Systematicity and the Cognition of Structured Domains

The Journal of Philosophy, 98(4), 1-19, 2001.

By Robert Cummins, James Blackmon, David Byrd, Pierre Poirier, Martin Roth, Georg Schwarz, University of California-Davis

I. Background

The current debate over systematicity concerns the formal conditions a scheme of mental representation must satisfy in order to explain the systematicity of thought.[1] The systematicity of thought is assumed to be a pervasive property of minds, and can be characterized (roughly) as follows: anyone who can think T can think systematic variants of T, where the systematic variants of T are found by permuting T's constituents. So, for example, it is an alleged fact that anyone who can think the thought that *John loves Mary* can think the thought that *Mary loves John*, where the latter thought is a systematic variant of the former.

The systematicity of thought itself, of course, cannot be directly observed. But it is easy to see why it is widely assumed. Anyone who can *understand* a sentence S can understand its systematic variants. Since understanding S requires having the thought it expresses, it follows that anyone who can think the thought expressed by a sentence S

[1] The debate was instigated by Jerry Fodor and Zenon W. Pylyshyn 1988. Connectionism and cognitive architecture. *Cognition* 28:3-71. In the present article, we mainly focus on that paper and on Fodor and Brian P. McLaughlin. 1990. Connectionism and the problem of systematicity: Why Smolensky's solution doesn't work. *Cognition* 35:183-205 (1990) and Paul Smolensky, Géraldine Legendre and Yoshiro Miyata. 1992. Principles for an integrated connectionist/symbolic theory of higher cognition. Tech. Report 92-08. Institute of Cognitive Science. University of Colorado.. See footnote 2 for other papers detailing Fodor's position and footnote 6 for papers detailing Smolensky's. Other important participants in the debate were: Kenneth Aizawa 1997. Explaining systematicity. *Mind and Language* 12:115-36; Keith Butler 1991. Towards a connectionist cognitive architecture. *Mind and Language* 6:252-72; Keith Butler, 1995. Compositionality in cognitive models: The real issue. *Philosophical Studies* 78:153-62; David J. Chalmers 1993. Connectionism and compositionality: Why Fodor and Pylyshyn were wrong. *Philosophical Psychology* 6:305-319; Nick Chater and Mike Oaksford 1990. Autonomy, implementation and cognitive architecture: A reply to Fodor and Pylyshyn. *Cognition* 34:93-107; Robert Cummins 1996. Systematicity. *The Journal of Philosophy* 93:591-614; Robert F. Hadley 1994a. Systematicity in connectionist language learning. *Mind and Language* 9:247-72; Robert F. Hadley Systematicity revisited. *Mind and Language* 9:431-44; Robert F. Hadley and Michael B. Hayward 1997. Strong semantic systematicity from Hebbian connectionist learning. *Minds and Machines* 7:1-55; Robert F. Hadley 1997. Cognition, systematicity, and nomic necessity. *Mind and Language* 12:137-53; Terence Horgan and John Tienson 1991. Structured representations in connectionist systems? In Steven Davis (ed.) *Connectionism: Theory and Practice*. New York: Oxford, pp.195-228; Robert J. Matthews 1994. Three-concept monte: Explanation, implementation, and systematicity. *Synthese* 101:347-63; Robert J. Matthews, 1997. Can connectionists explain systematicity? *Mind and* Language 12:154-77; Lars F. Niklassonand Tim van Gelder. 1994. On being systematically connectionist. *Mind and Language* 9:288-302; Tim van Gelder 1990. Compositionality: A connectionist variation on a classical theme. *Cognitive Science* 14:355-84.

can have the thought expressed by the systematic variants of S. The systematicity of all thought expressible in language seems to follow: If you can understand 'John loves Mary', you must be capable of the thought *that John loves Mary*. Since anyone who can understand 'John loves Mary' can understand 'Mary loves John', it follows that anyone who can have the thought *that John loves Mary* can have the thought *that Mary loves John*.

We get the expression for the thought you must have to understand the systematic variants of 'John loves Mary' by permuting the words in 'John loves Mary' (while maintaining grammaticality), and prefacing the result with 'that'. This leads to what we might call the “orthodox” position defended by Fodor, Pylyshyn and McLaughlin (hereafter FPM[2]) concerning the explanation of systematicity, namely that it is best understood as involving two parts.

(1) The Representational Theory of Thought: having the thought that p is having a p-expressing mental representation in a certain cognitive role.

For example, having a belief that p amounts to having a mental representation that p in the belief box.[3]

(2) Mental representation is “classical”: mental representation has a language-like combinatorial syntax and associated semantics.[4]

Putting these two parts together, we get that anyone who can think that John loves Mary can think that Mary loves John, since (i) thinking Mary loves John involves tokening a representation of the proposition that Mary loves John, and (ii) that representation has constituents corresponding to Mary, John the relation of loving, which can simply be permuted to yield a representation, and hence a thought, corresponding to the proposition that John loves Mary. FPM are thus led to conclude that the human system of mental representation must be “classical,” i.e., a language-like scheme having the familiar kind of combinatorial syntax and associated semantics first introduced by Tarski.[5]

An unfortunate consequence of the way FPM have characterized the systematicity of thought is that any theory that accounts for understanding every sentence will account for systematicity trivially: If one can understand every sentence, one can understand

[2] FPM’s position is fully laid out in the following articles: Fodor and Pylyshyn 1988; Fodor and McLaughlin 1990; Fodor 1997. Connectionism and the problem of systematicity (continued): Why Smolensky’s solution still doesn’t work. *Cognition* 62:109-19; McLaughlin 1992. Systematicity, conceptual truth, and evolution. In Christopher Hookway & Donald Peterson (eds.) *Philosophy and the Cognitive Sciences*. Royal Institute of Philosophy, Supplement no. 34, New York: Cambridge University Press, pp. 217-234; McLaughlin 1993. The connectionism/classicism battle to win souls. *Philosophical Studies*: 71: 163-190.

