This is an amended version of material that first appeared in A. Clark, Microcognition: Philosophy, Cognitive Science, and Parallel Distributed Processing (MIT Press, Cambridge, MA, 1989), Ch. 1, 2, and 6. It appears in German translation in Metzinger,T (Ed) DAS LEIB-SEELE-PROBLEM IN DER ZWEITEN HELFTE DES 20 JAHRHUNDERTS (Frankfurt am Main: Suhrkamp. 1999).

MICROFUNCTIONALISM: CONNECTIONISM AND THE SCIENTIFIC EXPLANATION OF MENTAL STATES*

Andy Clark

Philosophy/Neuroscience/Psychology Program
Department of Philosophy
Washington University
St. Louis, MO 63130
e-mail: andy@twinearth.wustl.edu

* What follows is an amended version of material that first appeared in A. Clark, *Microcognition: Philosophy, Cognitive Science, and Parallel Distributed Processing* (MIT Press, Cambridge, MA, 1989), Ch. 1, 2, and 6. Thanks to MIT Press for permission to re-use this material here.

1. CLASSICAL COGNITIVISM, CONNECTIONISM AND MENTAL STATES.

My goal in the present treatment is to sketch and compare two scientific approaches to understanding the mind. The first approach, that of classical cognitivism, depicts mind as a manipulator of chunky, quite high-level, symbols. The second approach, that of connectionism (Artificial Neural Networks, Parallel Distributed Processing) depicts mind as a product of the complex interactions between multiple so-called sub-symbolic elements. I shall try to clarify this contrast by associating classical cognitivism with the development of what I shall call semantically transparent systems, and connectionism with the deliberate eschewal of this strategy. Connectionism, I then argue, represents a subtle twist on the standard philosophical view of mental states as *functional* states. For it suggests a kind of microfunctionalism in which the inner roles do not map neatly onto roles determined by our everyday, contentful, purposive characterizations of the mental. (For the reader unfamiliar with some of these terms, such as functionalism, sub-symbolic, etc. -- don=t worry: these will be explained as we go along).

2. BACKDROP: TURING, MCCARTHY, NEWELL, AND SIMON.

The bigger the names, the harder they drop. These would dent the kinds of floors that supported ancient mainframes. It would be fair to say that Turing made AI conceivable, and McCarthy (along with Minsky, Newell, and Simon) made it possible. Despite occasional pronouncements to the contrary, I think we are still waiting to see it made actual, but more on that in due course.

Turing's (1937) achievement was to formalize the notion of computation itself, using the theoretical device we call a Turing machine. He thereby paved the way for mathematical investigations of computability. But significantly, Turing's formalization also 1) encompassed a whole *class* of mechanisms grouped together not by details of actual physical composition but by their formal properties of symbol manipulation, 2) showed how such mechanisms could tackle any sufficiently well specified problem that would normally require human intelligence to solve, and 3) showed how to define a special kind of Turing machine (the universal Turing machine) which could imitate any other Turing machine and thus perform any cognitive task that any other Turing machine could perform. I shall not review the details of Turing's demonstrations here[1]. For present purposes what matters is that Turing's ideas suggested the notion of machines that, by their formal structure, imitate (and even emulate) the mind. The material stuff (valves, silicon, or whatever) did not matter; the formal properties guaranteed in principle a capacity to perform any sufficiently well specified cognitive task. In the words of one major figure:

> Turing's work can be seen as the first study of cognitive activity fully abstracted in principle from both biological and phenomenological foundations.... It represents the emergence of a new level of analysis, independent of physics yet mechanistic in

[1] See Turing (1937) (1950). For thorough summaries see Haugeland (1985), Hodges (1983).

> spirit. It makes possible a science of structure and function divorced from material substance.... Because it speaks the language of mental structures and internal processes, it can answer questions traditionally posed by psychologists (Pylyshyn, 1986, p. 68).

Classical cognitivism, thanks to the work of Turing, was on the cards. It was some time, however, before it could develop into a viable, experimental discipline. That development required first the arrival of the general-purpose digital computer and second the availability of a powerful and flexible high-level programming language. John von Neumann provided the practical design, and John McCarthy, around 1960, provided a language. The language was called LISP, which stood for list processing, and it made possible the first sustained run of research and development within the classical-cognitivist paradigm[2]. This run of research and development became theoretically self-conscious and articulate with Newell & Simon's abstraction of the notion of a physical symbol system.

[2] In fact, the idea of list processing (in which data structures contain symbols that point to other data structures, which likewise contain symbols that point to other data structures and so on, thus facilitating the easy association of information with symbols) was introduced by Allan

3. THE PHYSICAL-SYMBOL-SYSTEM HYPOTHESIS.

A *physical symbol system*, according to Newell & Simon (1976, p. 40-42) is any member of a general class of physically realizable systems meeting the following conditions:

> 1) It contains a set of symbols, which are physical patterns that can be strung together to yield a structure (or expression).
>
> 2) It contains a multitude of such symbol structures and a set of processes that operate on them (creating, modifying, reproducing and destroying them according to instructions, themselves coded as symbol structures).
>
> 3) It is located in a wider world of real objects and may be related to that world by designation (in which the behavior of the system affects or is otherwise consistently related to the behavior or state of the object) or interpretation (in which expressions in the system designate a process, and when the expression occurs, the system is able to carry out the process).

Newell and Herbert Simon in their program Logic Theorist of 1956. For a detailed treatment of list processing see Charniak & McDermott (1985) Ch. 2, and also Newell & Simon (1976). One attractive and important feature of LISP was its universal function *eval*, which made it as adaptable as a universal Turing machine, barring constraints of actual memory space in any implementation.

In effect, a physical symbol system is any system in which suitably manipulable tokens can be assigned arbitrary meanings and, by means of careful programming, can be relied on to behave in ways consistent (to some specified degree) with this projected semantic content. Any general-purpose computer constitutes such a system. What, though, is the relation between such systems and the phenomena of mind (hoping, fearing, knowing, believing, planning, seeing, recognizing, and so on)? Newell & Simon are commendably explicit once again. Such an ability to manipulate symbols, they suggest, is the scientific essence of thought and intelligence, much as H20 is the scientific essence of water. According to the physical-symbol-system hypothesis, the necessary and sufficient condition for a physical system to exhibit general intelligent action is that it be a physical symbol system, Newell & Simon thus claim that any generally intelligent physical system will be a physical system (the necessity claim) and that any physical symbol system can be organized further to exhibit general intelligent action (the sufficiency claim). And general, intelligent action, on Newell & Simon=s gloss, implies the same scope of intelligence seen in human action.

It is important to be as clear as possible about the precise nature of Newell & Simon's claim. As they themselves point out (1976, p. 42), there is a weak (and incorrect) reading of their ideas that asserts simply that a physical symbol system is (or can be) a universal machine capable of any well-specified computation, that the essence of intelligence lies in computation, and that intelligence could therefore be realized by a universal machine (and hence by a physical symbol system). The trouble with this reading is that by leaving the nature of the computations involved so unspecified, it asserts rather too little to be of immediate psychological interest. Newell and Simon rather intend the physical-symbol-system hypothesis as a specific *architectural* assertion about the nature of intelligent systems (1976, p. 42, my emphasis). It is fair, if a little blunt, to render this specific architectural assertion as follows:

> *The strong-physical-symbol-system (SPSS) hypothesis.* A virtual machine engaging in the von Neumann-style manipulation of standard symbolic atoms has the direct and necessary and sufficient means for general intelligent action.

It will be necessary to say a little about the terms of this hypothesis and then to justify its ascription to Newell and Simon.

About the terms, I note the following. A virtual machine is a "machine" that owes its existence solely to a program that runs (perhaps with other intervening stages) on a real, physical machine and causes it to imitate the usually more complex machine to which we address our instructions (see, for example Sloman, 1984). Such high-level programming-languages as LISP, PROLOG, and POP11 thus define virtual machines. And a universal Turing machine, when it simulates a special-purpose Turing machine, may be treated as a virtual version of the special-purpose machine.

"Von Neumann-style manipulation is meant to suggest the use of certain basic manipulatory operations easily provided in a Von Neumann machine running a high-level language like LISP. Such operations would include assigning symbols, binding variables, copying, reading and amending symbol strings, basic syntactic, pattern-matching operations (more on which later), and so on. Connectionist processing, as we shall see, involves a radically different repertoire of primitive operations.

The next phrase to consider is "standard symbolic atoms." This highlights what *kinds* of entities the SPSS approach defines its computational operations to apply to. They are to apply to symbolic expressions whose parts (atoms) are capable of being given an exact semantic interpretation in terms of the concepts and relations familiar to us in daily, or at any rate public, language. These are words (atoms) such as "table," "ball," "loves,"orbit," "electron," and so forth. Some styles of connectionism and a few more conventional models (e.g., those of computational linguistics) involve a radical departure from the use of standard symbolic atoms. Since this contrast looms quite large in what follows, it will be expanded upon in section 5 below.

