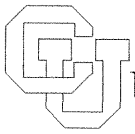
Connectionism, Constituency, and the Language of Thought

Paul Smolensky

CU-CS-416-88



University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

To appear in *Fodor and his critics*,
B. Loewer & G. Rey (Eds.); Blackwell's.

Connectionism, Constituency, and the Language of Thought

Paul Smolensky

CU-CS-416-88 November, 1988

Department of Computer Science &
Institute of Cognitive Science
University of Colorado
Boulder, CO 80309–0430

Abstract

In their paper, "Connectionism and Cognitive Architecture," Fodor and Pylyshyn (1988) argue that connectionism cannot offer a cognitive architecture that is both viable and different from the Classical language of thought architecture: if it differs from the Classical architecture it is because it reinstantiates simple associationism, and is therefore not a viable candidate; if it is viable, it is because it implements the Classical view and therefore does not offer a new cognitive architecture—just a new implementation of the old one. It is my purpose here to expose the false dichotomy in this argument, to show that the space of connectionist cognitive architectures is much richer than this simple dichotomy presumes, and that in this space is a large region of architectures that are implementations neither of a Classical architecture nor of a simple associationist architecture; these architectures provide structured mental representations and structure-sensitive processes in a truly non-Classical way.

In Section 1, I make a number of general remarks; in Section 2, I focus on the crux of their argument, which turns on the compositional structure of mental states. I develop in some detail the argument that, unlike simple associationist models, connectionist models using *distributed representations* can embody compositionality at the same time as providing a new cognitive architecture that is not an implementation of a Classical language of thought. In Section 3, I argue that the debate surrounding compositionality illustrates the general point that by finding new formal instantiations of basic computation notions in the category of continuous mathematics, connectionism can open up genuinely new and powerful accounts of computation and cognition that go well beyond the limited progress that can be afforded by the kind of implementationalist strategy that Fodor and Pylyshyn advocate.

Connectionism, Constituency, and the Language of Thought

Paul Smolensky

Department of Computer Science & Institute of Cognitive Science
University of Colorado at Boulder

In their paper, "Connectionism and Cognitive Architecture," Fodor and Pylyshyn (1988) argue that connectionism cannot offer a cognitive architecture that is both viable and different from the Classical language of thought architecture: if it differs from the Classical architecture it is because it reinstantiates simple associationism, and is therefore not a viable candidate; if it is viable, it is because it implements the Classical view and therefore does not offer a new cognitive architecture—just a new implementation of the old one. It is my purpose here to expose the false dichotomy in this argument, to show that the space of connectionist cognitive architectures is much richer than this simple dichotomy presumes, and that in this space is a large region of architectures that are implementations neither of a Classical architecture nor of a simple associationist architecture; these architectures provide structured mental representations and structure-sensitive processes in a truly non-Classical way.

In Section 1, I make a number of general remarks about connectionism, Fodor and Pylyshyn's argumentation, and the abuse of the term "implementation." In Section 2, I focus on the crux of their argument, which turns on the compositional structure of mental states. I develop in some detail the argument that, unlike simple associationist models, connectionist models using *distributed representations* can embody compositionality at the same time as providing a new cognitive architecture that is not an implementation of a Classical language of thought. In Section 3, I bring together the more technical discussion of Section 2 back in contact with the more general issues raised in Section 1. I argue that the debate surrounding compositionality illustrates the general point that by finding new formal instantiations of basic computation notions in the category of continuous mathematics, connectionism can open up genuinely new and powerful accounts of computation and cognition that go well beyond the limited progress that can be afforded by the kind of implementationalist strategy that Fodor and Pylyshyn advocate.

1. General Remarks

1.1. In-principle arguments

It is worth stating up front that, in my opinion, the highly general, *negative*, in-principle arguments one finds on the issue of cognitive architecture—like that of Fodor & Pylyshyn (1988; henceforth, F&P)—are stimulating and useful but quite inconclusive; this is as true of anti-Classical arguments as it is of anti-connectionist arguments. On the other hand, *positive* in-principle arguments can serve the valuable role of showing how to conceive of previously inconceivable accounts; they serve as programmatic statements of research agenda. It was in this spirit, for example, that Smolensky (1988a) was written; it was intended not as an in-principle attack on the Classical view but as a positive in-principle argument for a fairly comprehensive connectionist framework for cognitive science.

This issue of negative vs. positive in-principle arguments bears strongly on Section Four of F&P, which proceeds basically on the following plan:

(1) a. Collect many positive arguments for connectionism.
b. Present them as negative in-principle arguments against the Classical account.
c. Show that, as negative arguments, they don't hold up.

It is interesting to note the shift from an offensive to a defensive posture here. Whereas arguments from advocates of the Classical view used to be of the form, "the trouble with connectionism is that it can't do *X*," increasingly we now hear arguments of the form "well, Classical architectures can do *X* too." In Section Four, F&P offer us about a dozen defensive arguments of just this form, where *X* ranges from graceful degradation and massive parallelism to inexplicit rules and soft constraint satisfaction.

The moral seems to be that arguments of the form "formalism *Y* can't do *X*" are sometimes provocative but, at least when *Y* is a general, powerful formalism like symbolic or connectionist computation, such arguments are almost virtually sure to fail. Arguments standing a chance of surviving that concern such features as massive parallelism, soft constraints, and neural plausibility, must be positive arguments that connectionism achieves these particularly well, not negative arguments that the symbolic approach can't achieve them at all.

1.2. The argument structure

In their offensive attack, F&P follow an argument structure frequently used in attacks on connectionism:

(2) a. Propose a simplistic representation and process.
b. Claim this is what connectionism advocates.
c. Point out the inadequacies of the proposal.
d. Claim that any improvement requires abandoning connectionist commitments, and turns the proposal into a "Classical account" or an "implementation" of one.

This argument can be made rather cheaply through three tricks:

(3) a. Make the proposed representation and process as simple-minded as possible.
b. Bloat the territory covered by the term "Classical account" as much as possible.
c. Bloat the relations covered by the term "implementation" as much as possible.

Each of these tricks are used in good measure by F&P, as by other critics of connectionism, and in this response I will have to call them on all three accounts.

For F&P, the straw man required for step (2a) is provided, of course, by simple associationism. In order to expose F&P's trivialization of the real issue, I will argue

(4) a. that the true commitment of connectionism is not to simple associationism, but to something else that can be clearly distinguished from that to which the Classical view is committed;
b. that this less simple-minded view of connectionism provides for ways of instantiating the compositional structure of mental states; and
c. that these connectionist instantiations of compositionality do not all reduce to "an implementation of the Classical view," and in fact offer a new candidate for the cognitive architecture.

1.3. The true commitment of connectionism: PTC version

In this paper I adopt a view of connectionism that was presented and discussed at some length in Smolensky (1988a,b), a view I call *PTC* (for the Proper Treatment of Connectionism). Oversimplifying a bit, according to PTC, the true commitment of connectionism is to a very general formalism for describing mental representations and mental processes. The Classical view is of course committed to the hypothesis that mental representations are elements of a *symbol system*, and that mental processes consist of symbol manipulation operations. PTC is committed to the hypothesis that mental representations are *vectors* partially specifying the state of a dynamical system (the activities of units in a connectionist network), and that mental processes are specified by the differential equations governing the evolution of that dynamical system.

The main point is this: under the influence of the Classical view, computation and cognition have been studied almost exclusively under the umbrella of discrete mathematics; the connectionist approach, on the other hand, brings the study of computation and cognition squarely in contact with the other half of mathematics—continuous mathematics. The true commitment, according to PTC, is to uncovering the insights this other half of mathematics can provide us into the nature of computation and cognition.

On this account, simple associationism is a particularly impoverished and impotent corner of the connectionist universe. It may well be that the attraction a number of people feel to connectionism is an attraction to neo-associationism; but it is nonetheless a serious mistake to presume connectionism to be committed to simple associationist principles. To equate connectionism with simple associationism is no more appropriate than equating Classical symbolic theory with Aristotelean logic. (The temptation Fodor may provide his readers notwithstanding, I don't recommend the second identification any more than the first.)

In fact, the comparison with Aristotle is not wholly inappropriate. Our current understanding of the power of connectionist computation might well be compared with Aristotle's understanding of symbolic computation; before connectionism can take really serious shots at cognitive modeling, we probably have at least as far to go in developing connectionist computation as symbolic computation had to go between Aristotle and Turing. In giving up symbolic computation to undertake connectionist modeling, we connectionists have taken out an enormous loan, on which we are still paying nearly all interest: solving the basic problems we have created for ourselves rather than solving the problems of cognition. In my view, the loan is worth taking out for the goal of understanding how symbolic computation, or approximations to it, can emerge from numerical computation in a class of dynamical systems sharing the most general characteristics of neural computation.

Under this characterization of the commitments of the Classical and connectionist approach, to claim, as F&P explicitly do, that any cognitive architecture that incorporates structured mental representations and processes sensitive to that structure is a Classical Architecture, is to bloat the notion of "Classical Architecture" to an unacceptable degree—in accord with (3b).