[3] This assumption of the Representational Theory of Thought is controversial. One might think that a person can have the thought that p without having the representation that p. See Cummins 1996. *Representations, Targets, and Attitudes*. Cambridge, Mass.: MIT Press.

[4] We call these schemes "classical" in what follows (one could also call them "tarskian"). See Alfred Tarski 1936. The concept of truth in formalized languages. In Tarski 1956. *Logic, Semantics, Metamathematics*. Oxford: Oxford University Press, pp. 152-218; Tarski 1944. The semantic conception of truth. In Herbert Feigl and Wilfrid Sellars (eds.) (1949). *Readings in Philosophical Analysis*. New York: Appleton, pp. 52-84.

[5] Tarski, 1936.

every systematic variant of any given sentence. If the domain of sentences to be understood is finite, nonclassical schemes could be complete in the relevant sense, and hence account for systematicity. A look-up table, for instance, that uses arbitrary names to represent each sentence of the domain would do.

A natural reply to this point would be to claim that only representational schemes employing something like classical combinatorics could be complete for an unbounded domain like the domain of thought-expressing sentences in a natural language. However, Smolensky, Legendre, and Miyata (hereafter SLM[6]) have proven that for every classical parser, i.e., a parser defined over classical representations, there exists a tensor-product network that is weakly (input-output) equivalent to the classical parser but does not employ classical representations.[7] Thus, it appears that the classical explanation and the tensor-product explanation are on equal footing as explanations of the systematicity of thought.

There are two *philosophical* responses to the SLM result that we want to surface briefly and put aside. The first, due to Schwarz,[8] is that networks like that designed by SLM do not really have an unbounded competence. The second, suggested by FPM, and discussed briefly by Cummins,[9] is that, while both parsers[10] can account for the

[6] SLM's position is fully laid out in the following articles: Smolensky 1987. The constituent structure of connectionist mental states. *Southern Journal of Philosophy Supplement* 26:137-160; Smolensky 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence* 46:159-216; Smolensky 1991. Connectionism, constituency and the language of thought. In Barry Loewer & Georges Rey (eds.) *Meaning in Mind: Fodor and his Critics*. Oxford: Blackwell, pp. 201-227; Smolensky 1995. Constituent structure and explanation in an integrated connectionist/symbolic cognitive architecture. In Cynthia Macdonald and Graham Macdonald (eds.) Connectionism: Debates on Psychological Explanation. Oxford: Blackwell; Smolensky, Legendre and Miyata 1992. Smolensky and Legendre (forthcoming), Architecture of the Mind/Brain: Neural computation, optimality, and universal grammar in cognitive science. Cambridge, Mass.: MIT Press.

[7] Tensor product encoding is a general-purpose technique for binding fillers to roles (object to properties/relations). An activation vector representing the binding of a filler f to a role R is obtained by taking the tensor product of the vectors representing R and f. The resulting vector can then be added to (superimposed on) others representing bindings of fillers to roles, yielding a single vector that represents the binding of many fillers to many roles. SLM use this technique to construct recursive representations of binary trees. Matrices effecting the vector operation for constructing a tree from its left child p and right child q (or for decomposing it) are then defined, making possible direct implementation in one layer of connection weights.

This connectionist representation of trees enables massively parallel processing. Whereas in the traditional sequential implementation of LISP, symbol processing consists of a long sequence of car, cdr and cons operations, here we can compose together the corresponding sequence of W_{car}, W_{cdr}, W_{cons0}, W_{cons1} operations into a single matrix operation. Adding some minimal nonlinearity allows us to compose more complex operations incorporating the equivalent of conditional branching. (Smolensky, Legendre and Miyata, 1992).

The matrices can then be straightforwardly combined to yield a single layer of connection weights that implement parse-dependent predicates and functions.

[8] Georg Schwarz 1992. Connectionism, Processing, Memory. *Connection Science*, 1, 207-26.

[9] Cummins 1996. Systematicity. *The Journal of Philosophy* 93:591-614.

[10] SLM do not in fact construct a parser. What they do is show how to construct networks effecting parse-dependent functions and predicates (e.g., active-passive) that correspond precisely to arbitrarily complex parse-dependent LISP functions/predicates. In the context of the

systematicity data, the SLM explanation is unprincipled because the tensor-product parser is handcrafted to achieve the effect, while the classical parser is not. We discuss these briefly in turn.