Finally, the locution "*direct* and necessary and sufficient means for general intelligent action" is intended to capture a claim of architectural sufficiency. In effect, the claim is that a strong physical symbol system, as just defined, will be immediately capable of genuine intelligent action. That is, such a machine could be truly intelligent quite independently of any particular architectures (any other real or virtual machines on which it is built), and conversely, it could be so without simulating any other architectures or machines. The SPSS hypothesis thus makes a highly specific and laudably Popperian claim.

What evidence is there to associate such a claim with Newell and Simon? Quite a lot. Some of the evidence comes in the form of reasonably explicit assertions. Some can be inferred from the details of their actual work in AI. And some (for what it is worth) can be found in the opinions of other commentators and critics. A brief review of some of this evidence follows. It is perhaps worth noting that even if Newell and Simon were to deny any commitment to the SPSS hypothesis, the formulation would still serve our purposes, since something like that hypothesis informs the philosophers' view of artificial intelligence and without doubt, still informs (perhaps unconsciously) a great deal of work within AI itself.

4. BRINGING HOME THE BACON.

Still, a little evidence never goes amiss. For a start, we find the following comment sandwiched between Newell and Simon's outline of the nature of a physical symbol system and their explicit statement of the hypothesis: "The type of system we have just defined ... bears a strong family resemblance to all general purpose computers. If a symbol-manipulation language, such as LISP, is taken as defining a machine, then the kinship becomes truly brotherly" (Newell and Simon 1976, 41). Douglas Hofstadter (1985, 646, 664), who takes issue with the idea that baroque manipulations of standard LISP atoms could constitute the essence of intelligence and thought, is happy to ascribe just that view to Newell and Simon.

Moreover, Newell and Simon's own practice does seem to bear such an ascription out. Thus, all their work, from the early General Problem Solver (1963) to their more recent work on production systems[3] and on automating scientific creativity, has been guided by the notion of *serial heuristic search* based on protocols, notebook records, and observation of human subjects. (Heuristic search is a means of avoiding the expensive and often practically impossible systematic search of an entire problem space by using rules of thumb to lead you quickly to the area in which with a little luck the solution is to be found.) For our purposes, the things that most significantly characterize this work (and much other work in contemporary AI besides -- see, for example, the AM program mentioned below) are its reliance on a serial application of rules or heuristics, the rather high-level, consciously introspectable grain of most of the heuristics involved, and the nature of the chosen-task domains. I shall try to make these points dearer by looking at the example of BACON and some of its successors, a series of programs that aim to simulate and explain

[3] A production system is essentially a set of if-then pairs in which the if specifies a condition that, if satisfied, causes an action (the then) to be performed. Each if-then pair or condition-action rule is called a production (see for example, Charniak & McDermott (1985), p. 438-439).

the process of scientific discovery (Langley (1979); Simon (1979) (1987); Langley, et al. (1987)).

BACON sets out to induce scientific laws from bodies of data. It takes observations of the values of variables and searches for functions relating the values of the different variables. Along the way it may introduce new variables standing for the ratio of the value of the original variables. When it finds an invariant, a constant relation of the values of different variables, it has (in some sense) discovered a scientific law. Thus, by following simple heuristics of the kind a person might use to seek relations among the data ("try the simple relations first," "treat nonconstant products of ratios between variables as new variables," etc.), BACON was able to generate from Kepler's data "ratios of [successive] powers of the radii of the planets' orbits to [successive] powers of their periods of revolution, arriving at the invariant D^3/P^2 (Kepler's third law), after a search of a small number of possibilities," (Simon (1979), p. 1088). Similarly, BACON arrived at Ohms law by noticing that the product of electrical current and resistance is a constant.

Now for a few comments on BACON. First, BACON makes its discoveries by working on data presented in notational formats (e.g., measures of resistance, periods of planetary revolution) that represent the fruits of centuries of human labor. Manipulating these representations could be the tip of the iceberg; creating them and understanding them may constitute the unseen bulk. I say a little more about this later.

For now, simply note that BACON and other programs like AM and EURISKO[4], and MYCIN (below) help themselves to our high-level representational formalism. In a recent

[4] The AM program, as Lenat himself later pointed out, worked partly due to the amount of mathematical knowledge implicit in LISP, the language in which it was written (see Lenat (1983a) (1983b), Lenat & Brown (1984), Ritchie & Hanna (1984)). This result seems related to the observations concerning the amount of the work of scientific discovery already done by giving BACON data arranged in our representational notation. Lenat=s later work, EURISKO avoids the Adefect@ of trading on the quasi-mathematical nature of LISP. But it too relies on our giving it notationally predigested data.

work Langley et al. (1987), p. 326) are sensitive to the problem of creating new representational formalisms. But they insist that such problems can be solved within the architectural paradigm associated with the SPSS hypothesis.

Second and relatedly, the knowledge and heuristics that BACON deploys are coded rather directly from the level of thought at which we consciously introspect about our own thinking. This is evident from Simon's (1987) statement that he relies heavily on human protocols, laboratory notebooks, etc. BACON thus simulates, in effect, the way we reason when we are conscious of trying to solve a problem and it uses kinds of heuristics that with some effort we might explicitly formulate and use as actual, practical rules of thumb. This level of modeling is common to much but not all contemporary work in AI, including work in expert systems and qualitative reasoning (see, e.g., the section "Reasoning about the Physical World@ in Hallam & Mellish (1987)). Thus the MYCIN rule (Shortliffe (1976)) for blood injections reads: If (1) the site of the culture is blood, (2) the gram stain of the organism is gramneg, (3) the morphology of the organism rod, and (4) the patient is a compromised host, then there is suggestive evidence that the identity of the organism is pseudomonasaeruginosa (from Feigenbaum (1977), p. 1014).

Likewise, BACON's representation of data was at the level of attribute-value pairs, with numerical values for the attributes. The general character of the modeling is even more apparent in programs for qualitative discovery: GLAUBER and STAHL. GLAUBER applies heuristic rules to data expressed at the level of predicate-argument notation, e.g., reacts [inputs (HCl, NH3) outputs (NH4, Cl)]," and STAHL deploys such heuristics as "identify components: If *a* is composed of *b* and *c*, and *a* is composed of *b* and *d*, and neither *c* contains *d* nor *d* contains *c*, then identify *c* with *d*."

Third, BACON uses fairly slow serial search, applying its heuristics one at a time and assessing the results. Insofar as BACON relies on productions, there is an element of parallelism in the search for the currently applicable rule. But only one production fires at a time, and this is the seriality I have in mind. Serial behavior of this kind is characteristic of slow, conscious thought. And Hofstadter (1985), p. 632) reports Simon as asserting, "Everything of interest in cognition happens above the 100 millisecond level -- the time it takes you to recognize your mother." Hofstadter disagrees vehemently, asserting that everything of interest in cognition takes place below the 100 millisecond level. My position, outlined below, is sympathetic to Hofstadter's (and indeed owes a great deal to it). But I believe that the notion of a dispute over the correct level of interest here is misplaced. There are various explanatory projects here, all legitimate. Some require us to go below the 100-millisecond level (or whatever) while others do not. This relates, in a way I expand on later, to a problem area cited by Simon in a recent lecture (1987); see also Langley, et al. (1987), p. 14-16). Simon notes that programs like BACON are not good at very ill structured tasks, tasks demanding a great deal of general knowledge and expectations. Thus, though BACON neatly arrives at Kepler's third law when given the well-structured task of finding the invariants in the data, it could not come up with the flash of insight by which Fleming could both see that the mold on his petri dish was killing surrounding bacteria and recognize this as an unusual and potentially interesting event.

Listening to Simon, one gets the impression that he believes the way to solve these unstructured problems is to *expand* the set of high-level data and heuristics that a system manipulates in the normal, slow, serial way (i.e., by creating, modifying, and comparing high-level symbol strings according to stored rules). Thus, in a recent coauthored book he dismisses the idea that the processes involved in the flash-of-insight type of discovery might be radically different in computational kind, saying that the speed and subconscious nature of such events "does not in any way imply that the process is fundamentally different from other processes of discovery- that we must seek for other sources of evidence about its nature (i.e., subjects' introspections can no longer help)" (Langley et al. (1987), p. 329).

The position I develop holds rather that the folk-psychological term "scientific discovery" encompasses at least two quite different kinds of processes. One is a steady, Von Neumann-style manipulation of standard symbolic atoms in a search for patterns of regularity. And this is well modeled in Simon and Langley's work. The other is the flash-of-insight type of recognition of something unusual and interesting. And this, I shall suggest, may require modeling by a method quite different (though still computational).

In effect, theorists such as Langley, Simon, Bradshaw, and Zytkow are betting that all aspects of human thought will turn out to be dependent on a single kind of computational architecture. That is an architecture in which data is manipulated by the copying, reorganizing, and pattern matching capabilities deployed on list structures by a Von Neumann (serial) processor. The basic operations made available in such a setup *define* the computational architecture it is. Thus, the pattern matching operations which such theorists are betting on are the relatively basic ones available in such cases (i.e., test for complete syntactic identity, test for syntactic identity following variable substitution, and so on). Other architectures (for example, the PDP architecture discussed later) provide different basic operations. In the case of parallel distributed processing these include a much more liberal and flexible pattern-matching capacity able to find a best match in cases where the standard SPSS approach would find no match at all.

