even more strongly that the system could not simply be behaving *as if* it used rules, but must in fact have access to a symbolic encoding of the rules. [Other arguments for the representational theory of mind are offered by Fodor (1975; and this issue) and Chomsky (this issue).]

To summarize, then, a *cognitive process* is distinguished from a process in general inasmuch as it models mental events, rather than merely simulating some behavior. This means that the states of the process are representational, and that this representational content is hypothesized to be the same as the content of the mental states (i.e. tacit knowledge, goals, beliefs) being modelled. Thus the computational states do not *represent* biological states, unless by this we mean to suggest that the person being modelled is thinking about biology. The process is, however, realized or carried out by the fixed functional capacities provided by the biological substrate. These functional capacities are called the functional architecture. The decision as to whether or not a particular function should be explained by positing a cognitive process – i.e., by appeal to rules and representations – rests on certain empirical considerations. One sufficient (though not necessary) condition for a function; being determined by representations is that it be influencible by beliefs and goals – i.e., that it be cognitively penetrable. Hence only cognitively impenetrable functions constitute the fixed functional capacities, or the functional architecture, out of which cognitive processes are composed. We shall elaborate upon these conditions further in sections 8, 9, and 10.

7. Computational models and explanations

We return to the theme raised in section 4, namely the question of how a computational model, in the form of a symbol-processing algorithm, can be used to provide an explanation of cognitive phenomena. In general, what is commonly referred to as a model can be used in various ways in explanation. These range from using it merely as an illustration or metaphorical object, to using it as a constitutive part of the theory. Computational models have been particularly free in the way the explanatory burden has been shared between the model and the theorist. There has, in other words, been very little constraint on what a theorist is warranted to say about an algorithm. But if the algorithm is to be taken as a literal description of a mental process, then much stronger constraints must be imposed on how it is used in the explanatory account. In particular, the appeal to the algorithm must be *principled.*

Notice that the parallel problem is not as serious in the case of mathematical models in psychology, or models in physics. Here a clear separation is made among the fixed universal constants, empirical parameters estimated from the data, variables, and functional forms. Although in formulating the model the theorist is relatively free to choose a function (subject to available mathematical notation and technique, as well as to certain other intrinsic constraints, such as the general prevalence of linear and inverse-square law effects), thereafter the theorist is, to a considerable extent, accountable for the way estimates of parameter values are revised in response to observations. This is done by monitoring the degrees of freedom in the model. To prevent a theorist from arbitrarily revising the cognitive model, a notion similar to this accountability is needed. While I don't see an exact analogue to the degrees-of-freedom ledger emerging in cognitive science, I think there is an analogue to the technique of factoring the sources of variability in the system. Part of the requirement that the appeal to the computational model be principled, then, is the demand that various sources of the model's behavior be factored apart and independently justified. This is, in fact, an instance of an extremely general scientific maxim, namely, that a central goal of explanatory theories is to factor a set of phenomena, a problem, or a system into the most general and perspicuous components, principles, or subsystems.

Apart from the general application of this principle in, say, factoring apart knowledge of a domain, various problem-specific heuristics, and resource-limited performance mechanisms, there are two major classes of factors that a computational theory must explicitly recognize when appealing to an algorithmic model. I refer to these as a) the *functional architecture*, or the fixed capacities of the system, and b) the *functional requisites*, or the effectiveness principles relating the system to the task. Partitioning the explanatory appeal to the computational model in this way places two special burdens on the theory-building enterprise. The first is that the basic functional mechanisms or computational building-blocks out of which the model is constructed must be independently justified. The second is that one must be able to specify the characteristics of the model and the task in virtue of which the system is capable of carrying out that task or producing the behavior we are explaining. In other words, both the underlying architecture of the computational model, and the properties of the algorithm or task that enables the system to successfully carry it out, must be explicitly addressed in the explanatory story, and both must be independently justified.

The first factor, regarding assumptions about the underlying functional architecture, is the primary concern of most of the remainder of this paper, and hence a detailed discussion will be postponed until the next section. What is at issue is that the success of an algorithm in accounting for a certain domain of behavior must not be due merely to some quite fortuitous property of a particular computational architecture, which itself cannot be independently justified. For example (Winston 1975), some interest was once aroused by the fact that a certain artificial intelligence "blocks world" vision system [based on Guzman's (1968) program SEE] seemed to exhibit an effect very similar to the Muller-Lyer illusion (i.e., it took a line with arrows on its ends like ⟷ to be shorter than a line with forks on its ends like ≻—≺), and thus to provide a possible account of this illusion. This particular effect was due to the fact that the system's line-recognizing procedure used a diameter-limited scanner, which scanned the line looking for evidence of certain types of terminating vertices. Evidence for an "arrow" vertex accumulates sooner in the scan than does evidence for a "fork" vertex, because the secondary arrow lines enter the scan region even before the actual point of intersection does. Thus the system recognizes the end of the line segment earlier in scanning toward it when the segment terminates with an arrow than when it terminates with a fork, hence yielding a lower estimate of its length in the former case. By adjusting the diameter of the scan region, this method can account for some quantitative properties of the illusion reasonably well. Now in this example it is very clear that the phenomenon is associated with the assumption of a diameter-limited line scan. Thus, whether this particular account of the illusion is classed as a valid serendipitous finding or merely a fortuitous coincidence depends very much on whether the assumption concerning the mechanism, or the architectural property of the detector, can survive empirical scrutiny.

Although this example may seem fanciful, it is nonetheless the case that *every* computational model, if it is taken literally, must make assumptions about the mechanism. These are frequently not taken to be empirical hypotheses, since it can easily escape our notice that some of the system's performance is attributable to certain assumed architectural features. Our experience with a rather narrow range of possible computational architectures can blind us to the fact that our algorithms are relative to such architectures (as we shall see in section 8). Furthermore, when the assumptions are exposed and analysed, they do not always seem so plausible. A particular case worth mentioning concerns the architectural assumptions underlying the successful use of lists and two-dimensional matrices to model aspects of reasoning and the spatial character of images. We shall examine these briefly later.

The second class of factors – those relevant to answering the question of why the algorithm exhibits the ability it does (or the ability claimed) – represents a further application of the injunction that explanatory appeals to a computational model must be principled. It is also an attempt to deal with the (often justified) criticism that a hypothesized algorithm is *ad hoc* in the sense that there is no independent reason for it to be the way it is, as opposed to some other way, other than because it was designed to duplicate a certain set of observations. Earlier I characterized design decisions, motivated solely by the need to account for certain observations, as being equivalent to fixing the value of a free empirical parameter. Such decisions are unmotivated by general principles or constraints on processes, except perhaps ones that are stated in terms of properties of the represented domain, which, we have argued, makes them descriptive but not explanatory. Explanatory principles must characterize the operation of a system in terms of endogenous structural or functional properties of that system.

The nature of the requirement, that we be able to state the principles in virtue of which the algorithm is capable of achieving the claimed skill, is best illustrated by the following example, based on the work of Shimon Ullman (1979). It is known that the shape of a moving object can be perceived even in highly impoverished circumstances – such as when the presentation consists of a silhouette projection of a rotating unfamiliar wire shape, or even random unconnected elements on the transparent surface of the rotating forms. This perceptual ability was first studied by Wallach and O'Connell (1953) under the title "kinetic depth effect." Now, suppose someone designed an algorithm (perhaps one using statistical methods) that recognized shape successfully in over ninety percent of the cases presented to it. Could we consider this algorithm to be a model of the underlying process? If we answer this question in the affirmative, we must then ask the further question: In virtue of what principle does it have the claimed competence to perceive form from motion? In fact, it turns out that the appropriate order to ask these two questions is the opposite to the one given above. For it is only by trying to discern the principle governing the alleged ability that we can be sure that the system does indeed have that ability – that it does more than to "account for variance" or to mimic some segment of the behavior without embodying the competence in question.