Unbounded competences in connectionist networks. Georg Schwarz has argued that connectionist networks that are not simply implementations of classical architectures cannot have unbounded competences. According to Schwarz, we take an ordinary calculator to have an unbounded competence because it employs perfectly general numerical algorithms for addition, multiplication, subtraction and division. Consequently, its competence is limited only by time and memory.[11] We can add more memory to the calculator, and let it run longer, without altering the algorithms it exploits. But, Schwarz argues, the same does not hold of a connectionist calculator, for the only way to add memory is to add more nodes, or to add precision to the nodes. Doing either of these, however, will require retraining the network, and hence amounts to programming a new system that, in effect, executes a different algorithm. The fact that processing and memory are fully integrated in connectionist networks blocks the standard idealization away from memory limitations that licenses attribution of unbounded competences to classical systems whose controlling algorithms remain unchanged with the addition or subtraction of memory.

We believe this argument should be rejected for reasons given by Cummins.[12] The culprit is the assumption that the function computed is a function from input to output rather than a function from input and initial state to output and final state. The input-to-output relation exhibited by a classical parser, for example is a function of its stored knowledge. Change what the system knows and the same input will yield a different output. Because of this, the input-output relation is seldom a function. This is particularly obvious when we consider learning algorithms. These cannot be conceived as functions pairing inputs and outputs, for the whole point of a learning algorithm is to replace ineffective input-output pairings by better ones. By definition, learning algorithms remain constant over changes in input-to-output relations.

We accept the principle that a different function computed implies a different algorithm executed. But the function computed by a connectionist system must be conceived as a function from an activation vector and point in weight space to another activation vector-weight space pair. From this point of view, we do not build a new network when we change weights any more than we build a new rule based system when we change its stored knowledge.

Connectionist data coverage is "unprincipled." The idea here is that classical representational schemes *predict* systematicity, whereas connectionist schemes at best *accommodate* it.

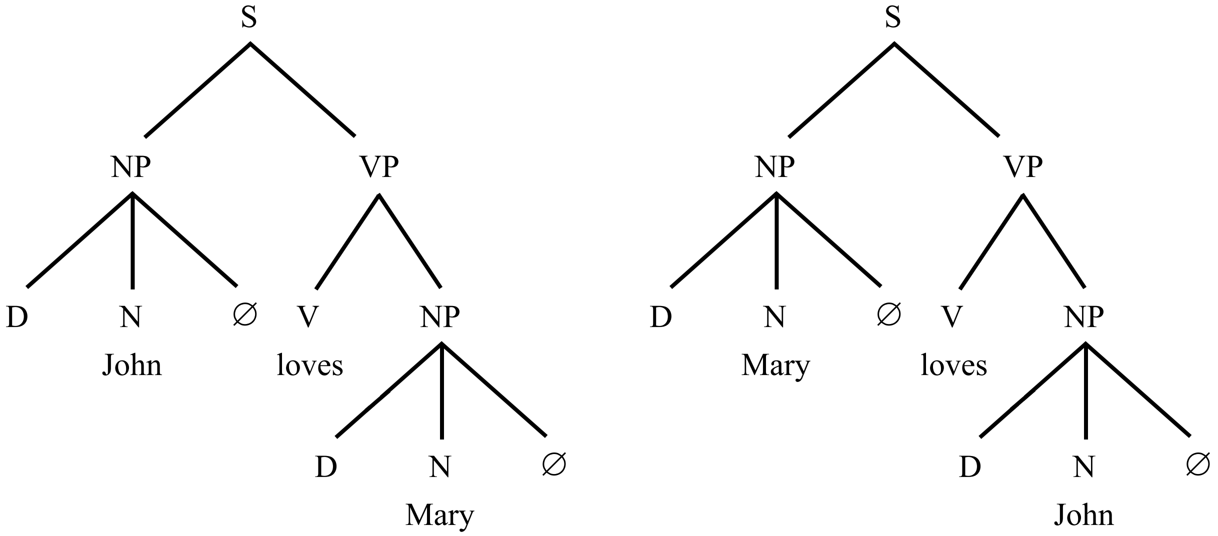
To get a concrete sense of this objection, suppose a classical system generates a phrase marker for 'John loves Mary'. Since 'Mary loves John' has precisely the same phrase marker, except that 'John' and 'Mary' have exchanged positions, the system is

current discussion, this amounts to showing that classical representation is not required for systematic and productive capacities defined over the constituent structure of a sentence.

[11] It is also limited, ultimately, by wear and tear, but this raises no special problem for connectionists.

[12] Cummins, 1996b. Systematicity. This Journal.

bound to be able to parse 'Mary loves John' if it can parse 'John loves Mary': the grammatical structure is the same, and so are the lexical resources.



By contrast, so the argument goes, a connectionist network could be trained to parse 'John loves Mary' but not 'Mary loves John'.

An obvious reply is that a classical system might be programmed to exhibit the very same incapacity. While this is certainly true, the resulting system would be obviously ad hoc. The incapacity does not appear ad hoc in the connectionist framework, however, since the representations of the sentences do not themselves have constituents corresponding to the relevant lexical items. It might seem, therefore, that there is no reason why the network would have to process the two representations in a similar way. They are simply different activation vectors, and hence there will be no special problem about a training set that simply declares the second parse unacceptable. Bias against a systematic variant of an acceptable sentence will present no learning problem different in principle from bias against any other unacceptable sentence, e.g., 'Loves Mary John'. A classical system, however, will give both parses if it gives either, unless some special rule is added to the grammar to block one or the other. There will be no rule or process in the deviant connectionist system that is special to this case. The same setting of weights that deals with every other sentence, whether acceptable or unacceptable, will deal with the deviation. There will, in short, be no principled distinction between the way it rejects 'Mary loves John' and the way it blocks 'Loves John Mary'.