Langley, Simon, et al. are explicit about their belief that the symbol processing architecture they investigate has the resources to model and explain all the aspects of human thought. Faced with the worry that the approach taken by BACON, DALTON, GLAUBER, and STAHL won't suffice to explain all the psychological processes that make up scientific discovery, they write, "Our hypothesis is that the other processes of scientific discovery, taken one by one, have [the] same character, so that programs for discovering research problems, for designing experiments, for designing instruments and for representing problems will be describable by means of the same kinds of elementary information processes that are used in BACON" (1987),p. 114). They make similar comments concerning the question of mental imagery (p. 336). This insistence on a single architecture of thought may turn out to be misplaced. The alternative is to view mind as a complex system comprising many virtual architectures. If this is true, psychological explanation will likewise need to deal in a variety of types of models, availing itself in each case of different sets of basic operations (relative to the virtual architecture).

Finally, a word about the methodology of BACON and its classical cognitivist cousins. These programs characteristically attempt to model *fragments* of what we might term recent human achievements. By this I mean they focus on tasks that we intelligent, language-using human beings perform (or at least think we perform) largely by conscious and deliberate efforts. Such tasks tend to be well structured in the sense of having definite and recognizable goals to be achieved by deploying a limited set of tools (e.g., games and puzzles with prescribed legal moves, theorem proving, medical diagnosis, cryparithmetic and so on). They also tend to be the tasks we do slowly and badly in comparison with perceptual and sensorimotor tasks, which we generally do quickly and fluently. Some AI workers are dubious about this choice of task domain and believe it essential to tackle the fluent, unconscious stuff first before going on to model more evolutionarily recent achievements. Marr (1977), p. 140) gives the classic statement of this: "Problem-solving research has tended to concentrate on problems that we understand well intellectually but perform poorly on.... I argue that [there are] exceptionally good grounds for not studying how we carry out such tasks yet. I have no doubt that when we do (e.g.) mental arithmetic we are doing *something* well, but it is not arithmetic and we seem far from understanding even one component of what that something is. Let us therefore concentrate on the simpler problems first. I have expressed similar views based on direct evolutionary arguments (Clark (1986)). There still seems to me to be much truth in such strictures. But the overall picture to be developed here is rather more liberal, as we shall see.

5. SEMANTICALLY TRANSPARENT SYSTEMS.

It is time to expand on the notion of a standard symbolic atom, introduced in section 3 above. One of the most theoretically interesting points of contrast between classical systems (as understood by philosophers like Fodor and Pylyshyn) and connectionist systems (as understood by theorists like Smolensky) concerns the precise sense in which the former rely on, and the latter eschew, the use of such symbolic atoms. To bring out what is at issue here, I shall speak of the classicist as (by definition) making a methodological commitment to the construction of semantically transparent systems. Credit for the general idea of semantic transparency (though not the label) belongs elsewhere. The analysis I offer is heavily influenced by ideas in Smolensky 1988. Thus let us say that:

> A system will be said to be *semantically transparent* just in case it is possible to describe a neat mapping between a symbolic (conceptual level) semantic description of the system's behavior and some *projectible* semantic interpretation of the internally represented *objects of* its formal computational activity.

The general notion of a semantically transparent system (STS) may be best appreciated from the perspective offered by Marr=s now-standard account of the levels of understanding of an information-processing task. Marr (1982) distinguishes three levels at which a machine carrying out an information-processing task needs to be understood.

> *Level 1, computational theory.* This level describes the goal of the computation, the general strategies for achieving it, and the constraints on such strategies.
>
> *Level 2, representation and algorithm.* This describes an algorithm, i.e., a series of computational steps that does the job. It also includes details of the way the inputs and outputs are to be represented to enable the algorithm to perform the transformation.
>
> *Level 3, implementation.* This shows how the computation may be given flesh (or silicon) in a real machine.

In short, level 1 considers what function is being computed (at a high level of abstraction), level 2 finds a way to compute it, and level 3 shows how that way can be realized in the physical universe.

Suppose that at level I you describe a task by using the conceptual apparatus of public language. (This is not compulsory at level 1 but is often the case.) You might use such words as “liquid,” “flow,” “edge,” and so on. You thus describe the function to be computed in terms proper to what Paul Smolensky calls the conceptual level, the level of public language. Very roughly, a system will count as an STS if the computational objects of its algorithmic description (level 2) are isomorphic to its task-analytic description couched in conceptual level terms (level 1). What this means is that the computational operations specified by the algorithm are applied to internal representations that are projectibly[5] interpretable as standing for conceptual-level entities.

Some examples may help to sharpen these levels. Consider the following specifications of functions to be computed.

(1)If (cup and saucer) then (cup)

If (cup and saucer) then (saucer)

(2)If (verb stem + ending) then (verb stem + ed)

The functions in (1) are clear examples of a conceptual level specification. Though (2) does not draw on daily language, it is nonetheless a related case. In each case the items in parentheses are structural descriptions whose structure is semantically significant.

[5] For clarification of this notion of projectability, see Clark (1989), ch. 6, section 3.

A semantically transparent system, we may now say, is one in which the *objects* (e.g., "cup and saucer) of state-transition rules in the task analysis (e.g., the rule "if (cup and saucer) then (cup)") have *structural analogues* in the actual processing story told at level 2. That is to say, in the case of (1), the level 2 story will involve computational operations defined to apply to representations sharing the complex structure of the expression "cup and saucer." In the case of (2), the level 2 story will involve computational operations defined to apply to descriptions of input verbs in a way that reveals them to have the structure "verb stem + ending." It is in this sense that classical, semantically transparent systems may be said to have a certain kind of *syntax.* For they posit mental representations that have actual structures echoing the semantic structures of our level-I description. As Fodor and Pylyshyn (1988) forcibly point out, this is very handy if we want our system to perform systematically with respect to a certain semantic description. For it has the effect of making the semantic description a real object for the system. Hence, any inferences, etc., that are systematic in the semantic description can easily be mimicked by relying on the syntactic properties of its internal representations. If we want our system to treat "(cup and saucer)" as an instance of a general logical schema "(a and b)" and hence to perform all kinds of deductive inferences on arguments involving cups and saucers, this will be a simple matter just so long as the system's representation of cups and saucers is semantically transparent and preserves structure.

Clearly, the notion of a semantically transparent system is intended to capture the substance of Fodor and Pylyshyn's definition of a classical approach to cognitive science. Classical and connectionist approaches differ, according to Fodor and Pylyshyn, in two vital respects.

> (1) AClassical theories -- but not connectionist theories -- posit a "language of thought". This means that they posit mental representations (data structures) with a certain form. Such representations are *syntactically structured,* i.e., they are

> systematically built by combining atomic constituents into molecular assemblies, which (in complex cases) make up whole data structures in turn. In short, they posit *symbol systems* with a combinatorial syntax and semantics.
>
> (2) "In classical models, the principles by which mental states are transformed, or by which an input selects the corresponding output, are defined over structural properties of mental representations. Because classical mental *representations* have combinatorial structure, it is possible for classical mental *operations* to apply to them by reference to their form." This means that if you have a certain kind of structured representations available (as demanded by point 1), it is possible to define computational operations on those representations so that the operations are sensitive to that structure. If the structure isn't there (i.e., if there is no symbolic representation), you couldn't to it, though you might make it *look* as if you had by fixing on a suitable function in extension. (Quotes are from Fodor and Pylyshyn (1988), p. 12-13.)

In short, a classical system is one that posits syntactically structured, symbolic *representations* and that defines its computational *operations to* apply to such representations in virtue of their structure.

The notion of a semantically transparent system is also meant to capture the spirit of Smolensky's views on the classical/connectionist divide, as evidenced in comments like the following:

> A symbolic model is a *system* of interacting processes, all with the same conceptual-level semantics as the task behavior being explained. Adopting the terminology of Haugeland (1978), this *systematic explanation* relies on a *systematic reduction* of the behavior that involves no shift of semantic domain or *dimension.* Thus a game-playing program is composed of subprograms that generate possible moves, evaluate them and so on. In the symbolic paradigm these systematic

> reductions play the major role in explanation. The lowest level processes in the systematic reduction, still with the original semantics of the task domain, are then themselves reduced by *intentional instantiation:* they are implemented exactly by other processes with different semantics but the same form. Thus a move-generation subprogram with game semantics is instantiated in a system of programs with list-manipulating semantics. (Smolensky (1988), p. 11)

Before leaving the subject of STSs, it is worth pausing to be quite explicit about one factor that is not intended as part of the definition of an STS. Under the terms of the definition an STS theorist is not committed to any view of how the system explicitly represents the rules adduced in task analysis (level 1). Thus, in my example (1), there is no suggestion that the rule 'If (cup and saucer) then (cup)" must *itself* be explicitly represented by the machine. A system could be an STS and be hard-wired so as to take the input "cup and saucer" and transform it into the output "cup." According to STS theory, all that must be explicit is the structured description of the objects to which the rule is defined to apply. The derivation rules may be tacit, so long as the data structures they apply to are explicit. On this Fodor and Pylyshyn rightly insist: "Classical machines can be *rule implicit* with respect to their programs.... What *does* need to be explicit in a classical machine is not its program but the symbols that it writes on its tapes (or stores in its registers). These, however, correspond not to the machine's rules of state transition but to its data structures" (1988), p. 61). As an example they point out that the grammar posited by a linguistic theory need not be explicitly represented in a classical machine. But the *structural descriptions of sentences* over which the grammar is defined (e.g., in terms of verb stems, subordinate clauses, etc.) must be. Attempts to characterize the classical/ connectionist divide by reference to explicit or non explicit rules are thus shown to be in error.