1.4. Implementation vs. refinement

The bottom line of F&P can be paraphrased as follows. "*Standard* connectionism is just simple associationism wrapped in new jargon, and as such, is fatally flawed. Connectionists should pursue instead a *nonstandard* connectionism, embracing the principles of compositionality and structure-sensitive processing: they should accept the Classical view and should design their nets to be implementations of Classical architectures." Behind this moral is the assumption that connectionist models with compositionally structured representations must necessarily be implementations of a Classical architecture; it will be my major purpose to show that this is false. The connectionist systems I will advocate hypothesize models that are not an *implementation* but rather a *refinement* of the Classical symbolic approach; these connectionist models hypothesize a truly different cognitive architecture, to which the Classical architecture is a scientifically important approximation. The reader may suspect that I will be splitting hairs and that the difference between "implementation" and "refinement" will be of no philosophical significance. But in fact the new cognitive architecture I will hypothesize lacks the most crucial property of Fodor & Pylyshyn's Classical architecture: mental representations and mental processes are *not* supported by the same formal entities—there are no "symbols" that can do both jobs.[1] The new cognitive architecture is fundamentally two-level: formal, algorithmic specification of processing mechanisms, on the one hand, and semantic interpretation, on the other, must be done at two different levels of description.

1.4.1. Bloating "implementation"

There is a sense of "implementation" that cognitive science has inherited from computer science, and I propose that we use it. If there is an account of a computational system at one level and an account at a lower level, then the lower one is an *implementation* of the higher one if and only if the higher description is a complete, precise, algorithmic account of the behavior of that system. It is *not* sufficient that the higher-level account provide some sort of rough summary of the interactions at the lower level. It is *not* sufficient that the lower-level account involves some of the same basic ideas of how the problem is to be solved (for example, a decomposition of the problem into subproblems). Such weak usages of "implementation" abound in the literature, particularly in the numerous attempts to dismiss connectionism as "mere implementation"—following (3c). But in its correct usage, *implementation* requires that the higher-level account provide an exact, precise, algorithmic account of the system's behavior.

It's important to see that, unless this definition of implementation is adopted, it is impossible to legitimately argue to F&P's ultimate conclusion: as long as connectionists are doing implementation, they're not going to provide a new cognitive architecture. If it is shown only that connectionism "implements" the Classical Architecture under a looser definition of the term, then the conclusion that follows is that the Classical account

1. This point is brought out nicely in Cummins and Schwarz (1987), Schwarz (1987), and Cummins (1988).

provides a rough, higher-level approximation to the connectionist account, or involves some of the same basic ideas about how information is represented and processed. This is a *much weaker* conclusion than what F&P are after. They want the conclusion that only true implementation will license: since the Classical account provides a complete, precise, algorithmic account of the cognitive system, there is nothing to be gained by going to the lower level account, as long as the phenomena of interest can be seen at the higher level; and, of course, it is exactly those phenomena that the Classicist will count as "truly cognitive." To account for intrinsically lower-level phenomena—in which category the Classicist will certainly include neural phenomena and may also include certain perceptual/motor phenomena—the Classicist will acknowledge the need to condescend to a lower level account; but within the domain of "pure cognition," Classicists won't need to get their hands so dirty. These are the sorts of conclusions that Classicists have pushed for decades on the basis of analogies to higher- and lower-level computer languages. But of course these languages, *by design*, satisfy the *correct* definition of implementation; none of these conclusions follow from weaker definitions, and none follow from the connectionist position I defend here. Far from the conclusion that "*nothing* can be gained from going to the lower level account," there is *plenty* to be gained: completeness, precision, and algorithmic accounts of processing, none of which are in general available at the higher level, according to PTC.

Since F&P's conclusions cannot be licensed under any definition of "implementation" weaker than the correct one, it is that one I will use. I will show that distributed connectionism can, without implementing the Classical architecture, meet the basic demands F&P have outlined for a cognitive architecture. To repeat, then:

(5) X is implemented by Y if and only if X provides a complete, precise algorithmic higher-level account of the system described at a lower level by Y.

Alternatively, in place of the word "algorithmic" I often use "formal" to avoid the discrete, sequential connotations the former term can carry.

1.4.2. A two-level cognitive architecture

To see how the distributed connectionist architecture differs fundamentally from the Classical one—fails to provide an "implementation" using the correct definition of the term—I will now sketch how the connectionist architecture is intrinsically split over two levels of description. We'll consider the purest case: distributed connectionist models having the following two properties:

(6) a. Interpretation can be assigned to large-scale activity patterns but not to individual units;
b. The dynamics governing the interaction of individual units is sufficiently complex that the algorithm defining the interactions of individual units cannot be translated into a tractably-specified algorithm for the interaction of whole patterns.[2]

As a result of these two properties, we can see that there are two levels of analysis with very different characteristics. At the lower level, where the state variables are the activities of individual units, the processing is described by a complete, precise, and formal algorithm, but semantic interpretation cannot be done. At the higher level, where the system's state is described in terms of the presence of certain large-scale patterns, semantic interpretation can be done, but now complete, precise algorithms for the processing cannot be stated. As I have characterized this in Smolensky (1988a), the *syntax* or processing algorithm strictly resides at the lower level, while the *semantics* strictly resides at the upper level. Since both the "syntax" and the semantics are essential to the cognitive architecture, we have an intrinsically split-level cognitive architecture here: There is no account of the architecture in which the same elements carry both the syntax and the semantics. Thus we have a fundamentally new candidate for the cognitive architecture which is simply *not* an implementation of the Classical one.[3]

---

2. The complexity criterion here is very low: the interactions should be more complex than purely linear. A lengthy and hopefully accessible discussion may be found in Smolensky, 1986.

3. The criterion of "tractability" in (6b) is important here. For the purpose of identifying the scientific principles of cognitive science, tractable descriptions that provide understanding and make scientific explanation feasible are essential. Thus, while we might be able *in principle* to take the equations describing the interactions of individual units and rewrite them in some very com-

Note that the conclusions of this section depend crucially on the assumption (6a) that connectionist representations are *distributed* (when viewed at the level of individual units, the level at which processing algorithms can be identified (6b)). Thus, while F&P and others may attempt to give the impression that the issue of local vs. distributed representations is a little technical squabble between connectionists of no philosophical consequence, I believe this to be profound mistake. Distributed representations, when combined with (6b), entail that in the connectionist cognitive architecture, mental representations bear a fundamentally different relation to mental processes than is true in the Classical account. I will return to this crucial point in Section 3.

1.5. Summary

As indicated in Sections 1.3 and 1.4, I am arguing that what the Classical/connectionist debate should really be about is

(7) a. whether mental representations and processes are to be formally described by symbol systems, within the category of discrete mathematics; and
b. whether in the cognitive architecture the semantics of mental representation and the algorithms of mental processes both derive from the same formal entities and reside at a single level of description.

If this is correct, then any attempt to cast the connectionist position as either being defined by a commitment to simple associationism or as "merely implementing" the Classical Architecture must rely on oversimplifying the true commitment of connectionism, or to bloating either the terms "Classical Architecture" or "implementation" to a truly misleading and unacceptable degree. F&P have made all three of these moves.

2. Compositionality and distributed connectionist representations

In this section I consider the crux of F&P's argument, and argue that distributed connectionist architectures, without implementing the Classical architecture, can nontheless provide structured mental representations and mental processes sensitive to that structure.

2.1. The ultralocal case

Here is a quick summary of what I take to be the central argument of F&P.

---

plicated way so as to apply to large-scale activity patterns, unless those new equations are usable *in practice*, we really have no choice but to ground our understanding and explanation of processing in the lower level. Similarly, even if meaningful semantic interpretation is limited to the higher level, we can always *define* a unit to *represent* all situations in which it's active, but that's a useless definition unless those situations form some meaningful class in the domain of interpretation; in the kind of model we're now considering, by assumption that doesn't happen (or only very rarely), and as a result, we have no choice but to ground our semantic accounts in the higher level.

In general, these non-tractable descriptions of semantics or syntax on the wrong level incorporate complex transformations of the variables that effectively exploit the other level without admitting it. (A fairly general case is worked out in Smolensky, 1986). As an analogy, it is in principle possible to do predicate calculus using not the usual symbolic representation of formulæ, but their Gödel number representations; but the steps in going from one Gödel number to the other are in general intractably complex because they secretly involve transforming back to something isomorphic to the symbolic representation, doing the manipulation there, and transforming back to Gödel numbers. To say that logic can be done directly on Gödel numbers without reference to symbolic representations may be true in some sense; and in that same sense it may be possible to "do syntax" at the higher level of the connectionist cognitive architecture or to "do semantics" at the lower level. But this does not change the fact that to "do syntax" (describe processing), we must choose between (a) explicitly working on the lower level where the description is tractable, or (b) pretending to work at the higher level with descriptions that are intractable because they involve all the mess of secretly transforming to the lower level, doing the real work there, and secretly transforming back. A cognitive scientist working with a Classical Architecture is simply not forced to make this choice.

(8) a. Thoughts have composite structure.

By this they mean things like: the thought that *John loves the girl* is not atomic; it's a composite mental state built out of thoughts about *John*, *loves*, and *the girl*.

(8) b. Mental processes are sensitive to this composite structure.

For example, from any thought of the form p & q—regardless of what p and q are—we can deduce p.

F&P elevate (8) to the status of defining the Classical View of Cognition, and claim that this is what is being challenged by connectionism. I am arguing that this is wrong, but for now we continue with F&P's argument.

Having identified claims (8) as definitive of the Classical View, F&P go on to argue that there are compelling arguments for these claims.[4] According to these arguments, mental states have the properties of productivity, systematicity, compositionality, and inferential coherence. Without going into all these arguments, let me simply state that for present purposes I'm willing to accept that they are convincing enough to justify the conclusion that (8) must be taken quite seriously.

Now for F&P's analysis of connectionism. They assert that in (standard) connectionism, all representations are atomic; mental states have no composite structure, violating (8a). Furthermore, they assert, (standard) connectionist processing is association which is sensitive only to *statistics*, not to *structure*—in violation of (8b). Therefore, they conclude, (standard) connectionism is maximally non-Classical: it violates both the defining principles. Therefore connectionism is defeated by the compelling arguments in favor of the Classical View.