Here is a different but related way to see the alleged problem. Imagine a classical system that can parse 'John loves Mary' but cannot parse 'John despises Mary' because 'despises' is not in its lexicon. It seems intuitively plausible to suppose that a simple learning algorithm will add 'despises' to the lexicon, and that this will suffice to yield the new parse. SLM’s result shows that for each of these classical parsers, there is a correlate, weakly equivalent, connectionist parser. It is silent, however, about the relation between these two correlate connectionist parsers. In particular, it is silent about whether the correlate connectionist parser that handles 'despises' is in any sense a natural extension of the one that does not. For all the SLM result shows, the connectionist parser that handles 'despises' would have to be built from scratch rather than as a simple modification of the one that does not. By itself, SLM’s result does not show that the networks corresponding to the pre-and post-learning stages lie on any common connectionist learning trajectory. It shows only that *any* set of parses that can be captured by a classical system can be captured by a connectionist network. This at least suggests the possibility that the connectionist network does not parse the systematic variants of S *because they are systematic variants of an acceptable sentence*, but rather that it parses them simply because they are among the sentences in the target set defined by the classical system to be emulated. We have, in short, no reason to think that all the systematic variants of S are parsed in the same way. Rather, the prediction that the network will parse a systematic variant of S is derived simply from the facts (i) that the systematic variants of S are acceptable if S is, and (ii) that the network parses all the acceptable sentences.[13]

We will come back to the merits (and demerits) of the objection that connectionist data coverage is unprincipled in a later section. But notice that, at this point, the debate has shifted from empirical considerations of what best covers the data to philosophical considerations of what constitutes a principled explanation. We propose, therefore, to reformulate the issue in the hope of finding a way of retaining its essentially empirical cast.

II. The Issue Reformulated

The fact that a system cognizes a domain will manifest itself in a variety of psychological effects. By an *effect*, we mean a nomic regularity in behavior (what Millikan[14] calls a law *in situ*, i.e., a law that holds of a system in virtue of the special structure and organization of that system). For instance, the fact that humans cognize the color domain is manifested by the fact that humans make such and such discriminations (and fail to make others), by the fact that they will eat some foods and not others, etc.

We take it as uncontroversial that some domains are cognized by grasping their underlying structure. For example, we can recognize the melody of 'Mary Had A Little Lamb' whether it is played by an orchestra or on a kazoo, regardless of what key it is played in and, within limits, regardless of tempo. Sensitivity to melody across differences in timbre, tempo and key suggests that the processing of melody is done by processing information about the structure and arrangement of notes in a composition. Word problem solving in algebra is another case where grasping the underlying structure of a domain is required for cognizing the domain. Students are better able to solve word

[13] This problem is an instance of a more general problem we address elsewhere and will be pursued in further publications. See Pierre Poirier, Robert Cummins, James Blackmon; David Byrd; Martin Roth & Georg Schwarz. 1999. The epistemology of non-symbolic cognition: atomistic learning and forgetting. Tech. Report Phil99-3, University of California at Davis; see also Michael McCloskey and Neil J. Cohen 1989. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *The Psychology of Learning and Motivation*, 24, 109-165. Connectionist systems using distributed representation cannot learn atomically, that is, they cannot add single representations, such as the predicate 'despise', to their belief box or knowledge base without having to relearn everything they previously learned. Surely, this is not the way we acquire new representations in many domains. (But, for a possible solution, see James L. McClelland, Bruce L. McNaughton & Randall C. O'Reilly (1995). Why there are complementary learning systems in the hippocampus and the neocortex: Insights from the success and failures of connectionist models of learning and memory. *Psychological Review* 102: 419-57).

[14] Ruth Millikan 1984. *Language, Thought, and Other Biological Categories.* Cambridge, Mass.: MIT Press. A Bradford Book.

problems involving distances, rates, and times when they see the problems as having the same structure as problems they are already familiar with. For example, students who can solve problems involving wind speeds but have difficulty solving problems involving current speeds are able to perform well on the latter when it is pointed out that current is analogous to wind.[15] Only when students recognize an underlying structure they are familiar with do they seem to be able to give the correct answer (this suggests that sensitivity to structure is learned in some cases, or at least affected by prior knowledge). We also take it to be uncontroversial that some domains are cognized without grasping any significant underlying structure: knowing the capitals of twenty US states is no help in divining the capital of an unstudied state. Our ability to learn the state capitals does not depend on being sensitive to underlying structural features of state and their capitals. With respect to the previous distinction, we call an effect a *systematicity effect* if it is a psychological effect that implies sensitivity to the structure of the domain cognized.[16]

Is it possible to draw any conclusions about the form of mental representation from the presence of systematicity effects? We think it is. Recall that our main methodological objective here is to keep the issue at an empirical level in the hope that it can be solved there. Our first conclusion proceeds from the observation that not all systematicity effects are created equal: some may be computed by the system and some may be incidental effects of whatever algorithm is being computed. We call the first type of systematicity effect “primary” and the second “incidental,” and we address the distinction and what it tells us about mental representation in the next section.[17] Our second conclusion rests on a finer analysis of SLM’s tensor-product representations. Unlike classical representations that preserve information about the structure of represented elements by actually sharing that structure, tensor-product representations do not share structure with what they represent yet still manage to preserve structural information and make it available to processors. This allows a possible solution to a problem raised by Cummins[18] against classical representations—namely that they cannot possibly share structure with every domain in which we find systematicity effects—but the price for that solution may be one FPM are not ready to pay. We address this issue in section four.