6. FUNCTIONALISM.

It is now time to introduce a major and more straight-forwardly philosophical protagonist, the functionalist. The functionalist is in many ways the natural bedfellow of the proponent of the physical-symbol-system hypothesis. For the physical-symbol-system hypothesis claims that what is essential to intelligence and thought is a certain capacity to manipulate symbols. This puts the essence of thought at a level independent of the physical stuff out of which the thinking system is constructed. Get the symbol manipulating capacities right and the stuff does not matter. As the well-known blues number has it, "It ain't the meat, it's the motion." The philosophical doctrine of functionalism echoes this sentiment, asserting (in a variety of forms) that mental states are to be identified not with, say, physicochemical states of a being but with more abstract organizational, structural, or informational properties. In Putnam's rousing words 'We could be made of Swiss cheese and it wouldn't matter" (1975), p. 291). Aristotle, some would have it, may have been the first philosophical functionalist. Though there seems to be a backlash now underway (see, e.g., Churchland (1981), and Churchland (1986)), the recent popularity of the doctrine can be traced to the efforts of Hilary Putnam (1960) (1967), Jerry Fodor (1968), David Armstrong (1970) and, in a slightly different vein, Daniel Dennett (1981) and William Lycan (1981). I shall not attempt to do justice to the nuances of these positions here. Instead, I shall simply characterize the most basic and still influential form of the doctrine. First though, a comment on the question to which functionalism is the putative answer.

It is useful to distinguish the various explanatory projects for which ideas about the mind are put forward. Thus note that the classical *philosophical* project has been to formulate and assess *schemas* for a substantial theory of the essence of the mental. The notion of essence here may be unpacked as the search for the necessary and sufficient conditions for being in some mental state. In this restricted sense a theory of mind should tell us what it is about a being that makes it true to assert of that being that it is in a given mental state (e.g., believing it is about to rain, feeling sad, feeling anxious, suffering a stabbing pain in the left toe, and so forth).

For the moment let me simply assert that Newell and Simon's intended project (in common with a lot of workers in AI) is psychological explanation. Pending a fuller account of psychological explanation, it is not obvious that the project of psychological explanation is identical with the project of seeking the essence of the mental in the sense just sketched. But Newell & Simon's talk of the physical-symbol-system hypothesis as an account of the necessary and sufficient conditions of intelligent action effectively identifies the tasks. It follows that having a full psychological explanation in their sense would put you in a position to re-create or instantiate the analyzed mental state in a machine (barring practical difficulties). I elsewhere argue for a firm distinction between these projects of psychological explanation and psychological instantiation[6].

Functionalism, then, is a sketch or schema of the kind of theory that, when filled in, will tell us in a very deep sense what it *is* to be in some mental state. The most basic form of such a theory is known as Turing machine functionalism. Not surprisingly, the doctrine takes its cue from Turing's conception of the formal properties sufficient to guarantee that a task is computable by a mechanism regardless of the physical stuff out of which the mechanism was made (see section 2 above).

In Putnam's hands (1960) (1967) functionalism came to suggest a theory of mind (in the sense of a schema for a substantial theory of the essence of the mental) that was apparently capable of avoiding many of the difficulties that beset other such proposals.

[6] See Clark (1989), ch. 10.

Very sketchily, the situation was something like this. Dualism (the idea that mind is a ghostly kind of nonmaterial *substance)* had been discredited as nonexplanatory mysticism and was briefly displaced by behaviorism. Behaviorism (Ryle (1949) held that mental states were identical with sets of actual and counterfactual overt behaviors and that inner states of the subject, though no doubt causally implicated in such behaviors, were not theoretically important to understanding what it *is* to be in certain mental states.

This dismissal of the importance of internal states (for a philosophical theory of. mind) was resisted by the first wave of identity theories who claimed that mental states were identical with brain processes (Smart (1959)). But the identity theory, if one took the claims of its proponents rather literally (more literally, I am inclined to think, than they ever intended) lay open to a variety of criticisms. Especially relevant here is Putnam's (1960) (1967) criticism that identity theory makes far too tight the tie between being in a certain mental state (e.g., feeling pain) and being in a certain physicochemical or neural state. For on an extreme, type-type-identity reading, the identity of some mental state with, say, some neural state would seem to imply that a being incapable of being in that neural state could not, in principle, be in the mental state in question. But for rather obvious reasons, this was deemed unacceptable. A creature lacking neurons would be unable to occupy any neural state. But couldn't there be exotic beings made of other stuff who were nonetheless capable of sharing our beliefs, desires, and feelings? If we allow this seemingly sensible possibility then we, as philosophers, need some account of what physically variously constituted feelers and believers have in common that makes them feelers and believers. Behaviorism would have done the trick but its denial of the importance of inner states had been perceived as a fault. Identity theory, it seemed, had gone too far in the other direction.

Between the Scylla and the Charybdis sailed the good ship functionalism. What is essential to being in a certain mental state, according to the functionalist schema, is being in a certain abstract functional state. And that functional state is defined over two components: (1) the role of some internal states in mediating system input and system output (the behavior element) and (2) the role of the states or processes in activating or otherwise affecting other internal states of the system (the inner element). If we *also* presume that cognition is a computational phenomenon, then we can link this characterization (as Putnam [1960] did) to the notion of a Turing machine, which is defined by its input and output and its internal state transition profile. What Turing machine you are instantiating, not what substance you are made of, characterizes your mental states. As I said, it ain't the meat, it's the motions.

7. FUNCTIONALISM AND THE FORMAL SHADOWS OF MIND.

Now the bad news. Functionalism has had its problems. One charge is that of excessive liberalism (see Block (1980)). The worry is that Turing-machine functionalism allows too many kinds of things to be possible believers and thinkers. For example, it might in principle be possible to get the population of China to pass messages (letters, values, whatever) between themselves so as briefly to realize the functional specification of some mental state (Block (1980), p. 276-278). (Recall. it is only a matter of correctly organizing inputs, outputs, and internal state transitions, and these, however they are specified, won=t be tied to any particular kind of organism.) As Block (1980), p. 277) puts it, "In describing the Chinese system as a Turing machine I have drawn the line [i.e., specified what counts as inputs and outputs] in such a way that it satisfies a certain type of functional description-one that you *also* satisfy, and one that, according to functionalism, justifies attributions of mentality." But, says Block, there is at least prima facie reason to doubt that such a system could have any mental states at all. Could that overall system really constitute say, an agent in pain? Surely not. Surely there is nothing which it is like, either nice or nasty, to be such a system. It has no phenomenonal or subjective experience. Or as philosophers put it, the system has no qualia (raw feels, real subjectivity). Hence Block dubs this argument the absent-qualia argument.

Another worry is that typical functional/cognitivist models capture not the real essence so much as the shallow formal shadows of mind. In a series of influential publications (1980, 1983, 1984) John Searle has established himself as a leading opponent of the information-theoretic approach to mind. That approach, he thinks, is just tilting at the "formal shadows" of mind. But in contrast, real mentality, he says, depends on far wetter things, namely, on the physicochemical properties of human brains. Searle's criticism is targeted on what he calls "the hypothesis of strong AI." This is defined as the claim that "the appropriately programmed computer literally has cognitive states and that the programs thereby explain human cognition" (Searle (1980), p. 283). The attack begins with a now infamous thought experiment, the puzzling case of the Chinese room. This thought experiment aims to provide a general critique of the computational approach to mind. Its starting point is a specific program that might seem to simulate the intentional activity of understanding a story (Schank & Abelson (1977)). Very briefly, the program provides the computer with some background data concerning the topic of a story to be presented. The computer can then be given a story on this topic and afterward it will answer questions about the story that are not explicitly resolved in the story itself. Thus, to use Searle's example, we might program in background data on human behavior in restaurants. We may then tell the story of a man who enters a restaurant, orders a hamburger, and upon leaving, presents the waitress with a big tip. If the computer is then asked, “And did the man *eat* the hamburger?” it can answer "yes,” because it apparently knows about restaurants. Searle believes, I think rightly, that the computer running this program does not really know about restaurants at all, at least if by "know" we mean anything like "understand." The Chinese-room example is constructed in part to demonstrate this. But Searle believes his arguments against *that* sort of computational model of understanding are also arguments against any computational model of understanding.