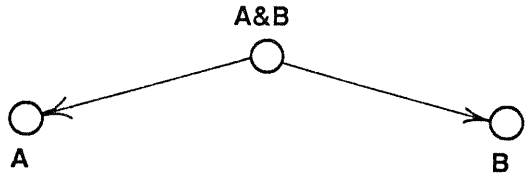
What makes F&P say that connectionist representations are atomic? The second figure of their paper (p. 16) says it all—it is rendered here as Figure 1. This network is supposed to illustrate the standard connectionist account of the inference from A & B to A and to B. It is true that Ballard and Hayes wrote a paper (Ballard & Hayes, 1984; also Ballard, 1986) about using connectionist networks to do automated resolution theorem proving in which networks like this appear. However it is a serious mistake to view this as the paradigmatic connectionist account for anything like human inferences of this sort. This kind of *ultralocal* connectionist representation, in which entire propositions are represented by individual nodes, is far from typical of connectionist models, and certainly not to be taken as *definitive* of the connectionist approach.[5]

---

4. They admit up front that these arguments are a rerun updated for the 80's, a colorized version of a film that was shown in black and white some time ago—where the color comes mainly from replacing everywhere the word "behaviorism" by "connectionism."

5. The conception of connectionist representation and processing embodied in Figure 1 is at the center of this entire argument, so it important to properly locate this network and the Ballard & Hayes paper in the connectionist landscape; for those not well familiar with the territory, this may be facilitated by a sociogeographical digression. Hayes is a leading figure in the logic-based approach to symbolic AI, and (to my knowledge) this collaborative exercise is his only foray onto connectionist turf. Ballard is a leading connectionist of the "Rochester school," which tends to favor local representations over distributed ones, and which as a result represents a radically different set of foundational commitments (see Feldman & Ballard, 1982) from those of the "San Diego" or "PDP" school, as articulated for example in the PDP books (Rumelhart, McClelland, and the PDP Research Group, 1986; McClelland, Rumelhart, and the PDP Research Group, 1986); my version of the PDP framework is articulated as PTC in Smolensky (1988a,b), which explicitly addresses the contrast with Feldman & Ballard (1982). (Incidentally, the name "PDP" was coined to differentiate the approach from the "connectionist" approach already defined by Feldman and Ballard, 1982; the referent of "connectionist" subsequently expanded to engulf the PDP approach [e.g., *Cognitive Science*, 1985]. This left what I have referred to as the "Rochester" approach without a distinctive name; the term "structured connectionist networks" is now sometimes used, but it is potentially quite misleading.) As already evidenced in Section 1.4.2, it turns out that on foundational issues generally, the local vs. distributed issue forces the two schools of connectionism to take quite different positions; a response to F&P from the Feldman & Ballard (1982) perspective would have to differ completely from the one I offer here. While F&P argue that distributed representations make no difference, I now proceed to identify a crucial fallacy in that argument, which this paper as a whole shows to be quite inadequate.

Figure 1: Fodor & Pylyshyn's network



A central claim in my response to F&P is that any critique of the connectionist approach must consider the consequences of using distributed representations, in which the representation of high level conceptual entities such as propositions are distributed over many nodes, and the same nodes simultaneously participate in the representation of many entities. Their response, in Section 2.1.3, (p. 19) is as follows. The distributed/local representation issue concerns (they assume) whether each of the nodes in Figure 1 refers to something complicated and lower level (the distributed case) or not (the local case). But, they claim, this issue is irrelevant, because it pertains to a *between-level* issue, and the compositionality of mental states is a *within-level* issue.

My response is that they are correct that compositionality is a within-level issue, and correct that the distributed/local distinction is a between-level issue. Their argument presumes that because of this difference, one issue cannot influence the other. But that is a fallacy. It assumes that the between-level relation in distributed representations can not have any consequences on the *within-level* structure of the relationships between the representations of *A & B* and the representation of *A*. And that's simply false. There are profound implications of distributed representations for compositionality; these are the subject of all of Section 2 of this paper. In particular, it will turn out that Figure 1 is exactly as relevant to a distributed connectionist account of inference as it is to a symbolic account. In the ultralocal case, Figure 1 is relevant and their critique stands; in the distributed case, Figure 1 is a bogus characterization of the connectionist account and their critique completely misses its target. It will further turn out that a valid analysis of the actual distributed case, based on suggestions of Pylyshyn himself, leads to quite the opposite conclusion: connectionist models using distributed representations describe mental states with a relevant kind of (within-level) constituent structure. The rather weak sense of constituent structure in generic distributed representations, identified in Section 2.2, will be made much stronger in explicitly designed distributed representations, discussed in Section 2.3, in which constituents can fill varying structural roles.

2.2. The distributed (weakly compositional) case

For now, the goal is to show that generic connectionist models using distributed representations ascribe to mental states the kind of compositional structure demanded by (8a), contrary to F&P's conclusion based on the ultralocal network of Figure 1.

2.2.1. The *coffee* story

My argument consists primarily in carrying out an analysis that was suggested by Zenon Pylyshyn himself at the 1984 Cognitive Science Meeting in Boulder. A sort of debate about connectionism was held between Geoffrey Hinton and David Rumelhart on the one hand, and Zenon Pylyshyn and Kurt VanLehn on the other. While pursuing the nature of connectionist representations, Pylyshyn asked Rumelhart: "Look, can you guys represent a cup of coffee in these networks?" Rumelhart's reply was "Sure" so Pylyshyn continued: "And can you represent a cup without coffee in it?" Waiting for the trap to close, Rumelhart said "Yes," at which point Pylyshyn pounced: "Ah-hah, well, the difference between the two is just the representation of *coffee*—you've just built a representation of *cup with coffee* by combining a representation of *cup* with a representation of *coffee*."

I propose to carry out exactly the construction suggested by Pylyshyn, and see what conclusions it leads us to. We'll take a *distributed* representation of *cup with coffee* and subtract from it a distributed representation of *cup without coffee* and we'll call what's left "the connectionist representation of *coffee*."

To generate these distributed representations I will use a set of "microfeatures" (Hinton, McClelland, & Rumelhart, 1986) that are not very micro—but that's always what happens in examples that are cooked up to be intuitively understandable in a nontechnical exposition. These microfeatures are shown in Figure 2.

Figure 2: Representation of *cup with coffee*

Units	Microfeatures
●	upright container
●	hot liquid
○	glass contacting wood
●	porcelain curved surface
●	burnt odor
●	brown liquid contacting porcelain
●	porcelain curved surface
○	oblong silver object
●	finger-sized handle
●	brown liquid with curved sides and bottom

Figure 2 shows a distributed representation of *cup with coffee*: a pattern of activity in which those units that are active (black) are those that correspond to microfeatures present in the description of a cup containing coffee. Obviously, this is a crude, nearly sensory-level representation, but, again, that helps make the example more intuitive—it's not essential.

Figure 3: Representation of *cup without coffee*

Units	Microfeatures
●	upright container
○	hot liquid
○	glass contacting wood
●	porcelain curved surface
○	burnt odor
○	brown liquid contacting porcelain
●	porcelain curved surface
○	oblong silver object
●	finger-sized handle
○	brown liquid with curved sides and bottom

Given the representation of *cup with coffee* displayed in Figure 2, Pylyshyn suggests we subtract the representation of *cup without coffee*. The representation of *cup without coffee* is shown in Figure 3, and Figure 4 shows the result of subtracting it from the representation of *cup with coffee*.

Figure 4: "Representation of *coffee*"

Units	Microfeatures
○	upright container
●	hot liquid
○	glass contacting wood
○	porcelain curved surface
●	burnt odor
●	brown liquid contacting porcelain
○	porcelain curved surface
○	oblong silver object
○	finger-sized handle
●	brown liquid with curved sides and bottom

So what does this procedure produce as "the connectionist representation of *coffee*"? Reading off from Figure 4, we have a burnt odor and hot brown liquid with curved sides and bottom surfaces contacting porcelain. This is indeed a representation of *coffee*, but in a very particular context: the context provided by *cup*.

What does this mean for Pylyshyn's conclusion that "the connectionist representation of *cup with coffee* is just the representation of *cup without coffee* combined with the representation of *coffee*"? What is involved in combining the representations of Figures 3 and 4 back together to form that of Figure 2? We assemble the representation of *cup with coffee* from a representation of a *cup*, and a representation of *coffee*, but it's a rather strange combination. There's also the representation of the *interaction* of the cup with coffee—like *brown liquid contacting porcelain*. Thus the composite representation is built from coffee *extracted* from the situation *cup with coffee*, together with *cup* extracted from the situation *cup with coffee*, together with their interaction.

So the compositional structure is there, but it's there in an *approximate* sense. It's *not* equivalent to taking a context-independent representation of *coffee* and a context-independent representation of *cup*—and certainly not equivalent to taking a context-independent representation of the relationship *in* or *with*—and sticking them all together in a symbolic structure, concatenating them together to form the kind of syntactic compositional structures like `with(cup,coffee)` that F&P want connectionist nets to implement.

To draw this point out further, let's reconsider the representation of *coffee* once the cup has been subtracted off. This, suggests Pylyshyn, is the connectionist representation of *coffee*. But as we have already observed, this is really a representation of *coffee* in the particular context of being inside a cup. According to Pylyshyn's formula, to get the connectionist representation of *coffee* it should have been in principle possible to take the connectionist representation of *can with coffee* and subtract from it the connectionist representation of *can without coffee*. What would happen if we actually did this? We would get a representation of ground brown burnt smelling granules stacked in a cylindrical shape, together with granules contacting tin. This is the connectionist representation of *coffee* we get by starting with *can with coffee* instead of *cup with coffee*. Or we could start with the representation of *tree with coffee* and subtract off *tree without coffee*. We would get a connectionist representation for *coffee* which would be a representation of brown beans in a funny shape hanging suspended in mid-air. Or again we could start

with *man with coffee* and get still another connectionist representation of *coffee*: one quite similar to the entire representation of *cup with coffee* from which we extracted our first representation of *coffee*.