The problem of inferring shape from motion is not solvable in general without bringing in additional constraints, because three-dimensional shape is underdetermined by its two-dimensional orthographic projection. But the imposition of extrinsic constraints on this inference should be, as we have emphasized, principled. In particular, it should be such as to guarantee a unique solution in all cases where a unique interpretation occurs perceptually. Ullman (1979) proposed that the constraint be viewed as a warranted assumption made by the interpretation scheme. He calls it the rigidity assumption, because it enjoins the system to interpret the moving elements in the two-dimensional projection as originating from a rigid body in motion, and to fail if such an interpretation is not possible (i.e., as a first approximation, not to produce an interpretation of the elements as belonging to an elastic or fluid medium). This is analogous to an assumption of grammaticality made in interpreting natural language. In that case the system would be asked to interpret a string as an instance of a structure generated by the grammar, and to fail if such an interpretation is not possible. Of course, in both cases a more elaborate model might produce analyses of the deviant cases as well (i.e. nonrigid and nongrammatical interpretations). However, this would not be done by abandoning the assumption completely, but rather by considering systematic departures from the strict form of the assumption (even ungrammatical sentences must be analysed in terms of grammatical rules, rather than simply in terms of such considerations as what is usually true of the referents of the terms, otherwise we could not understand a sentence about anything unexpected).

The rigidity assumption is warranted by two further findings. The first is the mathematical result (called the *structure from motion theorem*), showing that a set of four noncoplanar points in a rigid configuration is uniquely determined by three distinct orthographic views. The second finding is that the interpretation is in fact computable by a reasonable locally-parallel procedure, given the sorts of data available to the human visual system, and under approximately those conditions in which people do perceive three-dimensional shape.

The lesson of this example is that by first understanding what the demands of the task are, it is possible to specify constraints that must be met by any algorithm capable of successfully carrying out that task. This, in turn, makes it possible to answer the question *why* a certain algorithm has the claimed ability. Without this extra stage of theoretical analysis we would be unable to say *why* some *particular* algorithm appeared to work on the set of problems on which we tried it. What is even more serious, however, is that we could not even say, with any degree of precision or confidence, what the class of tasks was that the algorithm *could* handle. While this might not be considered a shortcoming if we were merely concerned to account for experimental variance (i.e. for observational adequacy), it would be a serious defect if our goal was to provide a theoretical *explanation* of some domain of competence (i.e., if we were concerned with explanatory adequacy). If we know neither what the scope of a model's abilities is, nor what the principles are in virtue of which it behaves as it does, we do not have the basis for explanation.

To develop the analogous point with respect to the other class of requirements on the explanatory use of computational models (viz., that the functional architecture be separately constrained), we shall first examine the notion of functional architecture itself.

8. The influence of architecture and notation on processes

Computation is generally understood to be completely independent of the particular way it is physically realized. After all, a program can be executed by an unlimited variety of quite different physical devices, operating on quite different physical principles and in radically different physical media, such as mechanical, optical, acoustical, fluid, or any other conceivable substance (even including a group of trained pigeons!). On the other hand, the way in which the device functions is critical in determining whether it is capable of executing an algorithm. The design of a physical system that can function as a computer is no simple matter. But in view of the unboundedness of the variety of physical forms it can take, one might well ask what it is about the structure of the device, or class of devices, that makes it a computer? To answer this question we must recognize a level of description of the device intermediate between its description as a physical system (governed by the appropriate physical laws that determine its physical state transitions) and its description as a representational system (in which its behavior is governed by the rules and representations it embodies). This is the level we have been calling the functional architecture.

This level of description lies somewhere between the physical and the representational, in the sense that, unlike the physical description, it is independent of the particular physical laws that characterize any one physical realization of the system, and, unlike the usual algorithmic or rules-plus-representations description, it is an uninterpreted rule schema that can be exploited, by an appropriate choice of initial expression and interpretation scheme, to actually carry out some intended algorithm. It thus serves as the interface between the two. The physics of the device, together with a specified mapping from classes of physical states onto expressions, defines the functional architecture. In the computer case we can then view the role of such utility software as assemblers, loaders, compilers, interpreters, and operating systems as providing various realizations of this mapping, and hence as defining (or emulating) different functional architectures.

The notion of functional architecture has a special role to play in computer science, where it is sometimes referred to as "the architecture of the underlying virtual machine." When writing programs for some particular computer, programmers only have the resources provided by some particular programming language available to them. They are (to a first approximation – neglecting such practical factors as cost and resource limits) unconcerned about how the real physical device operates at the hardware level, since what they can do with the system is fixed by the functional specification of the language. This specification consists of the sorts of things that are contained in the language user's manual – e.g., a list of the available operations and what they do, restrictions on how the operations can be put together, how the contents of memory can be accessed, how arguments are to be passed to functions, how control is transferred, and so on. This functional specification defines the programmer's *virtual machine,* which a programmer cannot distinguish from a real machine without approaching the device in a quite different mode (e.g., looking in the systems manual or examining the switches and lights on the console). As far as the programmer is concerned the device may well be wired to function exactly as the language manual specifies (indeed, contemporary computers are frequently designed to execute programs in LISP or PASCAL or some other high-level programming language at very nearly the level of real hardware). Thus, even though there may in fact be several layers of program interpretation between the programmer and the actual hardware, only the properties of the virtual machine architecture are relevant to the user, because that is the only level to which the user has uniform access (i.e., any changes to the machine's functioning can be made only by utilizing the facilities defined by this virtual machine).

For the cognitive scientist a similar distinction between the functional architecture of the "cognitive virtual machine" and the mental algorithm is important, though the principle for distinguishing the two must be stated in a slightly different form, since it will ultimately depend on empirical criteria. The functional architecture is, by assumption, that part of the system that is fixed in a certain respect (which we shall specify in the next section when we examine the notion of cognitive architecture more closely), and that is also, hopefully, universal to the species. Mental algorithms are viewed as being executed by this functional architecture. In view of the fact that a valid cognitive model must execute the *same* algorithms as those carried out by subjects, and in view of the fact that (as we shall see below) *which* algorithms can be carried out (keeping in mind that algorithms are individuated according to the criterion of strong equivalence) depends on the functional architecture of the device, and furthermore, in view of the fact that electronic computers clearly have a very different functional architecture from that of minds, we would expect that in constructing a computer model the mental architecture will first have to be *emulated* (i.e. itself modelled) before the mental algorithm can be implemented.

Consider, for example, how we would go about using a computational system as a cognitive model. First of all, in order to describe an algorithm so it can be viewed as a literal model of a cognitive process, we must present it in some standard or canonical form or notation. Typically this means formulating it as a program in some programming language, but it might also include graphical presentation (as a flowchart) or even a discursive natural language description. Now, what is typically overlooked when we do this is the extent to which the class of algorithms that can even be considered is conditioned by the assumptions we make regarding what basic operations are possible, how these may interact, how operations are sequenced, what data structures are possible, and so on. Such assumptions are an intrinsic part of our choice of descriptive formalism, since the latter defines what we are calling the functional architecture of the system.

What is remarkable is that the range of computer architectures available for our consideration is extremely narrow, compared with what could in principle be considered. Virtually all the widely available architectures are basically of the Von Neumann type, or closely related to it. This goes for both hardware (which uses serially organized, stored-list programs and location-addressable memory) and software (see a discussion of the latter in Backus 1978). Because our experience has been with such a rather narrow range of architectures, we tend to associate the notion of computation, and hence of algorithm, with the class of algorithms that can be realized by architectures in this limited class. For example, we tend to think of flow diagrams as a neutral way of exhibiting algorithms. This is the idea behind the TOTE unit of Miller, Galanter, and Pribram (1960). But flow diagrams (and TOTE units) are totally inappropriate as a way of characterizing algorithms implemented on unconventional (non-Von-Neumann) architectures – such as, fōr example, the less familiar architecture of production systems, Planner-like languages, or predicate calculus-based programming systems like PROLOG (see Bobrow and Raphael 1974 for a discussion of some of these languages). If we use the criterion of strong equivalence (even just the psychologically important subset of this equivalence relation, which I referred to as complexity-equivalence) to individuate algorithms, we will find that different architectures are in general not capable of executing strongly equivalent algorithms.