We are asked to imagine a human agent, an English monolinguist, placed m a large room and given a batch of papers with various symbols on it. These symbols, which to him are just meaningless squiggles identifiable only by shape, are in fact the ideograms of the Chinese script. A second batch of papers arrives, again full of ideograms. Along with it there arrives a set of instructions in English for correlating the two batches. Finally, a third batch of papers arrives bearing still further arrangements of the same uninterpreted formal symbols and again accompanied by some instructions in English concerning the correlation of this batch with its predecessors. The human agent performs the required matchings and issues the result, which I shall call "the response," This painstaking activity, Searle argues, corresponds to the activity of a computer running Schank's program. For we may think of batch 3 as the questions, batch 2 as the story, and batch 1 as the script or background data. The response, Searle says, may be so convincing as to be indistinguishable from that of a true Chinese speaker. And yet, and this is the essential point, the human agent performing the correlations understands no Chinese, just as, it would now appear, a computer running Schank's program understands no stories. In each case what is going on is the mere processing of information. If the intuitions prompted by the Chinese-room example are correct, understanding must involve something extra. From this Searle concludes that no computer can ever understand merely by "performing computational operations on formally specified elements." Nor, consequently, can the programs that determine such computational operations tell us anything about the special nature of mind (Searle (1980), p. 286).

Ramming the point home, Searle asks us to compare the understanding we (as ordinary English speakers) have of a story in English against the "understanding" the person manipulating the formal symbols in the Chinese room has of Chinese. There is, Searle argues, no contest. “In the Chinese case I have everything that Artificial Intelligence can put into me by way of a program and I understand nothing; in the English case I understand everything and there is so far no reason at all to suppose that my understanding has anything to do with computer programs -- i.e., with computational operations on purely formally specified elements" (Searle (1980), p. 286). In short, no formal account can be *sufficient* for understanding, since "a human will be able to follow the formal principles without understanding anything" (p. 287). And there is no obvious reason to think that satisfying some *formal* condition is *necessary* either, though as Searle admits, this could (just conceivably) yet prove to be the case. The formal descriptions, Searle thinks (p. 299), seem to be capturing just the shadows of mind, shadows thrown not by abstract computational sequences but by the actual operation of the physical stuff of the brain.

I shall argue that Searle is simply wrong thus *completely* to shift the emphasis away from formal principles on the basis of a demonstration that the operation of a certain kind of formal program is insufficient for intentionality. The position to be developed below (and in much more detail in Clark (1989)) focuses instead on the instantiation of a certain kind of formal description that is far more microstructural than the descriptions of the SPSS hypothesis. Undermining Searle=s strongest claims, however, is no simple matter, and we must proceed cautiously. The best strategy is to look a little more closely at the *positive* claims about the importance of the nonformal, biological, stuff.

8. SHOWING WHAT WE'RE MADE OF.

Searle considers several possible replies to his paper, only one of which need interest us here. It is what he calls the brain-simulator reply, and it goes like this. Suppose a program modeled the formal structure of actual Chinese brains engaged in understanding Chinese. Surely then it would constitute a genuine Chinese understanding. At this point, to his credit, Searle grasps the nettle. "No," he says, "we could imagine an elaborate set of water pipes and valves, and a human switcher, realizing that formal description too. But wherein would the understanding of Chinese reside? Surely the answer is >nowhere=@ (adapted from Searle (1980), p. 295). (This argument should recall the worries about excessive liberalism and absent qualia, which had functionalism in a vicelike grip.) Regarding the brain simulator then, Searle is in no doubt: "As long as it simulates only the formal structure of the sequence of neuron firings at the synapses, it won't have simulated what matters about the brain, namely its causal properties, its ability to produce intentional states" (Searle (1980), p. 295). Or again: "What matters about brain operation is not the formal shadow cast by the sequence of synapses but rather the actual properties of the sequences" (Searle (1980), p. 300).

The allusions to causal powers have struck many critics as unforgivably obscure. It is hard to see why. Searle's claim has two components: (1) The formal properties of the brain do not constitute intentionality. (2) the reason they do not constitute it is that only certain kinds of stuff can support thought. Well, (1) may be right (see Clark (1989), ch. 3), though not for the reasons cited in (2). But even so, (2) is surely not that obscure a claim. Searle cites the less puzzling case of photosynthesis. By focusing on this, we may begin to unscramble the chaos.

Photosynthesis, Searle suggests, is a phenomenon dependent on the actual causal properties of certain substances. Chlorophyl is an earthly example. But perhaps other substances found elsewhere in the universe can photosynthesize too. Similarly, Martians might have intentionality, even though (poor souls) their brains are made of different stuff from our own. Suppose we now take a formal chemical theory of how photosynthesis occurs. A computer could then work through the formal description. But would actual photosynthesis thereby take place? No, it's the wrong stuff, you see. The formal description is doubtless a handy thing to have. But if it's *energy* (or thought) you need, you had better go for the real stuff. In its way, this is fair enough. A gross formal theory of photosynthesis might consist of a single production, "If subjected to sunlight, then produce energy." A fine-grained formal theory might take us through a series of microchemical descriptions in which various substances combine and cause various effects. Gross or fine-grained, neither formalism seems to herald the arrival of the silicon tulip. Market gardening has nothing to fear from simulated gardening as yet.

Now, there are properties of plants that are irrelevant to their photosynthetic capacities, e.g., the color of blooms, the shape of leaves (within limits) the height off the ground, and so on. The questions to ask are: What do the chemical properties buy for the plant, and what are the properties of the chemicals by which they buy it? The human brain is made out of a certain physical, chemical stuff. And perhaps in conjunction with other factors, that stuff buys us thought, just as the plants stuff buys it energy. So, what are the properties of the physical chemical stuff of the brain that buy us thought? Here is one answer (not Searle's or that of supporters of Searle's emphasis on stuff, e.g., Maloney [1987]): the vast structural variability in response to incoming exogenous and endogenous stimuli that the stuff in that arrangement provides[7].

Suppose this were so. Might it not also be true that satisfying *some* kinds of formal description guaranteed the requisite structural variability and that satisfying other kinds of formal description did not? Such a state of affairs seems not only possible but pretty well inevitable. But if so, Searle's argument against the formal approach is, to say the least, inconclusive. For the only evidence against the claim that the formal properties of the brain

[7] I am indebted to Aron Sloman for teaching me to put this point in terms of structural variability.

buy it structural variability, which in turn buys it the capacity to sustain thought, is the Chinese-room thought experiment. But in that example the formal description was at a very gross level, in line with the SPSS hypothesis which in this case amounts to rules for correlating inputs, corresponding to sentences of Chinese, with similar outputs. It could well be that a system capable of satisfying that level of formal description need not possess the vast structural variability by which (on my hypothesis) the brain supports thought. This could be neatly tied in with Dreyfus's (1972) (1981) observations that implementations of conventional cognitivist programs are inflexible and lack common sense. Such programs do not depend on a suitably variable and flexible substructure and hence fail to instantiate any understanding whatsoever. (If this talk about suitably variable and flexible substructures seems mysterious, it should become less so once we look at new kinds of computational models of mind: connectionist or PDP models.)

But it might yet prove to be the case that formal descriptions at a *lower,* more microstructural level will have only instantiations that must constitute a system with the requisite structural variability. And as long as this possibility remains, the case for the importance of stuff is far from watertight. Moreover, as we shall see, cognitive science is just beginning to develop formal, microstructural theories that fit this general bill. The price of this maneuver is, of course, grasping Searle's nettle at the other end. If a set of pipes really did constitute a system with the requisite structural variability, then (subject, perhaps, to a few further stipulations) we should welcome it as a fellow thinker. I am ecumenical enough to do this. The more so since, I am reasonably convinced, it is at least physically impossible to secure the relevant variability out of such parts in the actual universe. If there are possible worlds subject to different physical laws than our own and if in those worlds collections of pipes, beer cans or whatever exhibit the relevant fine-grained formal properties (if, for example, they are organized into a value passing network with properties of relaxation, graceful degradation, generalization, and so on), we should bear them no ill will. Some beer cans, it seems, satisfy formal descriptions that our beer cans cannot reach.

9. MICROFUNCTIONALISM.

The defense of a formal approach to mind mooted above can easily be extended to a defense of a form of functionalism against the attacks mounted by Block. An unsurprising result, since Searle's attack on strong AI is intended to cast doubt on any purely formal account of mind, and that attack, as we saw, bears a striking resemblance to the charges of excessive liberalism and absent qualia raised by Block. Functionalism, recall, identified the real essence of a mental state with an input, internal state transition, and output profile. Any system with the right profile, regardless of its size, nature and components, would occupy the mental state in question. But unpromising systems (like the population of China) could, it seemed, be so organized. Such excessive liberalism seemed to undermine functionalism -- surely the system comprising the population of China would not *itself* be a proper subject of experience. The qualia (subjective experience or feels) seem to be nowhere present.