The point is that the representation of *coffee* that we get out of the construction starting with *cup with coffee* leads to a different representation of *coffee* than we get out of other constructions that have equivalent a priori status. That means that if you want to talk about the connectionist representation of *coffee* in this distributed scheme, you have to talk about a *family of distributed activity patterns*. What knits together all these particular representations of *coffee* is nothing other than a type of family resemblance.

2.2.2. Morals of the *coffee* story

The first moral I want to draw out of this *coffee* story is this: unlike the ultralocal case of Figure 1, with distributed representations, complex representations *are* composed of representations of constituents. The constituency relation here is a *within-level* relation, as F&P require: the pattern or *vector* representing *cup with coffee* is composed of a *vector* that can be identified as a distributed representation of *cup without coffee* together with a *vector* that can be identified as a particular distributed representation of *coffee*. In characterizing the constituent vectors of the vector representing the composite, we are *not* concerned with the fact that the vector representing *cup with coffee* is a vector comprised of the activity of individual microfeature units. The *between-level* relation between the vector and its individual numerical elements is *not* the constituency relation, and so section 2.1.4 (p. 19–28) of F&P is irrelevant—it addresses a mistake that is not being made.

The second moral is that the constituency relation among distributed representations is one that is important for the analysis of connectionist models, and for explaining their behavior, but it is *not* a part of the information processing mechanism within the connectionist model. In order to process the vector representing *cup with coffee*, the network does not have to decompose it into constituents. For processing, it is the *between-level* relation, not the within-level relation, that matters. The processing of the vector representing *cup with coffee* is determined by the individual numerical activities that make up the vector: it is over these lower-level activities that the processes are defined. Thus the fact that there is considerable arbitrariness in the way the constituents of *cup with coffee* are defined introduces no ambiguities in the way the network processes that representation—the ambiguities exist only for us who analyze the model and try to explain its behavior. Any particular definition of constituency that gives us explanatory leverage is a valid definition of constituency; lack of uniqueness is not a problem.

This leads directly to the third moral: the decomposition of composite states into their constituents is not precise and uniquely defined. The notion of constituency is important but attempts to formalize it are likely to crucially involve *approximation*. As discussed at some length in Smolensky (1988a), this is the typical case: notions from symbolic computation provide important tools for constructing higher-level accounts of the behavior of connectionist models using distributed representation—but these notions provide approximate, not precise, accounts.

Which leads to the fourth moral: while connectionist networks using distributed representations *do* describe mental states with the type of constituency required by (8a), they do *not* provide an implementation—correctly defined—of a symbolic language of thought. The context-dependency of the constituents, the interactions that must be accommodated when they are combined, the inability to uniquely, precisely identify constituents, the imperative to take seriously the notion that the representation of *coffee* is a collection of vectors knit together by family resemblance—all these entail that the relation between connectionist constituency and syntactic symbolic constituency is *not* one of implementation. In particular, it would be absurd to claim that even if the connectionist story is correct then that would have no implications for the cognitive architecture, that it would merely fill in lower-level details without important implications for the higher-level account.

These conclusions all address compositional representation (8a) without explicitly addressing structure-sensitive processing (8b). Addressing structure-sensitivity to the depth necessary to grapple with real cognitive modeling is far beyond the scope of this paper; to a considerable extent, it is beyond the scope of current connectionism. However, let me simply state the fundamental hypothesis of PTC that weaves the statistical sensitivity characteristic of connectionist processing together with the notion of structure sensitivity: *the mind is a statistics-sensitive engine operating on structure-sensitive (numerical) representations.* The previous arguments have shown that distributed

representations do possess constituency relations, and that, properly analyzed, these representations can be seen to encode structure. Extending this to grapple with the complexity of the kinds of rich structures implicated in complex cognitive processes is the topic of the next section. Here it suffices to observe that once we have complex structured information represented in distributed numerical patterns, statistics-sensitive processes can proceed to analyze the statistical regularities in a fully structure-sensitive way. Whether such processes can provide structure-sensitivity that is adequate to cope with the demands of linguistic and inferential processing is sure to be unknown for some time yet.

The conclusion, then, is that distributed models *can* satisfy (8). Whether (8) can be satisfied to the depth required by the full demands of cognitive modeling is of course an open empirical question—just as it is for the symbolic approach to satisfying (8). At the same time, distributed connectionist models do *not* amount to an implementation of the symbolic instantiations of (8) that F&P are committed to.

Before summing up, I'd like to return to Figure 1. In what sense can Figure 1 be said to describe the relation between the distributed representation of *A&B* and the distributed representations of *A* and *B*? It was the intent of the *coffee* story to show that the distributed representations of the constituents are, in an approximate but explanation-relevant sense, part of the representation of the composite. Thus, in the distributed case, the relation between the node of Figure 1 labeled *A&B* and the others is one kind of whole/part relation. An inference mechanism that takes as input the vector representing *A&B* and produces as output the vector representing *A* is a mechanism that extracts a part from a whole. And in this sense it is no different from a symbolic inference mechanism that takes the syntactic structure `A & B` and extracts from it the syntactic constituent `A`. The connectionist mechanisms for doing this are of course quite different than the symbolic mechanisms, and the approximate nature of the whole/part relation gives the connectionist computation different overall characteristics: we don't have simply a new implementation of the old computation.

It is clear that, just as Figure 1 offers a crude summary of the symbolic process of passing from `A & B` to `A`, a summary that uses the labels to encode hidden internal structures within the nodes, *exactly the same is true of the distributed connectionist case.* In the distributed connectionist case—*just as in the symbolic case*—the links in Figure 1 are crude summaries of complex processes and not simple-minded causal channels that pass activity from the top node to the lower nodes. Such a simple causal story applies only to the ultralocal connectionist case, which is the only legitimate target of F&P's attack.

Let me be clear: there is no serious distributed connectionist model, as far as I know, of the kind of formal inference F&P have in mind here. Many proponents of connectionism would be content to claim that formal inference is a specially trained, poorly practiced skill that is far from central to cognition, and that therefore we can afford to put off worrying about providing a connectionist model of it for a long time. I prefer to say that, at root, the F&P argument concerns an important and central issue: the constituent structure of mental states; formal inference is just one setting in which to see the importance of that constituent structure. So the preceding discussion of the constituent structure of distributed representations does address the heart of their critique, even if a well-developed connectionist account of formal inference remains unavailable.

2.3. The distributed (strongly compositional) case

But, one might well argue, the sense in which the vector encoding the distributed representation of *cup with coffee* has constituent vectors representing *cup* and *coffee* is too weak to serve all the uses of constituent structure—in particular, too weak to support formal inference—because the vector representing *cup* cannot fill multiple structural roles. A true constituent can move around and fill any of a number of different roles in different structures. Can *this* be done with vectors encoding distributed representations, and be done in a way that doesn't amount to simply implementing symbolic syntactic constituency? The purpose of this section is to describe research showing that the answer is affirmative.

A large class of connectionist representations, which I call *tensor product representations*, is defined and analyzed in Smolensky (1987a), and applied in Dolan & Smolensky (1988). We generate various members of this class by variously specifying several parameters in a highly general method for creating connectionist representations of structured information. The resulting parametric variation in the representations is very broad,

encompassing very simple representations such as the case of Figure 1, as well as representations that are close to true implementations of a syntactic language of thought. This class of representations covers the spectrum from fully distributed representations to ultralocal ones, and includes representations with a full sense of constituency, where role-independent constituents are assigned to roles in a structure and the representation of the structure is built up systematically from the representation of the constituents.

The problem that motivates this work is mapping complex structures such as parse trees into vectors of activity in connectionist networks, in such a way that the constituent structure is available for connectionist processing. A general formal framework for stating this problem is to assume that there is a set of discrete structures S (like parse trees) and a vector space V—a space of activity states of a connectionist network. A connectionist representation is a mapping from S to V; the theorist's job is to identify such mappings having various desirable properties. Tensor product representations can provide many of these properties.

A particular tensor product representation is constructed in two steps.

(9) a. Specify a decompositional process whereby the discrete structures are explicitly broken down as a set of constituents, each filling a particular role in the structure as a whole. This step has nothing to do with connectionism per se, it just amounts to being specific about the kind of constituent structure we want to represent.

b. Specify two connectionist representations: one for the structural roles and another for their fillers (the constituents). Thus, for every filler, we assign a vector in the state space of some network for representing fillers; similarly, we assign to every role a vector in the state space of some network for representing roles.