This point is best illustrated by considering examples of several simple architectures. The most primitive machine architecture is no doubt the original binary-coded Turing machine introduced by Turing (1936). Although this machine is universal, in the sense that it can be programmed to compute any computable function, anyone who has tried to write procedures for it will attest to the fact that most computations are extremely complex. More importantly, however, the complexity varies with such things as the task and the nature of the input in ways that are quite different from the case of machines with a more conventional architecture. For example, the number of basic steps required to look up a string of symbols in such a Turing machine increases as the square of the number of strings stored. On the other hand, in what is called a register architecture (in which retrieving a symbol by name or by "reference" is a primitive operation) the time complexity can, under certain conditions, be made independent of the number of strings stored. A register architecture can execute certain algorithms (e.g. the hash-coding lookup algorithm) that are impossible in the Turing machine – in spite of the fact that the Turing machine can be made to be weakly equivalent to this algorithm. In other words, it can compute the same lookup *function*, but not with the same complexity profile, and hence not by using an algorithm that is complexity-equivalent to the hash-coding algorithm. Of course, it could be made to compute the function by simulating the individual steps of the register machine's algorithm, but in that case the Turing machine would be *emulating* the architecture of the register machine and executing the algorithm in the emulated architecture, a very different matter from computing it directly by the Turing machine. The distinction between executing an algorithm and emulating a functional architecture is crucial to cognitive science, as we have already remarked, because it relates directly to the question of which aspects of the computation can be taken literally as part of the model, and which aspects are mere technical implementation details, necessitated by the fact that at the level of actual hardware, production-model electronic computers have a functional architecture different from that of brains.

Examples of the architecture-specificity of algorithms could be easily multiplied. For example, a register machine that has arithmetic operations and predicates among its primitive operations (and hence can use numerals as names – or, as they are more frequently called, "addresses") makes a variety of additional algorithms possible, including binary search (in which the set of remaining options is reduced by a fraction with each comparison, as in the game "twenty questions"). The existence of arithmetic primitives as part of the architecture also means that it is possible to specify a total ordering on names, and hence to primitively partition certain search spaces (as in an n-dimensional matrix data structue), so that search can be confined within a region while other regions are literally not considered – items in those regions are not even checked and discarded, as they would have to be if they were merely part of an unordered set of items. Such algorithms could not be implemented on a Turing machine architecture.

As we go to more unusual architectures, other algorithms – with quite different complexity profiles – become possible. For example, Scott Fahlman (1979) has proposed a design for an architecture (realizable only with unconventional hardware) that computes set intersection as a primitive operation (in time independent of set size). He argues that many otherwise complex combinatorial algorithms required for symbolic pattern recognition (i.e., for access to stored data through multiple descriptions) become simple in such an architecture. In other words, because this architecture allows interesting new classes of algorithms to be implemented, the locus of the difficulty of giving a computational account of a certain task domain is dramatically shifted. Fahlman also argues that the resulting complexity profiles of certain memory-retrieval processes in this architecture are more like those of "natural intelligence." Although he does not use the methodology of the psychological laboratory (viz. reaction times), the goal is similar. Along the same lines, Mitch Marcus (1977) has also proposed certain general architectural features of the processor associated with grammatical analysis, which have the interesting consequence that they provide a principled account of certain linguistic universals in terms of general architectural constraints.

As we have already remarked, even when information-processing theorists make no claims about the form of the functional architecture, and attempt merely to develop models of certain cognitive processes, they cannot in fact avoid making certain tacit assumptions about the underlying architecture. Furthermore, the implicit adoption of some particular architecture carries with it certain further assumptions about the nature of mental functions. To see exactly what is being further assumed tacitly when a certain architecture is adopted, it is useful to look at properties of the architecture in a more abstract way, and to ask what it is about it that makes it possible to carry out certain functions in certain ways. A fruitful way to view this question is in terms of the formal or mathematical *type* of the primitive operations built into the virtual machine. For example, the primitive relation that we call *reference* (or the operation of retrieving a symbol given another symbol that serves as its name) is of the formal type *asymmetric*. It can be used to define operations for computing other relations such as, for example, the operation we might call AFTER, which computes a relation that is irreflexive, antisymmetric, transitive, and connected over some specified set of names. Such a syntactic operation in the functional architecture can be freely interpreted, at the level of interpreted rules, as any relation of the same formal type, for example the relation of being of higher rank, or older, or being a superset of (a relation frequently used in semantic hierarchies known as ISA trees). In making such an interpretation we automatically inherit the formal properties of these built-in primitive relations. Thus we do not need to ***explicitly represent*** properties such as the reflexivity, symmetry, transitivity, noncyclicality, and so on, of the relation in the interpreted domain – i.e., we do not need to represent symbolically a rule such as "If *X* is older than *Y*, and *Y* is older than Z, then *X* is older than Z," if we have chosen to represent "older" by a primitive relation of the architecture that is of the appropriate formal type.

In an architecture that has what we call an "arithmetic unit" (really only a set of functional properties that are useful for, among other things, representing the syntactic rules relating numerals under certain mathematically interpretable operations), as well as an ordering relation over symbols, there is an even richer set of formal types to exploit in constructing an interpreted system of rules. For example, certain formal properties of the Peano axioms for arithmetic, as well as the metric axioms, can be modelled in terms of these primitive "arithmetic" relations. This means that, in a sense, certain aspects of metric scales are available in such an architecture. In fact, since the axioms of Euclidean geometry have a model in the real numbers, such an architecture allows us to choose an interpretation scheme that makes geometric properties available without the need to represent geometric axioms symbolically (e.g., if we interpret pairs of numerals as locations of points in space and use the "arithmetic" operations to define distances and movements through this space). If we interpret primitive relations in this way, a variety of spatial and metric properties can be represented, changed, and their consequences inferred *without the need for symbolic computation*.

This sort of exploitation of the functional architecture of computational systems is central to computer science, as we have already noted. From the point of view of cognitive science, however, it is not only important to choose functional properties of the computational architecture in such a way as to accomplish certain tasks efficiently, but it is equally important to be explicit about *why* it works, and to justify these crucial properties independently. That is, it is important for the use of computational models in an explanatory mode, rather than simply a performance mode, that we not take certain architectural features for granted simply because they happen to be available in our computer language. We must first explicitly acknowledge that certain noncomputational properties originate with certain assumed properties of the functional architecture, and then we must attempt to empirically motivate and justify such assumptions. Otherwise important features of our model may be left resting on adventitious and unmotivated assumptions.

For example, people have occasionally suggested that subjects do not need to have knowledge of concepts such as, say, transitivity, in making certain inferences, as in the three-term series problems ("John is taller than Mary and John is shorter than Fred. Who is tallest?"), because all they have to do is arrange the three items in order (either in a list or in an image) and read the answer off. But, of course, the fact that one can solve the problem this way does not entail that tacit knowledge of formal properties (e.g. transitivity) of the relation "taller than" is not needed, since the decision to represent "taller" by something like "further on the list" must have been based on the implicit recognition that the two relations were of the same formal type (a list would not, for example, have been suitable to represent the relation "is married to"). Furthermore, while ordering three names in a list and then examining the list for the position of a particular name may seem straightforward and free from logical deduction, a little thought will show that the ability to carry out this operation mentally, as distinct from physically, presupposes a great deal about the available primitive mental operations. For example, in the mental case, if we have the items *A*, *B*, and *C*, and we place *A* and *B* in a certain order and then add *C* next in the sequence, we must assume (in this example) that: a) placing *C* next to *B* leaves the relation between *A* and *B* unchanged, and b) the relation of *A* to *C* (with *B* between them) is the same with respect to the relevant represented aspect (e.g. tallness) as that between *A* and *B*. But such assumptions are justifiable only if the agent in question implicitly understands the logical properties of *succession*. Consequently, even if list operations are part of the functional architecture (which, as we saw earlier, assumes that the architecture incorporates primitive operations of the appropriate formal type), one is still not entitled to assume that the use of this capacity requires no further appeal to tacit knowledge of logical constructs. Furthermore, as Piaget has suggested, it may well be that either the availability of the relevant primitive operation *or* the tacit knowledge relevant to its appropriate use may develop with the maturation of the organism. If this were so, we might then wish to represent a logical property like transitivity as an explicit logical rule, rather than building it into the architecture [see Brainerd: "The Stage Question in Cognitive-Developmental Theory" *BBS* 1(2) 1978].