It is now open to us to respond to this charge in the same way we just responded to Searle. It all depends, we may say, on where you locate the grain of the input, internal state transitions, and output. If you locate it at the gross level of a semantically transparent system, then we may indeed doubt that satisfying *that* formal description is a step on the road to being a proper subject of experience. At that level we may expect absent qualia, excessive liberalism, and all the rest. But suppose our profile is much finer-grained and is far removed from descriptions of events in everyday language, perhaps with internal-state transitions specified in a mathematical formalism rather than in a directly semantically interpretable formalism. Then it is by no means so obvious (if it ever was -- see Churchland & Churchland (1981)) either that a system made up of the population of China *could* instantiate such a description or that if it did it would not be a proper subject of the mental ascriptions at issue (other circumstances permitting). My suggestion is that we might reasonably bet on a kind of *microfunctionalism*, relative to which our intuitions about excessive liberalism and absent qualia would show up as more clearly unreliable.

Such a position owes something to Lycan's (1981) defense of functionalism against Block. In that defense he accuses Block of relying on a kind of gestalt blindness (Lycan's term) in which the functional components are made so large (e.g., whole Chinese speakers) or unlikely (e.g., Searle's beer cans) that we rebel at the thought of ascribing intentionality to the giant systems they comprise. Supersmall beings might, of course, have the same trouble with neurons. Lycan, however, then opts for what he calls a homuncular functionalism, in which the functional subsystems are identified by whatever they may be said to do for the agent.

Microfunctionalism, by contrast, would describe at least the internal functional profile of the system (the internal state transitions) in terms far removed from such contentful, purposive characterizations. It would delineate formal (probably mathematical) relations between processing units in such a way that when those mathematical relations obtain, the system will be capable of vast, flexible structural variability and will have the attendant emergent properties. By keeping the formal characterization (and thereby any good semantic interpretation of the formal characterization) at this fine-grained level we may hope to guarantee that any instantiation of such a description provides at least potentially the right kind of substructure to support the kind of flexible, rich behavior patterns required for true understanding. These idea about substructure are fleshed out a little in the next section and in detail in Clark (1989) (1993).

Whether such an account is properly termed a species of functionalism, as I've suggested, is open to some debate. I have opted for a broad notion of functionalism that relates the real essence of thought and intentionality to patterns of nonphysically specified internal state transitions suitable for mediating an input-output profile in a certain general kind of way. This in effect identifies functionalism with the claim that structure, not the stuff, counts and hence identifies it with any formal approach to mind. On that picture, microfunctionalism is, as its name suggests, just a form of functionalism, one that specifies internal state transitions at a very fine-grained level.

Some philosophers, however, might prefer to restrict the "functionalism" label to just those accounts in which (1) we begin by formulating, for *each* individual mental state, a profile of input, internal state transitions, and output in which internal state transitions are described at the level of beliefs, desires, and other mental states of folk psychology, (2) we then replace the folk-psychological specifications by some formal, nonsemantic specification that preserves the boundaries of the folk-psychological specifications. Now there. is absolutely no guarantee that such boundaries will be preserved in a microfunctionalist account. Moreover, though it may, microfunctionalism need not aspire to give a functional specification *of each* type of mental state. (How many are there anyway?) Instead, it might give an account of the kind of substructure needed to support general, flexible behavior of a kind that makes appropriate the ascription to the agent of a whole *host of* folk-psychological states. For these reasons, it may be wise to treat "micro-functionalism" as a term of art and the defense of functionalism as a defense of the possible value of a fine-grained formal approach to mind. I use the terminology I do because I believe the essential motivation of functionalism lies in the claim that what counts is the structure, not the stuff (this is consistent with its roots -- see Putnam (1960) (1967) (1975b)). But who wants to fight over a word? Philosophical disquiet over classical cognitivism, I conclude, has largely been well motivated but at times overambitious. Block and Searle, for example, both raise genuine worries about the kind of theories that seek to explain mind by detailing computational manipulations of standard symbolic atoms. But it is by no means obvious that criticisms that make sense relative to those kinds of computational models are legitimately generalized to all computational models. The claim that structure, not stuff, is what counts has life beyond its classical cognitivist incarnation, as we shall next see.

10. Symbolic Flexibility.

Smolensky (1987) (1988) usefully describes connectionist or PDP models as working in what he calls the subsymbolic paradigm. In the subsymbolic paradigm, cognition is not modeled by the manipulation of machine states that neatly match (or stand for) our daily, symbolic descriptions of mental states and processes. Rather, these high-level descriptions (he cites goals, concepts, knowledge, perceptions, beliefs, schemata, inferences, actions) turn out to be useful labels that bear only approximate relations to the underlying computational structure. He argues that work in the subsymbolic (or distributed connectionist) paradigm aims to do justice to the "real data on human intelligent performance," i.e., to clinical and experimental results, while settling for merely emergent approximations to our high-level descriptive categories. The essential difference between the subsymbolic and the symbolic approach, as Smolensky paints it, concerns the question, Are the semantically interpretable entities the very same objects as those governed by the rules of computational manipulation that define the system?

In the symbolic paradigm, the answer is yes. Consider the STS approach we sketched back in section 5. Here we find computational operations directly applied to high-level descriptions of mental states presented as a means of capturing the computational backdrop of mind. Thus, we might find a model of scientific discovery in which operations are performed on states directly interpretable as standing for particular hypotheses concerning the laws governing some data. Against this kind of approach, the subsymbolic theorist urges that the entities whose behavior is governed by the rules of computational manipulation that define the system need not share the semantics of the task description. For what is so governed is just the activation profiles of individual units in a connectionist network[8]. And in a highly distributed model these units in the end will have no individual semantic interpretation, or at least none that maps neatly and projectibly onto our ordinary concepts of the entities to be treated in a model of the processing involved. Rather, what gets semantically interpreted will be general patterns of activation of such units. A single high-level concept like that of a kitchen or a ball may thus be associated with a continuum of activation patterns corresponding to the subtly different ideas about kitchen or ball that we entertain in various circumstances. Smolensky puts it nicely in the following passage:

> In the symbolic approach, symbols (atoms) are used to denote the semantically interpretable entities (concepts). These same symbols are the objects governed by symbol manipulations in the rules which define the system. The entities which are capable of being semantically interpreted are also the entities governed by the formal laws that define the system. In the subsymbolic paradigm, this is no longer true. The semantically interpreted entities are patterns of activation over a large number of units in the system, whereas the entities manipulated by formal rules are the individual activations of cells in the network. The rules take the form of activation passing rules, of essentially different character from symbol manipulation rules. (1978), p. 100)

The claim, in effect, is that PDP systems need not (and typically will not) be semantically transparent in the sense introduced earlier. Such a claim may not seem immediately

[8] For an introduction to connectionist models, see e.g., Clark (1989) (1993), Smolensky (1988).

plausible for the following reason. A system will count as semantically transparent just in case the entities found in a top-level task analysis of what the system does have neat syntactic analogues whose behavior is governed by the computational rules (explicit *or* tacit) of the system. Now clearly, it will not do to say that just because individual units cannot be treated as the syntactic analogues of such entities (e.g, as "coffee," "ball," "kitchen," and so on) the condition fails to be met. For why not treat patterns of activation of such units as the required analogues? The behavior of such patterns surely is governed by the computational rules of the system.

This is where the requirement that such analogues be *projectible* comes in. Consider a sentence like "the ball broke the window." A conventional AI system dealing with such a sentence will have a syntactic analogue (first in, say, LISP and hence down to machine code) for "ball" and "window." Consider now a connectionist representation (such as that developed in McClelland & Kawamoto (1986)) of the same sentence. There will be a pattern of active units, and it may well be possible to nonarbitrarily isolate a subset of that pattern that, we would like to say, stands for "ball". But that subpattern, it is important to note, will vary from context to context. "Ball" as it occurs in "The ball broke the window@ will have a different (though doubtless partially overlapping) syntactic analogue to "ball" as it occurs in "the baby held the ball." In one case, hardness related "microfeatures" will be active. In the other case, not. Thus, although in each individual case we can isolate a connectionist syntactic analogue for the entities spoken of in a conceptual account, these entities will not be neatly projectible, i.e., the same syntactic entity will not continue to correlate in other cases with the top-level semantic entity. This is the real sense in which PDP systems can constitute a move away from semantic transparency.

Examples could be multiplied. Smolensky (1988) makes similar comments about the symbol "coffee" as it occurs in various contexts. The general point, then, is that "the context [in PDP systems] alters the internal structure of the symbol: the activities of the sub-conceptual units that comprise the symbol -- its subsymbols -- change across contexts@ (Smolensky (1988), p. 17). Smolensky formalizes this point as a characteristic of the highly distributed PDP systems he is interested in as follows: "In the symbolic paradigm the context of a symbol is manifest *around* it and consists of *other symbols;* in the subsymbolic paradigm the context of a symbol is manifest *inside* it, and consists of subsymbols" (Smolensky (1988), p. 17). Both the intrinsic holism and flexibility of PDP systems can be seen to flow from this fact.