[15] Denise Cummins 1992. Role of analogical reasoning in the induction of problem categories. *Journal of Experimental Psychology: Learning, Memory, & Cognition* 5:1103-1124.

[16] These examples are meant to represent the extreme cases. There are likely to be cognitive tasks that require grasping varying degrees of structure depending on the domain, so the distinction between structured and unstructured is not all or nothing. Also, some cognitive tasks will be hybrid cases, where some aspects require grasping structural information while others do not. Cognizing language appears to be an example, since mastery of the primitive lexicon is surely more like learning the state capitals than like mastering the syntax.

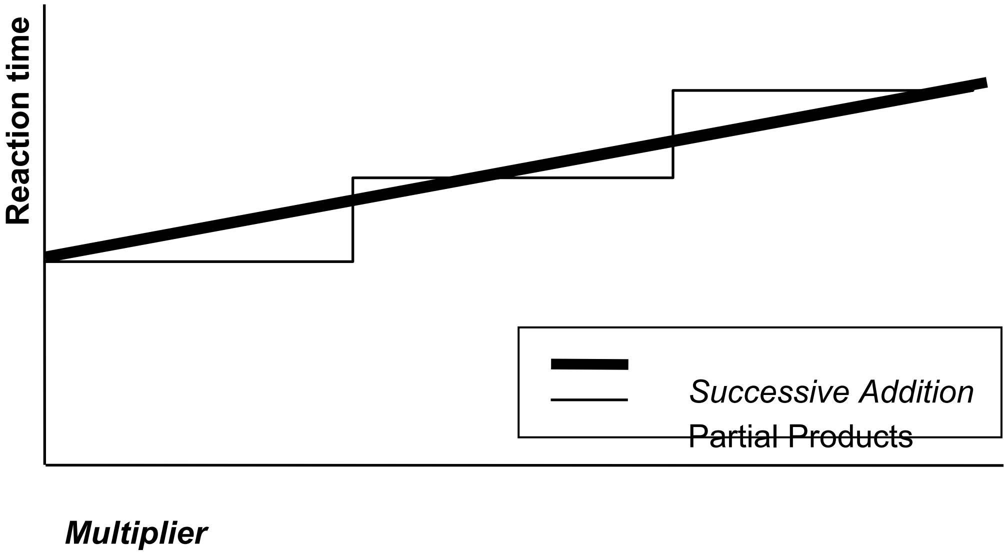
One must also be careful not to confuse structure in the domain with structure in the way a problem or question is posed. When a teacher requires that I learn the capital of California, the structure of my answer is determined in part by the structure of the question itself. Similarly, one can imagine a state capital learning device that represents states as one place predicates such that only one capital could be matched with any state. In this case the structure of the answer is partly determined by the way states are represented in the system.

[17] The distinction between primary and incidental systematicity effects is introduced in Cummins, 1996, This Journal, and also discussed in R. Cummins 2000. ‘How does it work?’ versus ‘What are the laws?’ Two conceptions of psychological explanation. In F.C. Keil et R.A. Wilson (eds.) *Explanation and Cognition*. Cambridge, Mass.: MIT Press, pp.xxx-xxx.

[18] Cummins, 1996. Systematicity. This Journal.

III. Primary vs. Incidental Systematicity Effects

The distinction between primary and incidental effects captures the important fact that a system's behavior results not only from the function it computes, but from a variety of other factors as well. Compare a system that multiplies by partial products with a system that computes products through successive addition. What the two systems have in common is the "multiplication effect," i.e., the fact that they both produce the same products from the same arguments. They differ, however, in how long it takes them to compute the value for a given argument. In the case of the successive adder, the response time is roughly proportional to the size of the multiplier: computing ten times some number will take approximately twice as long as computing five times that number. The other system, in contrast, does not display such a linearity effect. Its response time is, roughly, a step function of the number of digits in the multiplier, indicating how many partial products need to be added in the end.