To take another timely example, matrix data structures have frequently been used to represent the spatial properties of images (e.g. Kosslyn and Schwartz 1977; Funt 1977). This is a convenient way to represent spatial layout, partly because we tend to think of matrices in spatial terms anyway. In addition, however, this structure seems to make certain consequences available without any apparent need for certain deductive steps involving reference to knowledge of geometry. For example, when we represent the locations of imagined places in our model by filling in cells of a matrix, we can "read off" facts such as which places are adjacent, which places are "left of" or "right of" or "above" or "below" a given place, and which places are "in between" a given pair of places. Furthermore, when a particular object is moved to a new place, its spatial relations to other places need not be recomputed. In an important sense this is implicit in the data structure. Such properties make the matrix a much more natural representation than, say, a list of assertions specifying the shape of objects and their locations relative to other objects.

But, as in the case of the apparently noninferential consequences of using lists, such properties of matrices arise from the existence of certain formal properties of particular functional architectures. These properties would not, for instance, be available in a Turing machine architecture. In order for a matrix data structure with the desired properties to be realizable, the architecture must provide at least the primitive capacity to address the content of a representation by *place* – i.e., it must be possible to *name* a location and to ask for the content of a named location. This itself may require, for instance, what is known as a register architecture (or some other kind of location-addressable store). Furthermore, it must be possible in this architecture to primitively generate the names of places adjacent to a given place (i.e., it must be possible to do this without appealing to other representations or to tacit knowledge of geometry or anything else that would involve intermediate inferential steps). This is necessary in order to allow the representation to be "scanned." In addition there must be primitive predicates that, when applied to names, evaluate the relative directions of places corresponding to those names (e.g. two-place predicates such as "RIGHT-OF" must be primitive in the architecture). This, in turn, implies that there are at least two independent, implicit, total orderings over the set of names. In addition, if the relative distance between places is to be significant in this representation, then there might be further primitive operations that can be applied to place names so as to evaluate, say, relative size (e.g. the predicate "LARGER-THAN").

This whole array of formal properties is available in all common computer architectures, because they all use numerical expressions for register (i.e. place) names and have built-in primitive arithmetic operations. But these are part of such architectures for reasons that have nothing to do with the needs of cognitive science. When these features are exploited in building cognitive models, we are tacitly assuming that such operations are part of the functional architecture of the mind – an assumption that clearly needs to be justified. Arguments have rarely been provided for any such proposals. The only suggestions of an argument for such architectural features that I have seen are due to Piaget, who has been concerned with abstract formal characteristics of cognition, and Brouwer (1964) and Nicod (1970), who, for quite different reasons, proposed that *succession* be viewed as a cognitive primitive.

The general point concerning the intimate relation between virtual machine architecture and process is exactly the same as the observation that different notations for a formal system can lead to different expressive powers and even different axioms for the same system. For example, if we use conventional notation for algebraic expressions, we need to explicitly state facts about the associativity and precedence of arithmetic operators, whereas in Polish notation we do not need to represent such properties explicitly, because, in a sense, they are implicit in the notation. Similarly, propositional logic normally contains axioms for commutativity and for the complementarity expressed in de Morgan's principles. However, if we use the disjunctive normal form for logical expressions, such axioms need not be stated, because, as in the algebraic case, they are also implicit in the notation. Mechanical theorem-proving exploits a variety of such intrinsic formal properties of both the notation (e.g. using the disjunctive normal form) and the virtual machine architecture. For example, such theorem-provers can be made much more efficient and natural in certain domains by representing sets of propositions in the form of "semantic nets," which exploit the formal properties of the reference relation available in typical register machines (i.e in the usual Von Neumann architectures).

From the point of view of cognitive science, the notation we choose is important for reasons that go beyond questions of efficiency or naturalness. Because we claim that behavior is governed by symbolically encoded rules and representations, the exact format or notation that is used to encode these representations consitutes a claim about mental structures, and hence an empirical claim. Formats, like functional architectures, become empirically decidable in principle if we admit the relevance of the criterion of strong equivalence, and if we develop appropriate methodologies based on this criterion.

It is important to recognize that the greater the number of formal properties built into a notation, or the greater the number of primitively fixed formal properties of the functional architecture that must be exploited, the weaker the expressive power, and the more constrained will be the resulting computational system. This is because we no longer have the option of changing such properties at will. In choosing a particular notation or architecture, we are making a commitment concerning which aspects of the functions are to be viewed as the free parameters that are tailored to fit specific situations, and which are to be viewed as the fixed properties shared by all functions in the class of models that can be constructed using that notation. The more constrained a notation or architecture, the greater the explanatory power of resulting models. It provides a principled rationale for why the model takes one particular form, as opposed to other logically possible ones. Recall that the lack of such a rationale was one of the features that made some computational models *ad hoc*. One goal in developing explanatory cognitive models, then, would be to fix as many properties as possible by building them into the fixed functional architecture. Opposing this goal, however, is the need to account for the remarkable flexibility of human cognition. We shall see in section 10 that this character of cognition provides the strongest reason for attributing much of its manifested behavior to tacit knowledge of various kinds rather than to the sorts of fixed functional properties that have frequently been proposed.

9. Cognitive functional architecture

The architecture-algorithm distinction is central to the project of using computational models as part of an explanation of cognitive phenomena. The core notion of strong equivalence depends upon there being a common architecture among processes. Processes can only be compared if their grain or "level of aggregation" [to use Newell and Simon's (1972) phrase] is the same. In fact, even the formal semantic properties of programs, as developed by Scott and Strachey (1971) are relative to an abstract model of computation that, in effect, specifies an appropriate grain for the analysis. Furthermore, the privileged vocabulary claim (described in section 4) asserts that cognitive phenomena can be accounted for solely by appealing to the symbolic representations (i.e the algorithm and its associated data structures). Thus, any differences among such phenomena arise solely from the structure of these symbol systems – from the way component parts are put together. Thus, no distinctions among phenomena that we would classify as cognitive distinctions can be due to such things as differences in the way the primitive operations that constitute the algorithm themselves function. Furthermore, if differences in cognitive processing between individuals are to be explained within our computational framework, it is necessary that the basic functional architecture be universal.

Mental architecture can be viewed as consisting of just those functions or basic operations of mental processing that are themselves not given a process explanation. Thus they are functions instantiated in the biological medium. Unlike cognitive functions in general, they are, on the one hand, the primitive functions appealed to in characterizing cognition, and on the other hand, they are functions that are themselves explainable biologically, rather than in terms of rules and representations. Thus we see that the architecture-algorithm distinction parallels one we made earlier between functions symbolically computed. Since we gave, as two of the conditions for the appropriateness of the latter way of characterizing a function, that the relation between antecedent conditions and subsequent behavior be arbitrary and informationally plastic (or cognitively penetrable), these criteria will consequently be relevant to distinguishing between functions attributable to the architecture and functions attributable to the structure of the algorithm and the cognitive representations. This is our answer to Wittgenstein's "third man" puzzle. We do not need a regress of levels of interpreters, with each one interpreting the rules of the higher level and each in turn following its own rules of interpretation. Nor do we need to view our computational model as consisting of a cascade of "intentional instantiations," as Haugeland (1978) does. Only one uniform level of rules is followed, since only one symbolic level of *cognitively* interpreted representations and of cognitive process is involved. However complex the remaining functions are, they are considered to be instantiated by the underlying biological medium – i.e., they are part of the functional architecture. The reader will note a certain inevitable circularity in the above discussion of the relation between cognition and the architecture/algorithm boundary. On the one hand, cognitive phenomena are understood as those which, under the appropriate interpretation of observations, can be accounted for solely by examining the algorithm. On the other hand, the distinction between what is attributable to the algorithm and what is attributable to the functional architecture is to be decided on the basis of whether certain phenomena (namely cognitive ones) can be accounted for while keeping the architecture fixed. This circularity is not, however, vicious. The basic core intuitions concerning what constitutes a cognitive phenomenon will not have to be revised willy-nilly every time a new algorithm is formulated. While the domain of the theory will evolve gradually to accommodate the evolving theoretical system, yet at any time there will be phenomena clearly identifiable as cognitive. These are the ones that will adjudicate whether or not some function is legitimately viewed as part of the mental architecture that is to serve as the fixed primitive basis for constructing specific algorithms.