These two steps indicate the "parameters" in the general tensor product representational scheme that must be specified to individuate a particular representation. Once these are parameters are specified, two very simple operations from the theory of vector spaces are used to generate the representation of a particular discrete structure. The representation of the whole is built from the representation of its constituent parts by the operation of *superposition* which is simply *vector addition*: the vector representing the whole is the sum of the vectors representing the parts. Step (9a) above specifies exactly what constituents are involved in this process. The vector representing a given constituent is actually a role-sensitive representation: a representation of that constituent *in the role it plays in the whole*. This vector is built by taking a particular vector product of the vector that represents the constituent independent of any role, and the vector representing the role in the structure that is filled by the constituent. Step (9b) specifies a set of vectors that represent individual structural roles and another set of vectors that represent individual fillers for those roles (constituents) independently of any role. The product operation here is a vector operation called the *tensor product* that takes two vectors and produces a new vector; if the two vectors consist of n and m activity values, then their tensor product is a vector of nm activity values, each one being a different product (using ordinary numerical multiplication) of two activity values, one from each of the original vectors.[6]

The tensor product provides a general solution to a problem that has been nagging the distributed connectionist representational world for a long time, the so-called *variable binding problem*: How can we take an activity pattern representing a variable and another pattern representing a value and generate a connectionist representation of their binding that has the right computational properties? The simplicity of the tensor product makes it possible to show it does in fact satisfy the computational demands of (distributed) connectionist variable binding. The tensor product

6. The tensor product is closely related to what is called the *outer product* in matrix algebra (not to be confused with the physicists' *cross product* of three-dimensional vectors, which is sometimes also called the "outer product"). If $\mathbf{u}$ and $\mathbf{v}$ are two column-vectors, $[N\times1]$ and $[M\times1]$ respectively, their outer product is the $[N\times M]$ matrix $\mathbf{u}\mathbf{v}^T$, where $\mathbf{v}^T$ is the $[1\times M]$ row-vector transpose of $\mathbf{v}$. If the NM numbers in this $[N\times M]$ matrix are considered as the elements of vector (rather than a matrix) then this vector is the tensor product of $\mathbf{u}$ and $\mathbf{v}$. In order for the connectionist representational scheme we are considering to be able to handle recursive decompositions of structures, the product operation we use must be extensible to products of arbitrarily many vectors; with the tensor product, this is nonproblematic, whereas the utility of matrix algebra is essentially limited to outer products of two vectors.

technique is a generalization of specific tricks (especially, *conjunctive coding*: Hinton, McClelland, & Rumelhart, 1986; McClelland & Kawamoto, 1986; Smolensky, forthcoming) that have been used to solve this problem in particular instances in the past.

The tensor product representation of constituent structure considerably strengthens the notion of constituency brought out in the previous section through the *coffee* story. There we saw that the whole/part relation between *cup with coffee* and *coffee* is mirrored in a whole/part relation between their respective representations: the latter relation was not the whole/part relation between molecular symbolic structures and their atomic constituents, as in a symbolic language of thought, but rather the relation between a sum vector $\mathbf{w}$ and the component vectors that add up to it: $\mathbf{w} = \mathbf{c}_1+\mathbf{c}_2+\cdots$. The same is true here generally with respect to tensor product representations, but now in addition we can identify the representations of each constituent as a role-*dependent* representation built in a systematic way (through tensor product variable binding) from a role-*independent* representation of the filler and a filler-independent representation of its role.

Among the computational properties required of the variable binding mechanism is the possibility of *unbinding*: from the role-dependent representation of some constituent we must be able to extract the role-independent representation of that constituent. Similarly, given the vector representing a symbolic structure as a whole, it should be possible to extract the role-independent representation of the filler of any given role in the structure. Under a wide variety of conditions, this is possible with the tensor product representation, although when so many roles are simultaneously filled that the capacity of the representing network is exceeded, corruptions, confusions, and errors can be introduced during unbinding. The conditions under which error-free unbinding can be performed, and characterization of the errors occurring when these conditions are violated, can be computed (Smolensky, 1987a). Thus, for example, if we have a tensor product representation for $P\&Q$, and we wish to extract the first element P as part of a deductive process, then as long as the representing network is not trivially small, we can easily do so without error, using very simple (linear) connectionist processes.

So, returning to F&P's critique, let's see what the tensor product representational scheme can do for us in terms of the simple Aristotelean inference problems they talk about.

Using the tensor product technique, it is possible to define a family of representations of tree structures. We can consider a simple tree for $P\&Q$ consisting of $\&$ at the top, P as its left child, and Q as its right child; and we can view the roles as positions in the tree, the simplest kind of role decomposition. The tensor product representation of that tree structure is a vector $\mathbf{F}(P\&Q)$ which is related to the vectors representing the constituents, $\mathbf{F}(P)$ and $\mathbf{F}(Q)$, by a function $B_{\&}$ that is particular to constructing conjunctions:

$$\mathbf{F}(P\&Q) = B_{\&}[\mathbf{F}(P), \mathbf{F}(Q)]$$

The function $B_{\&}$ is defined by

$$B_{\&}(\mathbf{u}, \mathbf{v}) = \mathbf{c}_{\&} + \tau_0\mathbf{u} + \tau_1\mathbf{v}$$

where $\mathbf{c}_{\&}$ is a constant vector and τ_0 and τ_1 are linear operators (the most natural vector operators) that vary depending on how the parameters individuating the tensor product representation are chosen.

I have descended to this level of detail and used this notation because in footnote 9 (p. 14) of F&P, exactly this property is chosen to define $\mathbf{F}$ as a "physical instantiation mapping of combinatorial structure." In this sense the tensor product representation meets F&P's formal requirements for a representation of combinatorial structure.

But have we merely provided an implementation then of a symbolic language of thought? In general, the answer is "no." Depending on how we have chosen to set the parameters in specifying the tensor product representation (which determines the properties of τ_0 and τ_1), we can fail to have any of the following properties holding (Smolensky, 1987a):

(10) a. *Uniqueness with respect to roles or fillers.* If we're not careful, even though the above equation is satisfied, we can end up with *P&Q* having the same representation as *Q&P*, or other more subtle ambiguities about what fills various roles in the structure.

b. *Unbounded depth.* We may avoid the first problem (10a) for sufficiently small structures, but when representing sufficiently large or deep structures, these problems may appear. Unless the vector space in which we do our representation is infinite-dimensional (corresponding to a network with infinitely many units), we cannot solve (10a) for unbounded depth. (Of course, the same is true of Turing/von Neumann machines if they are only allowed bounded resources; but whereas the capacity limit in the symbolic case is a hard one, the tensor product representation allows for graceful degradation as resources are saturated.)

c. *Nonconfusability in memory.* Even when problem (10a) is avoided, when we have representations with uniquely determined filler/role bindings, it can easily happen that we cannot simultaneously store many such structures in a connectionist memory without getting intrusions of undesired memories during the retrieval of a given memory.

d. *Processing independence.* This is in a sense a generalization of the preceding point, concerning processing constraints that may arise even when problem (10a) is avoided. In simple associative processing, for example, we may find that we can associate two vectors representing symbolic structures with what we like, but then find ourselves unable to associate the representation of a third structure with what we like, because its associate is constrained by the other two.

With all these properties potentially failing to hold, it doesn't sound to me like we're dealing with an implementation of a symbolic language of thought. But at this point somebody's going to want to say, "Well, you've just got a *lousy* implementation of a symbolic language of thought." But it's not that simple. We may have lost some (superficially desirable, at least) features of a symbolic language of thought, but we've gained some (superficially desirable, at least) features of connectionist processing in return.

(849) a. *Massive parallelism.* Since we have a vector that represents an entire tree at once, we can feed it into the usual connectionist massively parallel processes. We don't have to chunk around taking the `car` of the `cdr` of the `cdr` of the `car` to get to one of the multitude of pieces of information we need for a given process: It's all there at once, all accessible in parallel.[7]

b. *Content-addressable memory.* This is the usual distributed connectionist story, but now it applies to *structured information.*

c. *Statistical inference.* F&P are among the first to attack connectionism for basing its processing mechanisms on statistical inference. One more reason for them to deny that the connectionist framework I am discussing truly constitutes an implementation of their preferred architecture. Yet their arguments *against* statistical processing are much less compelling than their arguments *for* structure-sensitive processing. We are now in a position to go after *both*, in a unified framework, dissolving a long-standing tension arising from a failure to see how to formally unify structure-sensitive and statistical processing. Rather than having to model the mind as *either* a structure cruncher *or* a number cruncher, we can now see it as a number cruncher in which the numbers

---

7. It's all well and good to say, as F&P do, that the Classical view has no commitment to serial processing. "We like parallel computation too." Fine, give me a massively parallel symbolic model that processes tree structures and I'll be happy to compare it to this. But I don't see it out there.

See Dolan & Smolensky (1988) for an actual distributed connectionist model, TPPS, that uses the tensor product to represent a symbolic structure and operate on it with massive parallelism. The system is an exercise in applying the tensor product representation to put on a somewhat more general and simple mathematical footing Touretzky & Hinton's (1985) Boltzmann machine implementation of a distributed connectionist production system, DCPS. Each production in TPPS does pattern matching against the whole symbolic structure in working memory in parallel, and does all parts of its action in parallel. Since it is an implementation of a traditional production system, however, productions are fired one at a time, although conflict resolution is done in parallel. In progress is the application of the tensor product representation to a fully parallel and distributed parser for context-free grammars.

crunched are in fact representing complex structures.[8]

d. *Statistical learning.* Since structure can now be brought fully into the world of connectionist learning research, we can move from declarations of dogma to actual empirical results about what structurally-rich representations and processes can and cannot be acquired from experience through statistically based learning. We can now foresee a time when it will be too late to put your money down on the fate of the "poverty of the stimulus" dogma.

The bottom line is that the parametric variation in tensor product representations extends from simple ultralocal representations of the sort F&P correctly dismiss towards—I hesitate to say all the way up to, but quite close to—a true implementation of a symbolic language of thought. If you want such an implementation, you have to go to a limit that includes the following characteristics:

(12) a. *Orthogonality.* The angle between the vectors representing different roles needs to go to 90 degrees, and similarly for vectors representing the fillers, to eliminate non-uniqueness and minimize interference in memory.

b. *Infinite-dimensional representations.* Otherwise, we can't represent unboundedly deep structures without confusion.

c. *Simple operations.* If we happen to want an implementation of sequential algorithms, then in processing these representations we insist that the vector equivalent of the primitive symbolic operations like `car`, `cdr`, and `cons` are all that can be done in one time step: We don't avail ourselves of the massively parallel operations that otherwise would be available to us.