Products are computed (primary)
Reaction time is incidental

The two systems have the same primary systematicity effects, since every argument pair leads to the same value in both systems: they are both multipliers. However, their incidental systematicity effects are different. For instance, the partial product multiplier will take approximately the same amount of time to multiply *N* by 10 as it will to multiply *N* by 99, whereas the successive addition multiplier will take roughly 10 times longer to multiply *N* by 10 as it will to multiply *N* by 99. In general, two systems that compute the same function using different algorithms will display different incidental effects, although they will be weakly equivalent in Pylyshyn's sense.[19]

One important source of incidental effects thus lies in the algorithm producing the primary effects. But systems that implement the *same* algorithm may exhibit different incidental effects. If the underlying hardware is sufficiently different, they may operate at

[19] Pylyshyn 1984. *Computation and Cognition.* Cambridge, Mass.: MIT Press. A Bradford Book.

greatly different speeds, as is familiar to anyone who has run the same program on an Intel 486/50 MHz and on a Pentium III running at 750 MHz. A significant difference in response time between two systems, in other words, does not entail that they implement different algorithms; it may be a direct result of a difference at the implementation level. Hence, two systems that have identical primary effects may exhibit different incidental effects either because they compute different algorithms or because the identical algorithms they compute are implemented in different hardware.

Incidental effects are not restricted to the complexity profiles of the algorithm or the constraints imposed by the implementing matter on the execution of the algorithm. Continuous operation will heat up the calculator, an incidental effect that is normally irrelevant to the system's performance. Nonetheless, when things get too hot, the system will start malfunctioning and ultimately break down. As a consequence, two systems that differ in performance do not necessarily compute different functions; one of the systems may simply have been subject to extraneous factors such as overheating, fatigue, or ADD (attention deficit disorder). By the same token, the general physical makeup of a system also determines the extent to which environmental conditions will have an impact on the system's ability to operate. The occurrence of a strong magnetic field, for example, will interfere with the operation of an electronic calculator but not with that of an abacus. In the case of single-purpose machines, this last kind of incidental effect is primarily relevant for explaining why a system fails to display its primary effects on certain occasions. But in the case of more complex systems, such effects have proven useful as an explanatory tool as well. Consider the fact that cognitive processing in the human brain is generally correlated with increased metabolic activity, a fact that has proved critical for the use of imaging technology (e.g., PET, fMRI) to study which cortical areas are involved in the processing of these tasks. Yet metabolic activity as such is not specific to the neural implementation of whatever computations are performed.

It is evident that the effects of implementation details and environment complicate inferences from sameness of incidental effects to sameness of underlying functional architecture. However, in what follows, we will make use only if the inference from differences in incidental effects to differences in functional architecture in cases in which implementational and environmental influences are not at issue.

Now SLM have proven that a connectionist tensor-product parser and the classical parser can exhibit the same primary effects (they are weakly equivalent, like our two multiplication algorithms). Any parse a classical parser computes can be computed by a corresponding connectionist parser. If systematicity effects are primary effects, then SLM have demonstrated, mathematically, that systematicity effects in language parsing can be accounted for without any appeal to classical representations. Hence, if the systematicity effects at issue are primary effects, then nothing can be concluded concerning whether the mental representations involved are classical or connectionist in form. As we saw, the consequence of that empirical deadlock has been to turn away from empirical considerations to philosophical issues concerning what constitutes a principled explanation. However, SLM's tensor-product parser employs a different algorithm than the classical parser, and thus the two systems are bound to exhibit different incidental effects. If the systematicity effects observed are incidental, then, barring cosmic coincidence, at most one of the two parsers can account for them. In particular, the classical explanation can gain leverage over the SLM explanation if the form in which information is represented is important to explaining the effect. An incidental effect of using classical representations in language processing might, for instance, involve priming. Since the lexical items and sentential structure of a recently processed sentence S may be easily accessible (they may still be in short-term memory, for instance), the recent processing of S may allow the system to process systematic variants of S faster than it would process sentences that are not systematic variants of S. An incidental effect of using connectionist representations is the flat temporal profile of parsing with a tensor-product parser. Since the latter will process all sentences, no matter how complex, in one step, it will process all of them in exactly the same amount of time. Should the empirical evidence show that subjects exhibit the kind of priming effect described above or a temporal profile different from the one we would expect from the use of a tensor-product parser, then, weak equivalence notwithstanding, FPM have a good argument in favor of their claim that classical representations better explain the systematicity of thought. Of course, should the evidence show that the incidental effects are those one would expect from an SLM-type parser, then it would seem that classical representations are not the source of the systematicity of thought.

FPM and their supporters cite no evidence from incidental systematicity effects, nor do their opponents. It seems likely that there are relevant effects reported in the literature on language processing, but they have not, to our knowledge, been brought to bear on this issue.[20] It should be emphasized, however, that systematicity effects of the sort FPM had in mind are clearly not incidental but primary[21] : they do not argue from evidence that processing Mary loves John makes it easier to process John loves Mary to the conclusion that mental representation is classical. They simply argue from the availability of a given thought (sentence understood) to the possibility of its systematic variants.

IV. Representational Pluralism and Structural Encodings

The inference from systematicity effects to classical representations involves three steps:

1) The observation of systematicity effects.

2) An inference from the presence of these effects to the conclusion that mental representations must preserve and carry information about the structure of the domain.

3) An inference from that conclusion to the further conclusion that the information about structure must be carried by classical representations.

Most will readily agree that systematicity effects can be observed. We noted in the previous section that care should be taken to distinguish primary from incidental systematicity effects since only the latter will allow conclusions to be drawn about the nature of representations. Step 2 is also uncontroversial. In unbounded domains (see above), how else can these effects be produced? It is step three of the inference, from the preservation of structural information to the necessary presence of classical representations, that we wish to address here.