We may therefore summarize the conditions under which a specific function or behavioral property may be attributed to properties of the fixed functional architecture as follows: 1) If the form of the hypothetical function or the putative property can be systematically influenced by purely cognitive factors, such as changes in instructions or in the information-bearing aspects of the context, or any other condition that clearly leads to differences in goals, beliefs, interpretations, and so on; or 2) if we must postulate variations in the form of the hypothetical function or in the putative property in order to account for certain systematic differences in observations of cognitive phenomena; then such a function or property may *not* be attributed to properties of the fixed functional architecture (or the "medium"). Consequently, if it is not attributable to a property of the functional architecture, it must be given a cognitive account that appeals to the structure of the process and to the content and form of the representations. This cognitive process model itself will, as we noted earlier, have to acknowledge explicitly the contribution of the task demands in setting the goals and in accounting for the skill exhibited by the model.

Conditions (1) and (2) above are required not only because of our proprietary vocabulary hypothesis, but also by the arguments raised earlier regarding when it is appropriate to invoke rules and representations to explain observed behavior. Notice how closely conditions (1) relates to the informational plasticity criterion of rule-governed behavior. The informal notion of informational plasticity refers to the property of a system in virtue of which certain aspects of its behavior can be systematically altered by information, and hence by how the organism encodes stimuli or interprets events. We argued that the explanation of such behavior should appeal to rules and representations. Such a property is, of course, also one of the clearest signs that we are dealing with cognitive phenomena. Hence the proprietary vocabulary hypothesis and the representation-governed behavior arguments converge on the same principled distinction, which we take as defining the architecture/algorithm boundary.

Condition (2) can also be viewed in a way that brings out its relation to the distinction between automatic and attentional processes, which has attracted considerable interest recently (e.g. Posner and Snyder 1975; Schneider and Shiffrin 1977). Condition (2) entails the claim that functions attributable to the functional architecture can consume only constant resources. In other words, they must not include what are referred to as "attentional" processes. For any differences in resource use must arise from different execution traces of cognitive processes, or different algorithmic sequences. Hence functions that exhibit such differences do not qualify as part of the fixed architecture. I have discussed this issue in connection with Anderson's (1978) behavioral mimicry claim (Pylyshyn, 1979b), where I pointed out some of the consequences of allowing varying complexity (e.g. as indexed by reaction time) to be exhibited by primitive operations (or individual steps) in a process.

We thus conclude that a basic methodological distinguishing mark of functions that are part of the basic fixed functional architecture of the mind is that they cannot be influenced in their operation by what might be called informational or cognitive factors, nor do variations in their operation need to be posted in order to account for observed cognitive phenomena. Such functions and properties remain, to use a phrase that I have adopted in other contexts (e.g. Pylyshyn 1979a; 1979b), *cognitively impenetrable.*

The criterion of cognitive impenetrability serves as the litmus by which we can decide whether a function is a fixed, built-in, causally explainable, primitive operation or property in terms of which cognitive processes are to be described, or whether it will itself require a computational or process explanation. It is clear why such a criterion is needed. Without it one could not distinguish between a literal and a metaphorical appeal to a computational model. By providing a principled boundary between the "software" and the "hardware," or between functions that must be explained mentally (i.e. computationally), and those that can only be explained biologically, one can factor the explantion into the fixed biological components and the more variable symbolic or rule-governed components. Like the factoring of fixed universal constants from particular conditions specific to the case at hand, which occurs in all physical explanations, or the factoring of linguistic universals (also taken to represent fixed properties of mind), from language-specific rules and comprehension algorithms, which occurs in models of linguistic competence, such factoring is essential for accounts to achieve explanatory adequacy.

Although, as I have repeatedly stressed, the explanation of how the primitive properties of the functional architecture are realized will ultimately be given in biological or physical terms, this should not be interpreted as meaning either that such properties must be stated in a biological vocabulary, or that they are to be inferred from biological observations. It has sometimes been claimed (e.g. Anderson 1978; Palmer 1978) that only biological data could help us decide between certain theoretical proposals for fixed properties of the cognitive system – such as between analogical and propositional forms of representation. It should be clear from the current discussion of functional architecture that this is not so. Although the architecture represents a crucial interface between properties requiring a cognitive process account and those requiring a biological account, the actual description of this architecture is a functional one. It simply specifies the primitive functions or fixed symbol-manipulation operations of the cognitive system. Furthermore, the point of the cognitive impenetrability condition is to provide a purely functional methodology for deciding whether a putative property qualifies as belonging in the caregory of architecture or in the category of cognitive process.

Finally, before concluding this section I must again reiterate the very strong contrast between the position I have been describing and psychophysical dualism. According to the present view, two distinct types of explanation of human behavior are possible. One of these is naturalistic (i.e., it appeals directly to intrinsic properties of the organism) and the other cognitive (i.e., it appeals to internal representations). I have tried to sketch some general criteria under which each is appropriate. But for many people there still remains the uneasy question of why it should be the case that a certain class of systems (e.g. certain organisms and computers) admit of these two analyses, while other (perhaps equally complex) systems (e.g space vehicles) do not. Since the minutest details of the operation of all systems are clearly governed by causal physical laws, one might ask what distinguishes these two classes of system.

In its most general form this question runs into the problem of the intentionality of human cognition – a problem that is clearly beyond the scope of the present essay. This is the question of what it is about a mental representation or a thought that makes it a thought *about* some particular thing rather than another. For example, since the Löwenheim-Skolem theorem assures us that any consistent formal system has a model in the integers, any such formal system could be just as legitimately viewed as representing, say, the natural numbers as anything else we care to name. In his accompanying target article, Fodor (this issue) suggests that an essential methodological strategy of cognitive psychology is to factor away the intentionality question and develop models that are concerned primarily with coherence. Interesting questions still remain, even if we somehow factor away the difficult issue of intentionality from the issue of epistemic mediation. However, even in that case we can still ask what it is about certain systems that makes them candidates for an epistemic mediation account that appeals to representations and rules.

Among the exogenous factors to which we have already alluded is the human imperative to explain cognitive behaviors in terms that parallel the form in which we plan and conceive of our own behavior. Perhaps as a direct consequence of this it also seems that the appropriate regularities in our behavior are to be found when it is described in terms of what Kenneth Pike (1967) referred to as an *emic* as opposed to an *etic* taxonomy (i.e., intensional as opposed to extensional, conceptual as opposed to objective, or perceptual as opposed to physical). Of course this applies only when the phenomena to be explained are themselves cast in such terms – as they must be in the case of meaningful human behavior. Thus, to explain how the mouth and tongue move and how acoustical energy patterns are generated, we appeal to the taxonomy and laws of physics and perhaps biology, whereas to explain what someone was saying at the time and their choice of words, we must appeal to cognitive concepts and processes.

But even if we accept such reasons for turning to a representational account of human behavior, there still remains the question of what intrinsic properties humans and computers share that make behavior fitting such an account possible. Many years ago Kohler (1929) offered some suggestions on this question. He distinguished between processes that were determined by anatomically fixed, or as he called them, "topographical factors," and those that were determined by nonlocalizable or "dynamic factors" such as "forces and other factors inherent in the processes of the system." I believe that Kohler was on the right track in this analysis (which he used in arguing for certain differences between biological and technological systems). However, he had too limited a view of how topographic factors could operate. Perhaps because of the types of artifacts with which he was familiar, Kohler took the spatial distribution of constraints to be the primary focus of the distinction. What this fails to recognize, however, is that in certain systems (e.g. computers) the functional structure can still change radically, even when the topographical structure remains fixed. Functionally, the part-whole relation in this kind of system is such that global discontinuities in function are produced by appropriate local changes in structure. Such propagation of local changes to produce systematic global effects is what Kohler believed would require a different sort of system – one governed primarily by nonanatomical dynamic factors such as field effects [see also Puccetti & Dykes: "Sensory Cortex and the Mind-Brain Problem" *BBS* 1(3) 1978].