11. GRADES OF SEMANTIC TRANSPARENCY.

Using this apparatus as a base, Smolensky (1988) formulates an interesting picture of the cognitive terrain. He suggests that some human knowledge (e.g., public scientific knowledge) exists in the first instance as linguistic items such as the principle "energy is conserved." Human beings, he suggests, may use such knowledge by deploying a virtual machine adapted to manipulate analogues of such linguistic representations. Such explicitly formulated knowledge he calls "cultural knowledge." The "top-level conscious processors@ of an individual is precisely, he argues, a virtual machine adapted to that end. This machine, which is realized by a PDP substructure, he calls the "conscious rule interpreter." It is contrasted with what he calls the "intuitive processor." The distinction again depends on the kind of entities processed. The conscious rule interpreter actually takes as its syntactic objects the semantic entities we use in describing the task domain (e.g., "energy"). The intuitive processor, by contrast, takes as its objects distributed microfeatural representations. These representations, we saw, bear only a fluid and shifting relationship to the semantic entities (like "coffee" and "ball") spoken of at the conceptual level. It thus follows that the programs running on the conscious rule interpreter have a syntax and semantics comparable to our top-level articulation of the domain. (This is no accident: they are precisely models of that top-level articulation.) While the programs running on the intuitive processor do not. In my terminology, programs running on the conscious rule interpreter will be semantically transparent, and the semantics will seep neatly down to the formal level, while those running on the intuitive processor will not. The intuitive processor is quite clearly to be seen as the more evolutionarily basic of the two and is responsible (he says) for all animal behavior and much human behavior, including: "perception, practiced motor behavior, fluent linguistic behavior, intuition in problem solving and game-playing. In short, practically all of skilled performance" (Smolensky (1988), p.5).

There need not, however, be an all-or-nothing divide between the semantically transparent processing of the conscious rule interpreter and the semantically opaque processing of the intuitive processor. For the cognitive system itself is presumed to be at root a subsymbolic system that, to a greater or lesser degree in various cases, *approximates* to the behavior of a symbolic system manipulating conceptual entities. The greater the so-called dimension shift between the conceptual description and the semantic interpretation of the in the network, the rougher such an approximation becomes.

Another route to the approximation claim is to regard the classical accounts as describing the *competence* of a system, i.e., its capacity to solve a certain range of well-posed problems (see Smolensky (1988), p. 19). In idealized conditions (sufficient input data, unlimited processing time) the PDP system will match the behavior specified by the competence theory. But outside that idealized domain of well-posed problems and limitless processing time, the performance of a PDP system will diverge from the predictions of the competence theory in a pleasing way. It will give sensible responses even on receipt of degraded data or under severe time constraints. This is because although describable in that idealized case as satisfying hard constraints, the system may actually operate by satisfying a multitude of soft constraints. Smolensky here introduces an analogy with Newtonian mechanics. The physical world is a quantum system that looks Newtonian under certain conditions. Likewise with the cognitive system. It *looks* increasingly classical as we approach the level of conscious rule following. But in fact, according to Smolensky, it is a PDP system through and through.

In the same spirit Rumelhart and McClelland suggest: 'It might be argued that conventional symbol processing models are macroscopic accounts, analogous to Newtonian mechanics, whereas our models offer more microscopic accounts, analogous to quantum theory.... Through a thorough understanding of the relationship between the Newtonian mechanics and quantum theory we can understand that the macroscopic level of description may be *only an approximation* to the more microscopic theory" (Rumelhart & McClelland (1986), p. 125). To illustrate this point, consider a simple example due to Paul Smolensky. Imagine that the cognitive task to be modeled involves answering qualitative questions on the behavior of a particular electrical circuit. (The restriction to a single circuit may appall classicists, although it is defended by Smolensky on the grounds that a small number of such representations may act as the chunks utilized in general purpose expertise -- see Smolensky (1986), p. 241.) Given a description of the circuit, an expert can answer questions like "If we increase the resistance at a certain point, what effect will that have on the voltage, i.e., will the voltage increase, decrease, or remain the same?"

Suppose, as seems likely, that a high-level competence-theoretic specification of the information to be drawn on by an algorithm tailored to answer this question cites various laws of circuitry in its derivations (what Smolensky refers to as the Ahard laws@ of circuitry: Ohm's law and Kirchoffs' law). For example, derivations involving Ohm's law would invoke the equation

voltage = current x resistance.

How does this description relate to the actual processing of the system? The model represents the state of the circuit by a pattern of activity over a set of feature units. These encode the qualitative changes found in the circuit variables in training instances. They encode whether the overall voltage falls, rises, or remains the same when the resistance at a certain point goes up. These feature units are connected to a set of what Smolensky calls "knowledge atoms," which represent patterns of activity across subsets of the featured units. These in fact encode the legal combinations of feature states allowed by the actual laws of circuitry. Thus, for example, "the system's knowledge of Ohm's law ... is distributed over the many knowledge atoms whose subpattems encode the legal feature combinations for current, voltage and resistance" (Smolensky (1988), p. 19). In short, there is a subpattern for every legal combination of qualitative changes (65 subpatterns, or knowledge atoms, for the circuit in question).

At first sight, it might seem that the system is merely a units-and connections implementation of a lookup table. But that is not so. In fact, connectionist networks act as lookup tables only when they are provided with an overabundance of hidden units and hence can simply memorize input-output pairings. By contrast, the system in question encodes what Smolensky terms "soft constraints," i.e., patterns of relations that usually obtain between the various feature units (microfeatures). It thus has general knowledge of qualitative relations among circuit microfeatures. But it does *not* have the general knowledge encapsulated in hard constraints like Ohm's law. The soft constraints are two-way connections between feature units and knowledge atoms, which *incline* the network one way or another but do not *compel* it, that is, they can be overwhelmed by the activity of other units (that's why they are soft). And as in all connectionist networks, the system computes by trying simultaneously to satisfy as many of these soft constraints as it can. To see that it is not a mere lookup tree of legal combinations, we need only note that it is capable of giving sensible answers to (inconsistent or incomplete) questions that have no answer in a simple lookup table of legal combinations.

The soft constraints are numerically encoded as weighted inter-unit connection strengths. Problem solving is thus achieved by "a series of many node updates, each of which is a *microdecision* based on formal numerical rules and numerical computations" (Smolensky (1986), p. 246).

The network has two properties of special interest to us. First, it can be shown that if it is given a well-posed problem and unlimited processing time, it will always give the correct answer as predicted by the hard laws of circuitry. But, as already remarked, it is by no means bound by such laws. Give it an ill-posed or inconsistent problem, and it will satisfy as many as it can of the soft constraints (which are all it really knows about). Thus "outside of the idealized domain of well-posed problems and unlimited processing time, the system gives sensible performance" (Smolensky (1988), p. 19). The hard rules (Ohm's law, etc.) can thus be viewed as an external theorist's characterization of an idealized subset of its actual performance (it is no accident if this brings to mind some of Dennett's claims about the intentional stance -- see Dennett (1981)).

Second, the network exhibits interesting *serial* behavior as it repeatedly tries to satisfy all the soft constraints. This serial behavior is characterized by Smolensky as a set of *macrodecisions* each of which amounts to a "commitment of part of the network to a portion of the solution." These macrodecisions, Smolensky notes, are "approximately like the firing of production rules. In fact, these 'productions' fire in essentially the same order as in a symbolic forward-chaining inference system" (Smolensky (1988), p. 19). Thus, the network will look as if it is sensitive to hard, symbolic rule at quite a fine grain of description. It will not *simply* solve the problem "in extension" as if it knew hard rules. Even the *stages* of problem solving may look as if they are caused by the system's running a processing analogue of the steps in the symbolic derivations available in the competence theory.

But the appearance is an illusion. The system has no knowledge of the objects mentioned in the hard rules. For example, there is no neat subpattern of units that can be seen to stand for the general idea of resistance, which figures in Ohm's law. Instead, some sets of units stand for resistance at R1, and other sets for resistance at R2. In more complex networks the coalitions of units that, when active, stand in for a top-level concept like resistance are highly context-sensitive. That is, they vary according to context of occurrence. Thus to use Smolensky's own example, the representation of coffee in such a network would not consist of a single recurrent syntactic item but a coalition of smaller items (microfeatures) that shift according to context. Coffee in the context of a cup may be represented by a coalition that includes the features (liquid) and (contacting-porcelain). Coffee in the context of jar may include the features (granule) and (contacting-glass). There is thus only an "approximate equivalence of the 'coffee vectors' across contexts," unlike the "exact equivalence of the coffee tokens across different contexts in a symbolic processing system" (Smolensky (1988), p. 17). By thus replacing the conceptual symbol "coffee" with a shifting coalition of microfeatures, the so-called dimension shift, such systems deprive themselves of the structured mental representations deployed in both a classical competence theory and a classical symbol processing account (level 2). Likewise, in the simple network described, there is no stable representation that stands for resistance.