[20] We are in the process of searching the literature for such evidence. Meanwhile, the fact that the SLM parser accomplishes all parses in a single step, regardless of the complexity of the sentence parsed, is surely suggestive.

[21] This is put misleadingly in Cummins, 1996, This Journal, p. 603. The point is that proponents of the systematicity argument are thinking of systematicity effects as primary when they should be thinking of them as incidental.

Sensitivity to the structure of a domain is best explained by a scheme of mental representation that carries information about the structure of that domain. One such scheme involves structural representations of the domain in question. A representation R is a *structural representation* of its target just in case R and its target share structure. Scale models, photographs, and maps are typical examples of structural representations. As Cummins[22] points out, it is natural to explain the systematicity effects we find in language, vision, and audition by positing structural representations of each of the respective domains. The obvious virtue of structural representation is that structural transformations and permutations of the representations yield representations of the systematic variants of the corresponding targets in the cognized domain. For every operation defined over elements in a domain there can be a corresponding operation defined over elements in the representational scheme. Thus, the representation of an item in the domain can be used to construct a representation of its systematic variants. The sensitivity to structure that is required to cognize certain domains is accomplished by actually having the structure of the domain in the representations. We are now in a position to appreciate that FPM's classical representations are a case of a scheme of structural representation for the linguistic domain. The idea is that, in order to be sensitive to the combinatorial syntax and associated semantics of a language, there must be a system of internal representations that has the same (or corresponding) syntactic and semantic features.

This way of explaining systematicity effects, however, limits classical schemes to the explanation of systematicity effects exhibited by the cognition of language-like domains, since these are the only domains that classical schemes can structurally represent. This leaves the systematicity effects we find in other differently structured domains to be explained by appeal to non-classical schemes. Since we apparently grasp the underlying structure of domains structurally distinct from language, we would require structurally distinct representational schemes to cope with each of them in the same way that classical schemes are supposed to facilitate coping with language. Apparently, structural representation comes at the price of (perhaps massive) representational pluralism.

The problem of representational pluralism was first noticed by Cummins.[23] Roughly, representational pluralism is the idea that for every differently structured domain for which we show a systematicity effect, we employ a scheme of mental representation that shares structure with that domain. Thus, if we exhibit systematicity effects in cognizing three differently structured domains, we employ at least three differently structured schemes of mental representation. FPM can make a plausible argument from systematicity effects in language processing to the conclusion that some mental representation is classical. However, there are other systematicity effects, such as those found in vision and audition, that cannot be accounted for by structural representations if all mental representation is classical, since the structure of the domains in language, audition, and vision are different. Since the inference from systematicity effects in domains other than language to nonclassical, structural representations is on par with the inference from systematicity effects in language to classical representations, the systematicities in vision and audition are good evidence for some nonclassical, structural representations. Thus, if the FPM inference is sound, it constitutes a good argument for representational pluralism. The only way FPM can retain the idea that mental representation is monistic is by allowing that some systematicity effects can be

[22] Cummins, 1996. Systematicity. This Journal.

[23] Cummins, 1996. Systematicity. This Journal.

adequately explained without recourse to a corresponding scheme of structural representations. Some systematicity will have to be explained by appeal to mere *encodings*. An encoding of a domain D is a mapping of the members of D onto the representations in a scheme R whose members do not share the structure of their images in D. Classical representations structurally represent linguistic structure, but they only encode the structure of music. By allowing some encoding, however, friends of FPM would forfeit the objection that connectionist data coverage is unprincipled, since encoding, on their view, forces us to be unprincipled somewhere. So it seems that either the objection must go, or FPM are forced to accept representational pluralism.

The SLM parser shows, however, that representational monism is compatible with an adequate account of primary systematicity effects. To see this clearly, we need a taxonomy of representational schemes that 1) makes evident what property holds of structural representations in virtue of which they account for primary systematicity effects, and 2) identifies an alternative sort of scheme that can be seen to have this critical property.

Our taxonomy relies on two distinctions. The first distinction has to do with whether a representational scheme represents its domain by making available representations that share structure with the items of the domain represented. Shared structure involves two things. First, the representations have constituents that represent the constituents of the domain. Second, these representational constituents are structurally related in a way that represents the ways in which the constituents of the content are structurally related. We call any scheme in which representations represent items in its target domain in virtue of sharing structure with the things they represent a structural representational scheme for that domain. We call schemes that represent a domain, but not in virtue of shared structure, an encoding of the domain. So the first distinction separates entities that represent in virtue of shared structure, that is, structural representations, from those that represent by some other means, that is, encodings.

The second distinction we need requires the notion of a recovery function. A recovery function is a function from a representational scheme to a domain. Intuitively, these functions allow one to recover contents from representational schemes. Recovery functions are not necessarily employed by cognitive systems; they may remain purely theoretical constructs by which we may interpret representational schemes. We distinguish between recovery functions that are *systematic* and those that are not. A recovery function is systematic if there is a general/productive algorithm for mapping the scheme to its contents. In such a case, the function does not need to be defined by a list of pairs each containing an element of the scheme and a content. If the recovery function is systematic, then the representational scheme is structural. If the recovery function is not systematic, then the representational scheme is pure or arbitrary.