Although Kohler may even have been right in his analysis of the structural nature of biological as opposed to artifactual systems, the realization of computational processes in topographically fixed systems shows that this particular dimension of difference is not a logical prerequisite for representation-governed behavior. Rather, what seems necessary is just what was hinted at above – namely, that the fixed factors, regardless of whether they are due to the topography or to laws of interaction of forces within prescribed boundary conditions or to any other fixed constraints, enable a certain kind of radical second-order flexibility. That flexibility might be characterized in terms of the global alteration of function that can be effected through local changes, or in terms of the existence of instantaneous functional networks that can be varied over and above the fixed topographical network, or in some other way. I do not know of a satisfactory way of precisely specifying this function in abstract terms, but I believe that it amounts to the requirement that the functional architecture be universal in Turing's sense – i.e., that it be capable of computing any computable function or of simulating any Turing machine. Whatever the correct general specification of this function is, it must distinguish between the long-term, structurally fixed functions that we have called the functional architecture, and the instantaneously alterable functions that are necessary to sustain rule-governed behavior and to enable behavior to change radically in response to such transitory effects as the inferences that the system makes upon receipt of new information. Only a system that has this character can be a candidate for a rule-governed system. What further requirements it must meet beyond that I cannot say, but clearly this particular aspect of rule-governed behavior does not raise any problems for a materialist view of mind.

10. The psychological relevance of the notion of functional architecture

In this section I shall discuss some of the implications of the present view for the development of cognitive theory. In the next section I will consider the traditional importance of something like the notion of functional architecture in psychology and examine several "fixed functions" that might be viewed as proposed properties of such an architecture.

The analysis we have been giving can be viewed as setting out a particular program of research – namely that of designing a cognitive virtual machine, or rather a system or programming language having a functional architecture appropriate for implementing cognitive algorithms. Such a machine (which, of course, would have to be emulated on available computers) would, among other things, display the appropriate resource-limited constraints characteristic of human processing. For example, it might conceivably be possible to implement various algorithms on such a machine for carrying out essentially the same function but with different trade-offs. It might, for instance, be possible to have an algorithm that attended to (i.e. whose behavior depended on) a larger number of symbolic expressions, in exchange for requiring more steps (thus trading off speed and memory load), or one that took fewer steps but failed to attend to some of the potentially relevant data (thus perhaps exhibiting a speed/accuracy trade-off).

As we have already pointed out, such an architecture would also make possible the goal of designing algorithms that were strongly equivalent to ones used by humans. Computational complexity profiles of processes, as the latter ranged over various systematic changes in inputs, could simply be read off from properties of the execution of the algorithms such as the number of primitive steps they took in each case. Such complexity features would now be empirically significant, rather than merely incidental properties of the model. These could then be compared with various hypothesized empirical correlates of human processing complexity – such as reaction times, perhaps. In the ideal case every operation and every feature of algorithms implemented on such an architecture would constitute an empirical hypothesis, just as every theorem derivable from Newtonian axioms constitutes an empirical prediction. One would not say of an algorithm executable on this architecture that the mind cannot compute such a function since, by hypothesis, processing constraints such as resource limitations, or any other fixed universal limitation in processing capacity, would have been incorporated into the architecture. Thus *every* algorithm that could be executed on this virtual machine would now be considered a humanly possible cognitive process, just as any grammar generatable by an ideal universal grammar (in Chomsky's sense) is a possible structure of a human (or humanly accessible) language.

This fact itself has the interesting consequence that it eliminates a certain asymmetry between what appear to be two radically different ways in which the rules and representations in a model could be modified. On the one hand there are the changes that come about as a consequence of various "inputs" to the model. These produce only orderly and psychologically appropriate modifications in representational states. On the other hand there are the changes that come about when the programmer intervenes by adding or deleting rules or inserting new representations. Since in conventional programming systems these changes are not constrained in any way by psychological considerations, they could result in the most bizarre and humanly inaccessible algorithms being specified. With the functional architecture appropriately constrained, however, this distinction in large part disappears, since all and only cognitively possible (though for various reasons perhaps not actual) algorithms and representations are permitted, regardless of their genesis.

The most concerted effort at designing a cognitive virtual machine (or at least at developing design specifications for an appropriate functional architecture) is currently being pursued by Allen Newell. The present form of this design is called a "production system." There are some interesting reasons for this particular choice, which we shall not go into here. They relate, in part, to Newell's concern with modelling certain resource-consuming properties of computation, which in conventional systems are not visible to the user but remain a part of the "backstage" activity of the virtual machine – namely, the control structure itself. The reader is referred to Newell (1973; in press) for a technical discussion of these issues.

However, we should note that the view expressed here on the importance of the functional architecture does not commit one to the project of designing a complete cognitive virtual machine as the first step. Indeed, simply having in mind that every computational model commits one to implicit assumptions concerning the underlying architecture can keep one from making at least some of the more problematic assumptions. As one example of this approach, Pylyshyn, Elcock, Marmor, and Sander (1978) report some highly preliminary results of a research project aimed at understanding aspects of perceptual-motor coordination as these are involved in drawing diagrams and making inferences from what is seen. This project was primarily concerned with certain design issues that arise when one avoids certain technically simple, but in our view cognitively implausible, ways of dealing with the problem of representing spatial relations in a computational model, especially in the context of perceptual-motor coordination [see Gyr et al.: "Motor-Sensory Feedback and Geometry of Visual Space" *BBS* 2(1) 1979]. When experimental data concerning some particular human cognitive function were not available, we followed the strategy of adopting the least powerful mechanism we could devise to carry out that function.

For example, instead of maintaining a matrix structure as a global representation of the spatial layout, our system only used a qualitative representation of spatial relations, together with a limited number of pointers to specific places on the retina. Coordination with the motor system was accomplished through a minimal mechanism, which was able to maintain a "cross-modality" binding between visual and kinesthetic spaces for only two places. We thought of these as "fingers," whose position (when they were on the retina) could be seen, and whose location in proprioceptive space could continue to be sensed even when they were off the retina. Thus we had a primitive ability to hold on to off-retinal locations in order to glance back to them. This "two-finger" mechanism made possible a quite general drawing capability, which extended beyond the foveal region. The general strategy was to opt for *minimal mechanisms* wherever possible, even at the cost of a computationally awkward system, on the grounds that such mechanisms committed one to the weakest presuppositions about the underlying architecture, and hence ones that could easily be revised upward without the need to radically redesign the system. A number of such minimal mechanisms were proposed and are described in the Pylyshyn et al. (1978) paper. This is an instance of the application of the principle of "least committment" to the design project. Newell's production-system architecture proposal, referred to above, and Marr and Nishihara's (1977) minimal mechanism for rotating the principle axes of a three-dimensional model to bring it into congruence with a given two-dimensional retinal pattern, could both be viewed as examples of this strategy.

In addition to these more ambitious goals of applying the idea of a cognitive virtual machine directly to the design task, one can also appeal to it in deciding the soundness of certain proposals concerning the nature of cognitive processing. It is useful to have a general criterion for distinguishing among classes of mechanisms when these are proposed as components of a cognitive model or a cognitive explanation. For example, it is useful to have a principled basis for deciding whether a certain cognitive phenomenon ought to be viewed as arising from the nature of some analogue representational medium, as frequently claimed, or whether it will require an explanation in terms of the nature of the symbolic cognitive process itself. In the latter case the properties we observe might be seen as arising from tacit knowledge, rather than from the nature of the mechanism or medium, as we suggested earlier was the case for the "mental scanning" phenomena.

Recall that one of the features of analogues was that they were nonsymbolic and noncomputational. Analogue representations, as generally conceived, are not articulated symbolic expressions, and analogue processes are not viewed as rule-governed symbolic computations. They can thus be viewed as characteristics of the functional architecture. I believe that it is this quality of incorporating fixed constraints into the architecture, and therefore of weakening the expressive power and consequently increasing the explanatory value of the models, that makes analogue systems particularly attractive in cognitive science. It is not the fact that such systems may be more efficient or more natural, nor even that they may be continuous or holistic in some sense, that ought to be their attraction. None of these are essential properties of analogues: analogue processes don't have to be efficient, nor must they be continuous, as opposed to quantized, to retain their analogue character (e.g., imagine your favorite analogue model approximated by a discrete but finely grained quantization). What they do seem to require is that the formal property that characterizes their function be explainable as being instantiated or exhibited, rather than as being symbolically computed by the operation of rules on symbolic representations. Thus, whether some particular function should be explained as a noncomputational or as an analogue process is the same kind of question we raised earlier in this essay; namely, whether the appropriate explanation of some behavior is one that appeals to rules and representations or merely to primitive dispositions. An examination of a number of proposed analogue mechanisms suggests that, as formulated (i.e. in their simplest and most attractive form), they all exhibit some degree of cognitive penetration and hence are not eligible as part of the noncomputational function of the system. Examples such as those discussed by Pylyshyn (1979a; 1979c) considerably weaken the case for the existence of a large degree of nonsymbolic processing in cognition, at least of the kind that has frequently been proposed in discussions of, say, imagery phenomena. Every case of cognitive penetration is an instance in which we are forced to relax the hypothesized fixed functional constraints. While it will always be possible in principle to build very flexible inferential processes, which at various arbitrary points in their operation turn to highly constrained subsystems (as, for example, in the suggestion put forth by Kosslyn et al. 1979, for a way to retain the holistic analogue rotation in the face of the data I reported), it is much more problematic to justify such hybrid models. Such proposals work only by subverting the main motivation for positing such fixed architectural capacities as, say, the ability to "mentally rotate" an image – namely the greater explanatory power inherent in the weakest or least expressive system. In such hybrid models we would need to have strong independent reasons for retaining the (now redundant) constrained subprocesses.