It seems, then, that by treating subsymbolically the entities spoken of in our conceptual-level descriptions, we buy the flexibility, shading, and general lack of rigidity and brittleness required of a system if its subsequent behavior is ever to warrant the ascription too it of a genuine grasp of concepts. Symbolic flexibility of understanding is brought about by the increased low-level variability of the PDP approach. In this way such systems may avoid the excessive rigidity and lack of insight endemic to conventional AI. Notice also that subsymbolic models remain formal in the sense outlined in sections 1 through 6, and that this is a microfunctionalist theory as defined in section 9. That is, it specifies a system only in terms of input-output profiles for individual units and thus is not crucially dependent on any particular biological substrate. But the entities figuring in the formal profile do not correspond to, or otherwise nearly preserve, the boundaries of any conceptual-level description of thought.

12. CONCLUSIONS.

Connectionism, I have tried to show, is an existence proof of the possibility of a kind of *microfunctionalist* approach to understanding the mind. Connectionism remains committed to the basic formalist belief that a certain syntactic organization is sufficient for mentality. But it describes that formal organization in a much finer-grained way than (most) traditional AI. In particular, it does not bind its syntactic elements directly to the conceptual items characteristic of conscious thought and its linguistic expressions. Such an approach, I now believe[9], needs to be combined with a better appreciation of the way bodily dynamics, motion, and environment-manipulating action alter and transform the problem-spaces that confront biological brains. Cognition and mind are then best understood by further pursing the microfunctionalist approach in the special contexts of robotics, autonomous agent theory, and neuroethology[10].

[9] See Clark (1997).

[10] Two good introductions are Langton (1995), Boden (1996).

REFERENCES.

Armstrong, D. (1970). The nature of mind. In N. Block (Ed.), *Readings in Philosophy of Psychology* (Vol. 1, pp. 191-199). London: Methuen & Company.

Block, N. (1980). Troubles with functionalism. In N. Block (Ed.), *Readings in Philosophy of Psychology* (Vol. 1, pp. 268-305). London: Methuen & Company.

Boden, A. (1996). *The Philosophy of Artificial Life*. Oxford: Oxford University Press.

Charniak, E., & McDermott, D. (1985). *Introduction to Artificial Intelligence*. Reading, MA: Addison-Wesley.

Churchland, P. (1981). Eliminative materialism and the propositional attitudes. *Journal of Philosophy, 78*(2), 67-90.

Churchland, P. (1986). *Neurophilosophy: Towards a Unified Theory of the Mind-Brain*. Cambridge, MA: MIT Press.

Churchland, P., & Churchland, P. (1981). Functionalism, qualia, and intentionality. In J. Biro & R. Shahan (Eds.), *Mind, Brain, and Function* . Oklahoma: University of Oklahoma Press.

Clark, A. (1986). A biological metaphor. *Mind and Language, 1*(1), 45-64.

Clark, A. (1989). *Microcognition: Philosophy, Cognitive Science and Parallel Distributed Processing*. Cambridge: MIT Press.

Clark, A. (1993). *Associative Engines: Connectionism, Concepts and Representational Change*. Cambridge: MIT Press.

Clark, A. (1997). *Being There: Putting Brain, Body and World Together Again*. Cambridge, MA: MIT Press.

Dennett, D. (1981). *Brainstorms*. Sussex: Harvester Press.

Dreyfus, H. (1972). *What computers can't do*. New York: Harper & Row.

Dreyfus, H. (1981). From micro-worlds to knowledge representation: AI at an impasse. In J. Haugeland (Ed.), *Mind Design* (pp. 161-205). Cambridge, MA: MIT Press.

Feigenbaum, E. (1977). The art of intelligence: 1. Themes and case studies of knowledge engineering. *Proceedings of the fifth International Joint Conference on Artificial Intelligence, 11*, 1014-1029.

Fodor, J. (1968). The appeal to tactic knowledge in psychological explanation. *Journal of Philosophy, 65*, 627-640.

Fodor, J., & Pylyshyn, Z. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition, 28*, 3-71.

Hallam, J., & Mellish, C. (Eds.). (1987). *Advances in Artificial Intelligence* . Chichester: Wiley & Sons.

Haugeland, J. (1985). *Artificial Intelligence: The Very Idea*. Cambridge: MIT Press.

Hodges, A. (1983). *Alan Turing: The Enigma*. New York: Simon and Schuster.

Hofstadter, D. (1985). *Metamagical Themas: Questing for the Essence of Mind and Pattern*. Middx, London: Penguin.

Langley, P. (1979). *Rediscovering Physics with BACON 3*. Paper presented at The Sixth International Joint Conference on Artificial Intelligence.

Langley, P., Simon, H., Bradshaw, G., & Zytkow, J. (1987). *Scientific Discovery: Computational Explorations of the Creative Process*. Cambridge, MA: MIT Press.

Langton, C. (Ed.). (1995). *Artificial Life: An overview* . Cambridge, MA: MIT Press.

Lenat, D. (1983a). EURISKO: A program that learns new heuristics and domain concepts. *Artificial Intelligence, 21*, 61-98.

Lenat, D. (1983b). Theory formation by heuristic search. *Artificial Intelligence, 21*, 31-59.

Lycan, W. (1981). Form, function and feel. *Journal of Philosophy, 78*(1), 24-50.

Maloney, J. (1987). The right stiff. *Synthese, 70*, 349-372.

Marr, D. (1977). Artificial Intelligence: A personal view. In J. Haugeland (Ed.), *Mind Design* (pp. 129-142). Cambridge, MA: MIT Press.

Marr, D. (1982). *Vision*. San Francisco, CA: W.H. Freeman.

McClelland, J., & Kawamoto, A. (1986). Mechanisms of sentence processing: Assigning roles to constituents of sentences. In J. McClelland, D. Rumelhart, & P. R. Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Vol. II, pp. 216-271). Cambridge, MA: MIT Press.

Newell, A., & Simon, H. (1976). Computer Science as Empirical Enquiring. In J. Haugeland (Ed.), *Mind Design* . Cambridge: MIT Press.

Putnam, H. (1960). Minds and machines. In S. Hook (Ed.), *Dimensions of Mind* . New York: New York University Press.

Putnam, H. (1967). Psychological Predicates. In W. Capitan & D. Merill (Eds.), *Art, Mind and Religion* : University of Pittsburgh Press.

Putnam, H. (1975a). The meaning of "meaning". In H. Putnam (Ed.), *Mind, Language and Reality* (pp. 215-275). Cambridge: Cambridge University Press.

Putnam, H. (1975b). Philosophy and our mental life. In H. Putnam (Ed.), *Mind, Language, and Reality* (pp. 291-303). Cambridge: Cambridge University Press.

Pylyshyn, Z. (1986). *Computation and cognition*. Cambridge, MA: MIT Press.

Ricthie, G., & Hanna, F. (1984). AM: A case study in AI methodology. *Artificial Intelligence, 23*, 249-268.

Rumelhart, D., Smolensky, P., McClelland, J., & Hinton, G. (1986). Schemata and sequential thought processes in PDP models. In J. McClelland, D. Rumelhart, & P. R. Group (Eds.), *Parallel Distributed Processing: Explorations in the microstructure of cognition* (Vol. II, pp. 7-57). Cambridge, MA: MIT Press.

Ryle, G. (1949). *The Concept of Mind*. London: Hutchinson.

Schank, R., & Abelson, R. (1977). *Scripts, plans, goals, and understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Searle, J. (1980). Minds, brains and programs. In J. Haugeland (Ed.), *Mind Design* (pp. 282-307). Cambridge, MA: MIT Press.

Searle, J. (1983). *Intentionality*. Cambridge: Cambridge University Press.

Searle, J. (1984). Intentionality and its place in nature. *Synthese, 61*, 3-16.

Shortlife, E. (1976). *Computer Based Medical Consultations: MYCIN*. New York: Elsevier.

Simon, H. (1979). Artificial intelligence research strategies in the light of AI models of scientific discovery. *Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 2*, 1086-1094.

Sloman, A. (1984). The structure of the space of possible minds. In S. Torrence (Ed.), *The Mind and the Machine* . Sussex: Ellis Horwood.

Smart, J. (1959). Sensations and brain processes. *Philosophical Review, 68*, 141-156.

Smolensky, P. (1987). Connectionist AI, and the brain. *Artificial Intelligence Review, 1*, 95-109.

Smolensky, P. (1988). On the Proper Treatment of Connectionism. *Behavioral and Brain Sciences, 11*, 1-74.

Turing (1936). On Computable Numbers, with an Application to the Entscheidung Problem. *Proceedings of the London Mathematical Society*, *2*(42), 230-265.

Turing, A. (1950). Computing machinery and intelligence. *Mind*, *LIX*(2236), 423-460.