	Systematic Recovery Function	**No Systematic Recovery Function**
Do not share structure	Structural Encodings	Pure Encodings
Share structure	Structural Representations	

This taxonomy allows us to distinguish three categories of representational entities: structural representations, structural encodings, and pure encodings. Schemes that belong to either of the first two categories are such that the information about the structure of the domain is systematically recoverable. This is relevant because if structure can be recovered systematically from a scheme, then the scheme has preserved information about the structure of the domain. The possibility of systematic recovery implies the preservation of structure.

One obvious way of preserving structure is for the representational scheme to simply share the structure of the domain. This is what structural representations do. But this is not the only way. Structural encodings preserve structure without sharing it. Gödel numbering and tensor product schemes are both examples of structural encodings. In the Gödel scheme for encoding sentences, words are assigned natural numbers while their positions in the sentence are assigned prime numbers in ascending order. A number m in position n yields the number n^m. We can say that n^m stands for the word numbered m standing in the place numbered n. This number is uniquely factorable into n, m times. The Gödel string of a sentence is determined by multiplying all these uniquely factorable numbers together. This yields a number, expressed by a Gödel string, which is uniquely factorable into a list of numbers. An example of a Gödel string is '1,093,500' which factors into $2^2 \times 3^7 \times 5^3$ which breaks down further into a list of 2's, 3's, and 5's. The values of these numbers have been assigned to places and the number of occurrences of any number has been assigned to words. We may suppose the string encodes the sentence 'Mary loves John' where 'Mary', 'loves', and 'John' are assigned the numbers 2, 7, and 3, respectively. The Gödel string itself does not share structure with its content; however, this structure is systematically recoverable and thus the representation has preserved information about the structure of the sentence. SLM's tensor-product representations have the same property. The tensor-product activation patterns themselves do not share structure with their contents, but the existence of recoverability functions for such schemes entails that such schemes preserve structure.

Structural encodings permit recovery but do not share structure with their domains. Pure encodings do not permit systematic recovery and so do not preserve structure. If, for example, we were to take the names of the state capitals as encodings of the names of the states we would be employing a pure encoding scheme, for there is no systematic way by which the names of the states could be recovered from the names of the capitals.[24] In this case, a look-up table is required, and no information about structure is preserved.

Structural encodings, as opposed to pure encodings, are adequate to account for primary systematicity effects because, under such schemes, information about structure is preserved and that is all that is required. This is, in fact, the property which holds of structural encodings and structural representations in virtue of which it is possible to construct architectures using either one that exhibit systematicity effects. To take our previous example, any system capable of processing 1,093,500 (the Gödel string for 'Mary loves John') is also capable of processing 437,400 (the Gödel string for 'John loves Mary'). The parallel with the previous symmetric phrase markers is perfect. The tensor product scheme employed by SLM is a structural encoding, not a pure encoding

[24] Another way to see the same point: Given a new capital name, there is no way to infer what state name it corresponds to.

as FPM seem to assume, and this is why SLM are able to prove their otherwise puzzling result. It is only when cognitive systems employ such structure-preserving schemes that they can be causally sensitive to the structure of the domain cognized, and thus exhibit systematicity effects. Systems employing pure encodings cannot exhibit such effects because there is no common form or structure to which the system could be causally sensitive.

Once we see that structural encoding is adequate to account for systematicity effects—at least the primary ones—we are free to cleave to representational monism. Since a scheme like that employed by SLM need not share structure with the domains it represents in order to account for primary systematicity effects, such a scheme could, in principle, account for primary systematicity effects in a variety of structurally distinct domains. The same point can be made about classical representation: since it can be used to structurally encode domains such as music that are structurally unlike language, a classical scheme could account for primary systematicity effects in non-linguistic domains. Since classical and nonclassical structural encodings are evidently on a par in this respect, we conclude again that there is no sound argument from primary systematicity effects to classical representation. There is, of course, a sound argument from systematicity effects against pure encoding, but that should come as no surprise. What is important to systematicity is preserving information about the relevant structure of the target domain. It does not much matter how that information is encoded, provided an architecture exists that is capable of exploiting the information in that encoding. SLM demonstrate that connectionist networks can exploit such information as encoded in tensor-product activation vector schemes.

The staunch defender of classical representations may object at this point that there still is a methodological virtue in accepting classical representations-plus-pluralism over encodings-plus-monism because it is the lawfulness of systematicity that needs to be explained and encodings just do not account for it. But it not clear that they do not. We agree that with pure encodings there could be minds that encode Mary loves John without encoding John loves Mary, since the ability to encode the former does not imply the ability to encode the latter. With structural encodings such as the tensor product scheme employed by SLM, however, matters are different. Since the encodings are generated recursively from the filler and role vectors, the ability to encode 'Mary loves John' does imply the ability to encode 'John loves Mary', since the two encodings employ the same filler and role vectors. The only difference between them is how the vectors get multiplied and added to create the respective tensor-product encodings of the two sentences.[25]

[25] Versions of this article were presented before the Society for Philosophy and Psychology (June 1999), CogSci99 (July 1999), and the APA (April 2000) and at various universities. We thank audiences and commentators for their helpful remarks. Research was funded by the National Science Foundation, Grant # 9976739.