11. The search for fixed architectural functions

If we take a broad view of the notion of functional architecture, we can recognize that proposals for components of such an architecture are not infrequent in psychological theorizing – in fact, they have often characterized the differences among psychological schools. For example, the dominant assumption, from the time of the British empiricists until about fifteen years ago (and even today in some quarters), was that the principal built-in functional capacity of organisms was their ability to form associative links. For this assumption to be predictive it was further necessary to specify the sorts of entities over which the associations could be formed (e.g., for behaviorists these had to be behaviorally defined), as well as conditions on the formation and evocation of these links (e.g. conditions such as contiguity, reinforcement, generalization gradients, and so on). Such a hypothetical capacity, like many of the more contemporary nonbehaviorist proposals [cf. Bindra: "How Adaptive Behavior is Produced" *BBS* 1(1) 1978], was intuitively appealing because it agreed with our informal observation of certain aggregate (i.e. statistical) regularities of behavior, as well as of behavior in certain highly controlled situations.

It is no accident that the controlled situations that were investigated from the conditioning perspective universally involved the suppression of what we have been calling cognitive factors. The experimental paradigm was always contrived in such a way that beliefs, goals, inferences, or interpretations were rendered irrelevant as much as possible. Furthermore, critical observations were invariably expressed as frequencies or probabilities, thus averaging over any remaining cognitive effects or strategies. However, cognitive factors inevitably still left their effect in the case of research involving human subjects. In those cases, as Brewer (1974) has eloquently argued, the most plausible explanation of human conditioning phenomena is one given in terms of change in belief. In other words, the most straightforward explanation of what reinforcers do is that they inform the subject of the contingent utilities. Thus, for example, the same effects can typically be produced by other ways of persuasively providing the same information (such as explaining to subjects the conditions under which they will or will not be shocked and backing up the story by showing them the wiring).

However, I don't doubt that there are ways of incorporating such results into the conditioning account (especially since individuation criteria for "stimulus," "response," or "reinforcer" are unspecified within the theory, allowing our informal "folk psychology" to come to our aid in describing the situation appropriately and thus smuggling knowledge of cognitive factors into the predictions). Yet, whatever approach one takes to explaining such phenomena, Brewer's review and analysis of a large number of studies makes it clear that even the simplest and most paradigmatic cases of conditioning in humans (e.g. avoidance conditioning of finger withdrawal to shock, or conditioning of the eyeblink reflex) exhibit cognitive penetration – i.e., they are radically, yet systematically, influenced by what the subjects believe (or by what they are told or shown). Thus even these simplest cases do not demonstrate that conditioning is a primitive function of the fixed architecture. As before (e.g. the mental rotation case mentioned earlier), attempts to retain this function as a primitive might well be possible at the cost of considerably weakening the claims that can be made about the role of conditioning – e.g., by relegating it to the role of one small element in a large cognitive process involving rules, representations, goals, inferences, and so on. But then the burden of proof falls on those who posit this mechanism to show that it is still necessary, given the flexibility of the remainder of the system that has to be posited in any case.

An intrinsic part of the conditioning account – and the only part that could conceivably explain novel behavior, is the notion of generalization. A more sophisticated contemporary version of the generalization gradient is the similarity space. Just as conditioning requires dimensions along which it can generalize to new stimuli, so some contemporary theories that appeal to prototypes require dimensions of similarity in order to account for the relations among prototypes as well as between prototypes and various exemplars or instances. Among those theoretical approaches that seek to avoid the complexities of inferential and problem-solving processes in accounting for such phenomena as recognition and classification (e.g. Rosch 1973), the most common proposal is the functional similarity space. Even Quine (1977) speaks of a biologically endowed "quality space." Though no account is given of how such a space might be realized (presumably neurophysiologically), or how the location of novel stimuli in the space is determined, it can still be viewed as a possible component of the functional architecture. Thus it is appropriate once again to inquire whether this view can be empirically sustained.

The first thing to notice is that such a view cannot be applied to stimuli such as sentences, since it is clear that there is an unbounded variety of ways in which sentences can relevantly differ – i.e., there is no finite set of dimensions or categories of comparison that can exhaustively locate the meaning of a sentence in a similarity space. While it seems likely that the same is true of visual patterns, the existence of at least some quantitative dimensions of similarity (e.g. size, orientation, color-distance) makes it not as clear as in the language case. On the other hand, the very strong interactions among such putative dimensions – demonstrated repeatedly by the Gestaltists – could also be cited against the view that these can be treated as orthogonal dimensions in a similarity space. One of the clearest and simplest demonstrations of the inadequacy of the similarity space view, however, comes from the work of one of the pioneers of the multidimensional scaling technique. Shepard (1964) showed that when stimuli varied along several dimensions, judgments of their similarity could yield ratings that conform to a Euclidean metric along any of the dimensions of variation, depending on what subjects are instructed to attend to. But when the subjects were not given specific attention instructions and were left to attend to whatever they wished, the resulting data failed to conform to any metric (Euclidean or nonEuclidean) in the number of dimensions along which the stimuli varied. Shepard concluded that subjects' noticing strategy determined the similarity structure of their judgements – and these were free to move from one possible similarity space to another at will. In other words, the similarity space function is itself cognitively penetrable.

Once again, one could probably get around this sort of counterexample by allowing a cognitive component to preanalyse the stimulus prior to the use of the similarity space. But as in the previous case, in which such added cognitive processes had to be posited, it is no longer clear what function the space would now be serving, other than as an *ad hoc* parameter to increase the precision of prediction. For, if we need a cognitive process to analyse and oversee the recognition, we already have a mechanism that, at least in principle, is capable of accounting for the similarity structure of the set of stimuli. Again the burden would be on those who posit such a fixed function to show exactly what principled role it plays (not just in providing similarity judgements, but as a general representational system).

The latter result is only one of a large number of cases in psychophysics in which attempts to posit fixed psychophysical functions have run into difficulty because they turned out to be cognitively penetrable. Almost any psychophysical judgement that requires the subject to attend selectively to certain specific aspects of the stimulus, while ignoring other aspects, is likely to be cognitively penetrable. One of the best known examples of a simple function that turned out to be cognitively penetrable is the simple sensory threshold. It was shown to be penetrable by subjects' beliefs concerning the utilities of alternative responses – a finding that generated considerable interest in the theory of signal detectability (a decision-theoretic and therefore cognitive analysis) as an alternative to the threshold function.

Similarly, it has been argued (e.g. Segall, Campbell, and Herskowits 1963) that visual phenomena such as the Muller-Lyer illusion are also penetrable, though not as directly as the threshold function. The argument is that in time the function responsible for the illusion can be influenced by cognitive experience (in particular by experience with large three-dimensional rectangular shapes such as buildings). The "new look" movement in perception in the 1950's (e.g. Bruner 1957) was also built upon the recognition that a great deal of the perceptual process is penetrable. Among the better known experimental results in cognitive psychology that could be viewed in this way are the various instances of cognitive penetration demonstrated in studies of selective attention. It seems as though every time one investigator has proposed an attention-selection filter of one sort, another investigator has found evidence that the operation of that filter was sensitive to aspects of the information that the filter was supposed to have eliminated, and that therefore must have been getting through (see Norman 1969 for a review of some of these results). That is why, in an early model of selective attention, Norman (1968) found it necessary to include a factor he called "pertinence," which affects the recognition threshold of the postfilter processing. As the name suggests, the introduction of the pertinence factor is nothing but an admission that such processing is cognitively penetrable.