I have talked so far mostly about representations and little about processing. If we are interested, as F&P are, in inferences such as that from *P&Q* to *P*, it turns out that with tensor product representations, this operation can be achieved by a simple linear transformation upon these representational vectors, the kind of transformation most natural in this category of representations.[9] Not only can this structure-sensitive process be achieved by connectionist mechanisms on connectionist representations, but it can be achieved through the simplest of all connectionist operations: linear mapping. All in an architecture that differs fundamentally from the Classical one; we have not implemented a symbolic language of thought.

3. Connectionism, implementationalism, and limitivism

In this final section I'd like to bring together the arguments of the first two sections, showing how the debate over constituent structure relates to larger issues such as the import of rejecting implementationalism and of viewing the commitment of connectionism as being the development of accounts of computation and cognition that exploit insights from vector space theory and other branches of continuous mathematics.

3.1. The methodological implications of implementationalism and limitivism

Summarizing the constituency argument, we've got F&P principles of structure (8), and we've got a symbolic instantiation of these in a language of thought using syntactic constituency. According to F&P, what connectionists should do is take that symbolic language of thought as a higher level description and then produce a connectionist implementation. The syntactic operations of the symbolic language of thought then provide an exact formal higher level account of mental representations and processes.

---

8. Like connectionist networks, traditional computers were originally viewed exclusively as number processors. Newell and Simon are credited with teaching us that traditional computers could also be used as powerful structure processors. I am essentially trying to make the same point about connectionist networks.

9. This is true provided the parameter values defining the representation satisfy the very weak constraint that the simplest possible confusions are avoided (such as confusing *P&Q* with *Q&P* or with *P or Q*).

By contrast, I have argued that the distributed view of connectionist compositionality allows us to instantiate the same basic principles (8) *without* going through a symbolic language of thought. By going straight to distributed connectionist models we get new formal instantiations of compositionality principles.

I happen to believe that the symbolic descriptions *will* provide scientifically important *approximate* higher level accounts of how the ultimate connectionist cognitive models compute—but that these distributed connectionist models will not implement a symbolic language of thought, under the relevant (and correct) definition of the word. The approximations involved demand a willingness to accept context-sensitive symbols and interactional components present in compositional structures, and the other funny business that came out in the *coffee* example. If we're willing to live with all those degrees of approximation, then we can usefully view these symbolic level descriptions as approximate higher level accounts of the processing in a connectionist network.

An important overall conclusion on the constituency issue, then, is that *the Classical and connectionist approaches differ not in whether they accept principles (8), but in* ***how they formally instantiate them.*** To really confront the Classical/connectionist dispute, one has to be willing to descend to the level of the particular formal instantiations they give to the nonformal principles (8). To fail to descend to this level of detail is to miss much of the issue. In the Classical approach, principles (8) are formalized using syntactic structures for mental representations and symbol manipulation for mental processes. In the distributed connectionist approach (8) are formalized using vectorial representations for mental representations, and the corresponding notion of compositionality, together with numerical mental processes that derive their structure sensitivity from the differential way that they treat the parts of vectors corresponding to different structural roles.

In terms of research methodology, this means that the agenda for connectionism should be not be to develop a connectionist implementation of the symbolic language of thought but rather to develop formal analysis of vectorial representations of complex structures and operations on those structures that are sufficiently structure-sensitive to do the required work. This is exactly the kind of research that, for example, tensor product representations are being used to support.

Thus the PTC position is that distributed representations provide a description of mental states with semantically interpretable constituents, but that there is no complete, precise formal account of the construction of composites or of mental processes in general that can be stated solely in terms of context-independent semantically interpretable constituents. On this account, there *is* a language of thought—but only approximately; the language of thought by itself does not provide a basis for an exact formal account of mental structure or processes—it cannot by itself support a precise formal account of the cognitive architecture.[10]

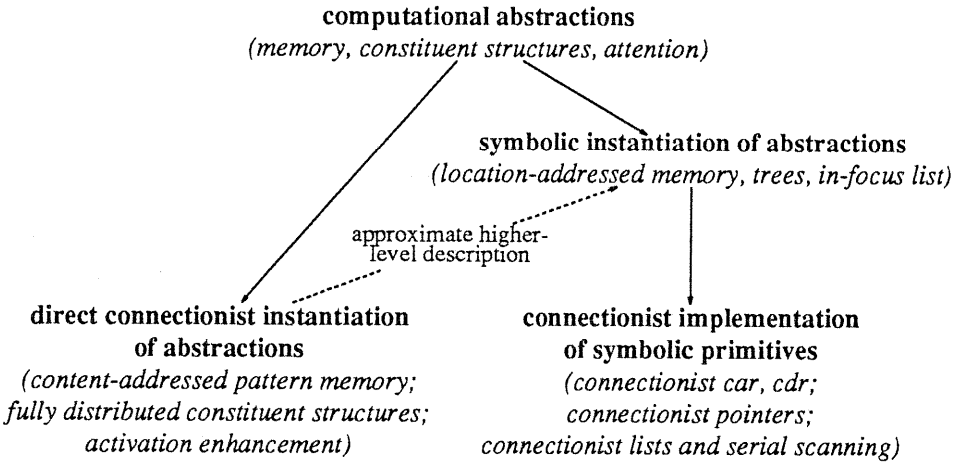
Constituency is one illustration of a central component of the general PTC approach to connectionism: the relation hypothesized between connectionist models based on continuous mathematics and Classical models based on discrete, symbolic computation. That relationship might be called the *cognitive correspondence principle*: When powerful connectionist computational systems are appropriately analyzed at higher levels, elements of symbolic computation appear as emergent properties.

Figure 5 schematically illustrates the cognitive correspondence principle. At the top are nonformal notions: the central hypotheses that the principles of cognition consist in principles of memory, of inference, of compositionality and constituent structure, etc. In the F&P argument, the relevant nonformal principles were their compositionality

10. An important open question is whether the kind of story I have given on *cup of coffee* using those hokey microfeatures will carry over to the kind of distributed representations that real connectionist networks create for themselves in their hidden units—if the analysis is made appropriately more sophisticated. The resolution of this issue depends on the (as yet largely inscrutable) nature of these representations for realistic tasks. The nature of the network's task is important, for it is perfectly likely that connectionist networks will develop compositional representations in their hidden units only when this is advantageous for the problem they are trying to solve. As F&P, and the entire Classical paradigm, argue, such compositional representations are in fact immensely useful for a broad spectrum of cognitive tasks. But until such tasks—which tend to be considerably more sophisticated than those usually given to connectionist networks—have been explored in some detail with connectionist models, we won't really know if hidden units will develop compositional representations (in the approximate sense discussed in this paper) when they "should."

principles (8).

Figure 5: PTC vs. implementationalism
(Reprinted with permission of *The Behavioral and Brain Sciences*.)



The nonformal principles at the top of Figure 5 have certain formalizations in the discrete mathematical category, which are shown one level down on the right branch. For example, memory is formalized as standard location-addressed memory or some appropriately more sophisticated related notion. Inference gets formalized in the discrete category as logical inference, a particular form of symbol manipulation. And so on.

The PTC research agenda consists in taking these kinds of cognitive principles and finding new ways to instantiate them in formal principles based on the continuous mathematics of dynamical systems; these are shown in Figure 5 at the lowest level on the left branch. The concept of memory retrieval is reformalized in terms of the continuous evolution of a dynamical system towards a point attractor whose position in the state space is the memory; we naturally get content-addressed memory instead of location-addressed memory. (Memory storage becomes modification of the dynamics of the system so that its attractors are located where the memories are supposed to be; thus the principles of memory storage are even more unlike their symbolic counterparts than those of memory retrieval.) When reformalizing inference principles, the continuous formalism leads naturally to principles of statistical inference rather than logical inference. And so on.

The cognitive correspondence principle states that the general relationship between the connectionist formal principles and the symbolic formal principles—given that they are both instantiations of common nonformal notions, and to the extent that ultimately they are both scientifically valid descriptions of the same cognitive system—is that if we take a higher level analysis of what's going on in the connectionist systems we find that it matches, to some kind of approximation, what's going on in the symbolic formalism. This relation is indicated in Figure 5 by the dotted arrow.

This is to be contrasted with an implementational view of connectionism such as that which F&P advocate. As portrayed in Figure 5, the implementational methodology is to proceed from the top to the bottom not directly, via the left branch, but indirectly, via the right branch: connectionists should take the symbolic instantiations of the nonformal principles and should find ways of implementing *them* in connectionist networks.

The PTC methodology is to be contrasted not just with the implementational approach, but also with the eliminativist one. In terms of these methodological considerations, eliminativism has a strong and a weak form. The weak form advocates taking the left branch of Figure 5 but ignoring altogether the symbolic formalizations, on the belief that the symbolic notions will confuse rather than enlighten us in our attempts to understand connectionist computation. The strong eliminativist position states that even viewing the nonformal principles at the top of Figure

5 as a starting point for thinking about cognition is a mistake—that it is better, for example, to pursue a blind bottom-up strategy in which we take low-level connectionist principles from neuroscience and we see where they lead us, without being prejudiced by archaic prescientific notions such as those at the top of Figure 5.

In rejecting both the implementationalist and eliminativist positions, PTC views connectionist accounts in significant part as reducing and explaining symbolic accounts. Connectionist accounts serve to refine symbolic accounts, to reduce the degree of approximation required, to enrich the computational notions from the symbolic and discrete world, to fill them out with notions of continuous computation. Primarily that's done by descending to a lower level of analysis, by exposing the hidden microstructure in these kinds of large-scale, discrete symbolic operations.