The various proposals that go by the title of "direct perception," as developed by J. J. Gibson (1979) and others, can be viewed as proposals for functions that are part of the fixed functional architecture of the perceptual system. The denial by this school of any "epistemic mediation" in perception makes it particularly clear that they view functions, such as those that are said to "pick up information" concerning perceptual invariants, as being instantiated by the functional architecture. But such proposals have generally not withstood the test being proposed here: the detection of everything from distance information to one's grandmother appears to be cognitively penetrable.

Howard (1978) has shown that even the perception of so obvious a property of a display as the horizontality of a colored fluid in a transparent container is strongly influenced by knowledge of the relevant physical principle. In his studies, conservatively conducted using trick forms of three-dimensional photographs and motion pictures, with the method of random presentation, Howard made the startling discovery that over half the population of undergraduate subjects he tested were unable to recognize anomalous stimuli, despite the fact that some of the mistakenly classified pictures depicted deviations of fluid levels as much as thirty degrees off horizontal, and despite the fact that the perceptual discriminability of the orientations involved was clearly above threshold, as evidenced by other methods of assessment. For example, the same subjects who failed the horizontality test could readily report when the surface of the fluid was not parallel to shelves visible in the background. What was even more surprising and relevant to the present discussion was that postexperimental interviews, scored blindly by two independent judges, revealed that *every* subject who recognized all anomalous stimuli that were out by at least five degrees could clearly articulate the principle of fluid level invariance, whereas *no* subject who failed to recognize such stimuli as deviant gave even a hint of understanding the relevant principle. What is especially surprising in these studies was that evidence of knowledge of the relevant principle was obtainable by verbal probing. Usually the inference that tacit knowledge is involved is much less direct – as is the case for phonological rules and for at least some instances of syntactic rules.

In a similar vein I have argued for the cognitive penetrability of visual (or "imaginal") memory (Pylyshyn 1973; 1978c), of the proposed analogue function of "mental rotation" of images (Pylyshyn 1979a), and of the spatiality of images as inferred from "mental scanning" studies (Pylyshyn 1979c; also in section 5 above).

The point of these examples is not to suggest that all mental functions are cognitively penetrable. Indeed, I do not see how a computational view of cognition could be developed if that were so. (I don't mean that it would be impossible; only that it would require a rather different conception of functional architecture, and hence of algorithmic process.) The point is merely to suggest that many of the more constraining (and hence potentially more explanatory) proposals fail because of the flexibility of human cognition.

Of course, there are numerous examples of proposals that have not failed the test of penetrability (at least not yet). Most of these involve more peripheral functions, and most of them have been studied only relatively recently. For example, the work of Marr (1979) and associates provide some rather clear examples of complex processes that do not appear to be penetrable. These involve early visual processes (e.g. derivation of what Marr calls the "raw primal sketch" from incoming visual information, combining information from two retinal inputs to derive stereoptically encoded structures, derivation of certain textural and form information from retinally local patterns, and so on). At the motor end there has similarly been considerable success in demonstrating cognitively impenetrable functions (the earliest of which goes back to the work by Bernstein 1967). [See Roland, *BBS* 1(1) 1978.]

It is rather more difficult to point to examples of more central processes that are clear instances of impenetrable architectural functions. Perhaps this is because central cognitive processes are more like deductions – in which virtually any two concepts can be connected by some inference path under the appropriate circumstances. Thus I suspect that the semantic net proposals (including the organization of the lexicon in Morton's (1970) Logogen model) could be shown to be penetrable, though I know of no direct evidence bearing on this question at the moment. My own suspicions are that the functions that will be considered part of the functional architecture of such higher level processes as thinking and common sense reasoning will be of two distinct kinds. On the one hand there will be extremely primitive elementary symbol processing operations, though they are unlikely to be the sorts of operations found in contemporary serial digital computers. On the other hand there will be extremely abstract constraints on data structures and on the control system. Further, the primitive functions needed may be quite different in different parts of the cognitive system, though there may well be common resource allocation mechanisms (e.g. a common type of control structure).

The apparent lack of highly constrained functional properties of the sort that our folk psychology might lead us to expect should not surprise us. This is one of the areas where we are seeking deeper explanatory principles and hence where folk psychology is least likely to be of service. The comparable case in which some success has been achieved in seeking highly constrained and universal properties is in the case of linguistic universals or universal grammar. There it has become obvious that properties of this sort could only be found at extremely high levels of abstractness, and at a considerable deductive distance from our intuitions and observations. There is no reason to expect the situation to be any different in the case of other areas of cognition.

Furthermore, to expect that observations such as those associated with the study of mental imagery will provide a direct route to the functional architecture of mind is to vastly underestimate the flexibility of mental processes. Even in those cases in which, out of habit or for some other reason (such as, for example, a certain typical way of interpreting the task demands of imagery tasks), it turns out that people very frequently do things in certain ways when they image, this need not be of any special theoretical significance, since it may not be due to any general properties of mind, but perhaps to certain long-standing habits. For example, when we find that people typically solve problems by imagining that they are viewing a sequence of events in more detail than is actually needed to find the solution, this fact itself could be due to some relatively uninteresting reason (from a theoretical standpoint). For example, it may be that this is what subjects believed they were supposed to do; or it may simply reflect a logical way of decomposing the task so as to make use of knowledge of certain elementary facts, such as what happens when certain small changes are made to an object (e.g. in the Shepard and Feng 1972 case, what happens when a single fold is made in a sheet of paper – see the discussion of this interpretation in Pylyshyn 1978c), or subjects may simply have been in the habit of doing it in some particular way for one reason or another. The point is that mere statistical regularities need not tell us anything about the nature of mental structures. More significant theoretical questions arise when we inquire into which of these regularities are inviolable because they arise from fixed properties of mind – i.e. from the functional architecture.

Rather than placing the emphasis on explaining regularities that may well reflect little more than aggregate averages over various habits of thought, we ought to be far more impressed with the extreme flexibility that thought *can* exhibit. For example, we ought to take seriously the fact that there seems to be no specifiable limit to what the human mind can imagine or think. As George Miller recently remarked to me, the salient property of mental life is surely the fact that we can will it to do practically anything we wish: given the appropriate goals and beliefs, we can alter our behavior and our thoughts to a remarkable extent by a mere act of will. Although psychology has typically focused on the things we cannot do well or on the habitual patterns of behavior we display, one should not lose sight of the fact that a psychological theory of any interest will also have to account for the fact that most of these patterns and limitations can be overcome, and hence that they tell us little about the underlying cognitive system. Thus, however uncomfortable may be the possibility that very many of the functions that have been studied by psychologists are not fixed mechanisms – in the sense that they are cognitively penetrable – we must be prepared to recognize that what is universal and fixed about the human mind may be very different from the sorts of gross functions that we readily infer from the patterns we observe in behavior. It could well turn out – and indeed it seems extremely likely, judging from the sorts of considerations explored in this paper – that the major problem in understanding how cognition proceeds will be to explain how the vast tacit knowledge at our disposal is organized and how it is brought to bear in determining our thoughts, our imaginings, and our actions. To do this will require that we pay special attention to formalisms adequate to the task of representing the relevant knowledge, rather than primarily addressing the issue of how *typical* behavior might be generatable. This, in turn, presupposes that we take seriously such distinctions as between competence and performance (see Chomsky 1964; Pylyshyn 1977) or, to use McCarthy and Hayes's (1969) term, the distinction between the epistemological and the heuristic problems of intelligence. One considerable advantage that the computational view of cognition gives us is the potential to explore these more abstract issues formally and to work with longer deductive chains between observations and explanatory principles.

°Present address: Departments of Psychology and Computer Science, The University of Western Ontario, London, Ontario, Canada N6A 5C2

ACKNOWLEDGMENTS

This paper was written while the author was a visiting fellow at the Center for Cognitive Science, Massachusetts Institute of Technology. The support of M.I.T. and the Social Science and Humanities Research Council of Canada is gratefully acknowledged.

I also wish to acknowledge my indebtedness to Allen Newell who provided the inspiration for several of the ideas developed here. In particular the notion of mental architecture first arose from Newell's work on production systems and its importance became clear to me after some discussions with Newell at a workshop at Carnegie-Mellon University in 1973.