I have dubbed the PTC position *limitivism* because it views connectionism as delimiting the domain D of validity of symbolic accounts, and explaining the validity of the symbolic approximation through passage to the "Classical limit," a general theoretical limit incorporating, e.g., the specifics described in (12), in which connectionist accounts admit, more and more exactly, higher-level symbolic accounts—at least in the limited domain D. This limitivist position on the relation between connectionism and symbolic theory is obviously modeled after a relation frequently observed in the refinement of physical theories, e.g., the relation between quantum and Newtonian mechanics.

The cognitive correspondence principle is so named because I believe that it has a role to play in the developing microtheory of cognition that's analogous to the role that the quantum correspondence principle played in the development of microtheory in physics. This case from physics instantiates the structure of Figure 5 quite directly. There are certain fundamental physical principles that arch over both the classical and quantum formalisms: the notions of space and time and associated invariance principles, the principles of energy and momentum conservation, force laws, and so on. These principles at the top of Figure 5 are instantiated in particular ways in the classical formalism, corresponding to the point one level down on the right branch. To go to a lower level of physical analysis requires the development of a new formalism. In this quantum formalism, the fundamental principles are reinstantiated: they occupy the bottom of the left branch. The classical formalism can be looked at as a higher level description of the same principles operating at the lower quantum level: the dotted line of Figure 5. Of course quantum mechanics does not *implement* classical mechanics: the accounts are intimately related, but classical mechanics provides an approximate, not an exact, higher-level account.[11] In a fundamental sense, the quantum and classical theories are quite incompatible: according to the ontology of quantum mechanics, the ontology of classical mechanics is quite impossible to realize in this world. But there is no denying that the classical ontology and the accompanying principles are theoretically essential, for at least two reasons: (a) to provide explanations (literally, perhaps, approximate ones) of an enormous range of classical phenomena for which direct explanation from quantum principles is hopelessly infeasible, and (b) historically, to provide the guidance necessary to discover the quantum principles in the first place. To try to develop lower-level principles without looking at the higher-level principles for guidance, given the insights we have gained from those principles, would seem—to put it mildly—inadvisable. It is basically this pragmatic consideration that motivates the cognitive correspondence principle and the PTC position it leads to.

3.2. Constituency via vector decomposition, explanatory relevance, and causal efficacy

As a final topic I would like to show how the previous methodological considerations relate specifically to the technical heart of this paper. I want to show that, if we take the general position advocated above that the research agenda of distributed connectionism is to find formal means within the continuous mathematics of dynamical systems for naturally and powerfully embodying central nonformal principles of computation and cognition, then the connectionist analysis of constituent structure I have described here is, if not inevitable, then at least perfectly natural. I take up this topic because it has been suggested that in my analysis, perhaps in order to cook up a

11. Many cases analogous to "implementation" *are* found in physics: Newton's laws provide an "implementation" of Kepler's laws; Maxwell's theory "implements" Coulomb's law; the quantum principles of the hydrogen atom "implement" Balmer's formula.

refutation of F&P, I have seriously contorted the notion of constituency; that superposition of vectors, and tensor product binding, are just not appropriate means of instantiating constituency.

At the same time, I will consider the central question: "Is the sense in which vector decomposition constitutes a constituency relation adequate to make constituency *explanatorily relevant* to or *causally efficacious* in the account of the systematicity of thought, the basic problem motivating F&P's critique?"

Let me begin with a few words about the idea of decomposing a vector into a sum or superposition of component vectors: $\mathbf{w} = \mathbf{c}_1+\mathbf{c}_2+\cdots$. This technique is very commonly used to explain the behavior of dynamical systems; it works best for simple linear systems, where the equations governing the interaction between state variables are linear (such as the very simplest connectionist models). In that case—and the technique gets more complicated from there—the story is as follows.

We want to know, if we start the system off in some initial state described by the vector $\mathbf{w}$, what will the system's subsequent behavior be? (In the connectionist case, $\mathbf{w}$ characterizes the input, and we want to know what states the system will then go through; especially, what the later state that determines the output will be.) First we ask, how can the vector $\mathbf{w}$ be decomposed: $\mathbf{w} = \mathbf{c}_1+\mathbf{c}_2+\cdots$, so that the component vectors $\mathbf{c}_i$ are along certain special directions, determined by the linear interaction equations of the system; these directions $\mathbf{m}_i$ are called the "normal modes" of the system, and each $\mathbf{c}_i = c_i\mathbf{m}_i$, where the coefficient c_i tells how strongly represented in this particular input $\mathbf{w}$ the i^{th} normal mode is. Once we have decomposed the vector into components in the directions of the normal modes, we can write down in a closed form expression the state of the system at any later time: it is just the superposition of the states arising from each of the normal modes independently, and those normal modes are defined exactly so that it is possible to write down how they evolve in time.[12] Thus, knowing the interaction equations of the system, we can compute the normal modes and how they evolve in time, and then we can explain how *any* state evolves in time, simply be decomposing that state into components in the directions of the normal modes. To see an example of this technique applied to actual connectionist networks, see the general analysis of Smolensky (1986) and the specific analysis in Anderson & Mozer (1981) of the categorization performed in J. A. Anderson's "Brain-State-in-a-Box" model. (Both these analyses deal with what I call *quasi-linear* networks, a class covering many actual connectionist systems, in which the heart of the computation is linear, but a certain degree of non-linearity is also important.)

Thus, to explain the behavior of the system, we usually choose to decompose the state vector into components in the directions of the normal modes, which are conveniently related to the particular dynamics of this system. If there is change in how the system interacts with itself (as in connectionist networks that learn), over time we'll change the way we choose to break up the state in order to explain the behavior. There's no unique way to decompose a vector. That is to say, there are lots of ways that this input vector could be viewed as composed of constituents, but normal mode decomposition happens to enable a good explanation for behavior over time. In general, there may well be other compositions that are explanatorily relevant.

So, far from being an unnatural way to break up the part of a connectionist state vector that represents an input, decomposing the vector into components is exactly what we'd expect to need to do to explain the processing of that input.

Now, how reasonable is it to view this decomposition process as a formalization of the notion of decomposing a "structure" into its "constituents"? I take it that it is a reasonable use of the term "constituent" to say that "electrons are constituents of atoms." In modern physics, what is the relation between the representation of the electron and the representation of the atom?

The state of the atom, like the states of all systems in quantum theory, is represented by a vector in an abstract vector space. Each electron has an internal state (its "spin"); it also has a role it plays in the atom as a whole: it occupies some "orbital," essentially a cloud of probability for finding it at particular places in the atom. The internal state of an electron is represented by a "spin vector"; the orbital or role of the electron (part) in the atom (whole) is

---

12. For example, in a dynamical system that oscillates, the evolution of the normal modes in time is given by: $\mathbf{m}_n(t) = e^{i\omega_n t}\,\mathbf{m}_n$. Each particular normal mode $\mathbf{m}_n$ consists of an oscillation with a particular frequency ω_n.

represented by another vector, which describes the probability cloud. The vector representing the electron as situated in the atom is the tensor product of the vector representing the internal state of the electron and the vector representing its orbital. The atom as a whole is represented by a vector that is the sum or superposition of vectors, each of which represents a particular electron situated in its orbital. (There are also contributions of the same sort from nucleons.)

Thus the vector representing the whole is the sum of tensor products of pairs of vectors; in each pair, one vector represents the parts independent of its role in the whole, and the other represent the role in the whole independent of the part that fills the role. This is exactly the way I have used tensor products to construct distributed connectionist representations for wholes from distributed connectionist representations of their parts (and from distributed representations of the roles of parts in the whole)—and this is exactly where the idea came from.

So someone who claims that the tensor product representational scheme distorts the notion of constituency has some explaining to do.

So does someone who claims that the sense in which the whole has parts is not explanatorily relevant. We explain the properties of atoms by invoking properties of their electronic configuration all the time. Quantum theory aside, physical systems whose states are described by vectors have for centuries had their behavior explained by viewing the state vector as a superposition of component vectors, and explaining the evolution of the total state in terms of the evolution of its component vectors—as I have indicated in the preceding discussion of normal modes.

Are the constituents of mental representations as I have characterized them in distributed connectionist systems causally efficacious in mental processing?

The term "causally efficacious" must be used with some caution. The equations that drive the atom do not work by first figuring out what the components particles are, and then working on each of them separately. The equations take the elements comprising the vector for the whole atom and change them in time. We can *analyze* the system by breaking up the vector for the whole into the vectors for the parts, and in general that's a good way to do the analysis; but nature doesn't do that in updating the state of the system from one moment to the next. So, in this case, are the constituents "causally efficacious" or not?

The same question arises in the connectionist case. The fact is, if the connections that mediate processing of the vectors representing composite structures have the effect of sensible processing of the vector in terms of the task demands, it is very likely that in order to *understand and explain* the regularities in the network's behavior we will need to break the vector for the structure into the vectors for the constituents, and relate the processing of the whole to the processing of the parts. That this decomposition, and not arbitrary decompositions into meaningless component vectors, is useful for explaining the processing is a consequence of the connections that embody the process. Those particular components are useful for those particular connections. In general, what makes one decomposition of a state vector useful for predicting behavior and not other is that the useful decomposition bears some special relation to the dynamics in the system. It may well turn out that to explain various aspects of the system's behavior (for example, various cognitive processes acting on a given input), we will want to exploit various decompositions.

As Fodor and Pylyshyn will I believe agree, care in treating "causal efficacy" is also required for the Classical case. When we write a Lisp program, are the symbolic structures we think in terms of "causally efficacious" in the operation of the computer that runs the program? There is a sense in which they are: even though we normally think of the "real" causes as physical and far below the symbolic level, there is nonetheless a complete and precise algorithmic (temporal) story to tell about the states of the machine described at the level of symbols. Traditional computers (the hardware and especially the software) are designed to make that true, and it is the main source of their power.

The hypothesis I have attributed to distributed connectionism is that there is no comparable story at the symbolic level in the human cognitive architecture: no algorithm in terms of semantically interpretable elements that gives a precise formal algorithmic account of the system's behavior over time. That is a difference with the Classical view that I have made much of. It may be that a good way to characterize the difference is in terms of whether the constituents in mental structures are causally efficacious in mental processing.

Such causal efficacy was not my goal in developing the tensor product representation; rather, the goal was and is the design of connectionist systems that display the kinds of complex systematic behavior seen, for example, in language processing—and the mathematical explanation of that systematicity. As the examples from physics show, it is not only wrong to claim that to explaining systematicity by reference to constituent structures requires that those constituents be causally efficacious: it is also wrong (but more honest) to claim (as Fodor often does) that such an explanatory strategy, while not provably unique, constitutes "the only game in town." There is an alternative explanatory strategy that has been practiced very effectively in physics for centuries, and that strategy can be applied in cognitive science as well. There are now at least two games in town, and rather than pretending otherwise, we should get on with the business of playing those games for all we can. Odds are, given how hard cognitive science is, we'll need to be playing other games too before long.

The Classical strategy for explaining the systematicity of thought is to hypothesize that there is a precise formal account of the cognitive architecture in which the constituents of mental representations have causally efficacious roles in the mental processes acting on them. The PTC view denies that such an account of the cognitive architecture exists,[13] and hypothesizes instead that, like the constituents of structures in quantum mechanics, the systematic effects observed in the processing of mental representations arises because the evolution of vectors can be (at least partially and approximately) explained in terms of the evolution of their components, even though the precise dynamical equations apply at the lower level of the individual numbers comprising the vectors and cannot be pulled up to provide a precise temporal account of the processing at the level of entire constituents—i.e., even though the constituents are not causally efficacious.[14]

4. Summary

Shifting attention away from the refutation of F&P's argument, let me try to summarize what I take to be the positive contributions of the argument presented in this paper.

(13) a. As F&P plead, it *is* crucial for connectionism for connectionism to separate itself from simplistic associationist psychology.

b. Connectionism should accept (not deny) the importance of a number of computational principles fundamental to traditional cognitive science, such as those relating to structure that F&P emphasize, which go beyond the computational repertoire of simple traditional connectionist networks.

c. The computational repertoire of connectionism should be extended by finding ways of directly, naturally, and powerfully realizing these computational principles within the continuous mathematics of dynamical systems (not indirectly, as F&P advocate, by implementing the discrete symbolic formalization of these principles in connectionist networks).

d. The resulting connectionist cognitive models aim to refine the account of cognition provided by symbolic models: the symbolic models provide a scientifically important higher-level approximate account of the connectionist model.

e. Just as a set of symbolic structures offers a domain for modeling structured mental representation and processing, so do sets of vectors, once the appropriate notions are recognized in the new mathematical category. Thus distributed (but not localist) connectionist representations provide a computational arena for structure processing.

f. The tensor product representation is a general technique for creating vectorial (distributed) representations of structures; these representations are built up systematically by binding role-independent vectors representing constituents to vectors representing their roles in the structure as a whole, and superimposing the vectors representing these bindings.

---

13. Except for that limited part of the architecture I have called the "conscious rule interpreter"; see Smolensky, 1988a.

14. I use this characterization rather tentatively because I am not yet convinced that it will not be contaminated by problems with the notion of "causally efficacious."

g. Superposition and tensor products provide simple, natural, but powerful means to instantiate within continuous mathematics the basic computational ingredients needed for representing and processing structured mental states.

h. The resulting connectionist model of mental processing is characterized by context-sensitive constituents, approximately (but not exactly) compositional semantics, massively parallel structure-sensitive processing, statistical inference and statistical learning with structured representations.

i. This connectionist cognitive architecture is intrinsically two-level: semantic interpretation is carried out at the level of patterns of activity while the complete, precise, and formal account of mental processing must be carried out at the level of individual activity values and connections.

j. Thus, mental representations are carried by activity vectors while mental processes are carried by activity values: Mental representation and mental processes reside at two different levels of analysis.

k. Thus, not only is the connectionist cognitive architecture fundamentally different from the Classical one, so is the basic strategy for explaining the systematicity of thought. The systematic behavior of the cognitive system is to be explained by appealing to the systematic constituent structure of the representational vectors, and the connectivity patterns that give rise to and manipulate these vectors: but the mechanism responsible for that behavior does not (unlike in the Classical account) operate through laws or rules that are expressible formally at the level of the constituents.

Acknowledgements

I have benefited greatly from personal conversations with Jerry Fodor and Zenon Pylyshyn, conversations extending from February 1986, when Fodor presented an early version of the argument ("Against connectionism") at the Workshop on the Foundations of AI in Las Cruces, New Mexico, through a (surprising enjoyable) public debate held at MIT in March 1988. The concerns that drove my research to tensor product representations were kindled through that interaction. I have learned a tremendous amount from Georges Rey, thanks to his wonderful insight, open-mindedness, and patience; I would also like to thank him for the invitation to contribute to this volume. Thanks too to Terry Horgan for very helpful discussions, as well as to the other participants of the Spindel Conference on Connectionism and the Philosophy of Mind (which resulted in the collection of papers in which Smolensky, 1987b, appears). I thank Horgan, John Tiensen, and the *Southern Journal of Philosophy* for permission to include a portion of Smolensky (1987b) here. Rob Cummins and Georg Schwarz have helped me enormously in sorting out a number of the issues discussed here, and I refer the reader to their papers for a number of important insights that have not been given their due here. Finally I would like to thank Geoff Hinton, Jay McClelland, Dave Rumelhart, David Touretzky, and more recently, Alan Prince, for many helpful discussions on issues relating to connectionism and structure processing.

This work has been supported by NSF grants IRI-8609599 and ECE-8617947 to the author, by a grant to the author from the Sloan Foundation's computational neuroscience program, and by the Optical Connectionist Machine Program of the NSF Engineering Research Center for Optoelectronic Computing Systems at the University of Colorado at Boulder.

References

Anderson, J.A. & Mozer, M.C. (1981). Categorization and selective neurons. In G. E. Hinton and J. A. Anderson, Eds., *Parallel models of associative memory.* Hillsdale, NJ: Erlbaum.

Ballard, D. (1986). Parallel logical inference and energy minimization. Technical Report TR142. Computer Science Department, University of Rochester. March.

Ballard, D. & Hayes, P.J. (1984). Parallel logical inference. *Proceedings of the Sixth Annual Conference of the Cognitive Science Society.* Rochester, N.Y. June.

Cognitive Science. (1985). Special issue on connectionist models and their applications. **9** (1).

Cummins, R. (1988). *Meaning and mental representation.* Cambridge, MA: MIT Press/Bradford Books.

Cummins, R., & Schwarz, G. (1987). Radical Connectionism. *Southern Journal of Philosophy*, **XXVI** (Supplement), 43–61.

Dolan, C. & P. Smolensky. (1988). Implementing a connectionist production system using tensor products. In D. Touretzky, G. E. Hinton, & T. J. Sejnowski (Eds.), *Proceedings of the Connectionist Models Summer School, 1988.* Morgan Kaufmann.

Feldman, J.A. & Ballard, D.H. (1982). Connectionist models and their properties. *Cognitive Science,* **6**, 205–254.

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

Hinton, G.E., McClelland, J.L., & Rumelhart, D.E. (1986). Distributed representations. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group, *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 2: Psychological and biological models.* Cambridge, MA: MIT Press/Bradford Books.

McClelland, J.L. & Kawamoto, A.H. (1986). Mechanisms of sentence processing: Assigning roles to constituents. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group, *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 2: Psychological and biological models.* Cambridge, MA: MIT Press/Bradford Books.

McClelland, J.L., Rumelhart, D.E., & the PDP Research Group. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 2: Psychological and biological models.* Cambridge, MA: MIT Press/Bradford Books.

Rumelhart, D.E., McClelland, J.L., & the PDP Research Group. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations.* Cambridge, MA: MIT Press/Bradford Books.

Schwarz, G. (1987). *Explaining cognition as computation.* Masters Thesis, Department of Philosophy, University of Colorado at Boulder.

Smolensky, P. (1986). Neural and conceptual interpretations of parallel distributed processing models. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group, *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 2: Psychological and biological models.* Cambridge, MA: MIT

Press/Bradford Books.

Smolensky, P. (1987a). On variable binding and the representation of symbolic structures in connectionist systems. Technical Report CU–CS–355–87. Department of Computer Science, University of Colorado at Boulder. (Revised version to appear in *Artificial Intelligence*.)

Smolensky, P. (1987b). The constituent structure of connectionist mental states: A reply to Fodor and Pylyshyn. *Southern Journal of Philosophy*, **XXVI** (Supplement), 137–163.

Smolensky, P. (1988a). On the proper treatment of connectionism. *The Behavioral and Brain Sciences*. **11**, 1–23.

Smolensky, P. (1988b). Putting together connectionism—again. *The Behavioral and Brain Sciences*. **11**, 59–74.

Smolensky, P. (forthcoming). *Lectures on connectionist cognitive modeling*. Hillsdale, NJ: Erlbaum.

Touretzky, D.S. & Hinton, G.E. (1985). Symbols among the neurons: Details of a connectionist inference architecture. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 238-243.

Table of Contents