Synchronous Firing Variable Binding is a Tensor Product Representation with Temporal Role Vectors

Bruce B. Tesar & Paul Smolensky
Department of Computer Science
University of Colorado
Boulder, CO 80309-0430 USA
{tesar,smolensky}@cs.colorado.edu

Abstract

Synchronous firing of neural units has recently been proposed as a new way of solving the variable binding problem in connectionist networks. Firing synchrony appears to be unrelated to earlier methods of variable binding, nearly all of which can be analyzed as species of tensor product representations, where vectors representing variables and values are bound together with the outer product. In this paper, we argue that, despite appearances, firing synchrony is also a case of tensor product representation. This analysis exposes two logically independent components of the synchronous firing idea. The most obvious is the idea of using time as a resource: spatio-temporal patterns of activation are used. This, we argue, is a purely implementational issue which does not bear on the complexity issues of variable binding. In contrast, the second idea does bear on genuinely representational issues, and is the source of most of the formal properties claimed for the synchrony scheme. Rather than *explicitly* binding a semantic role like giver to a semantic filler like John, these two are *implicitly* bound—by explicitly binding each to a common formal role, via the tensor product. The analysis situates synchronous firing in a typology of alternative variable binding schemes.

The Variable Binding Problem

A classic obstacle for connectionist networks processing structured data is the *variable binding problem*. One aspect of this problem is the binding of fillers to semantic roles, such as those distinguishing the arguments of a predicate. For example, the predicate give(x,y,z)—'x gives z to y'—has three semantic roles: giver, recipient, and give-object. (Here and throughout, we use the notation and terminology of Shastri & Ajjanagadde 1993). A proposition such as give(John, Mary, book) may be understood as having three variable bindings: giver = John, recipient = Mary, and give-object = book.

Binding by Synchronized Firings

A recent solution to the variable binding problem is inspired by phase synchronization of neurons, a suggested biological mechanism of feature segmentation and linking (von der Malsburg & Schneider 1986). Much of the recent biological data and modeling has focused on perceptual modalities, especially vision (Gray, Konig, Engel, & Singer 1989; Eckhorn, Reitboeck, Arndt, & Dicke 1990). Neurons functioning as feature detectors fire synchronously (or 'in phase') with other neurons responding to other features of the same entity, and out of phase with neurons responding to features of other entities.

Shastri and Ajjanagadde (1993) have proposed phase synchronization as a connectionist solution to the general dynamic variable binding problem, as have Hummel & Biederman (1992). They present their binding representation scheme as a more biologically plausible alternative to other kinds of connectionist variable binding schemes.

The representation system proposed in Shastri & Ajjanagadde (1993) uses single binary-valued nodes to represent roles and fillers. For the proposition give(John, Mary, book), separate, single nodes represent each of giver, recipient, give-object, John, Mary, and book (see Fig. 1 for an illustration of this example).

Time is thought of as divided into cycles, each cycle having duration P (the period of the nodes). An active node fires once per cycle (inactive nodes don't fire at all). Two nodes are said to be *bound* together if they are both active, and their firings are synchronized, that is, they fire at the same time during each cycle.

Because each firing unit has a fixed pulse width W during which it is on, the number of independent sets of synchronized firings that can be represented within a cycle is P/W which we'll call N. A cycle may thus be viewed as a set of N 'binding slots,' with each slot permitting the representation of the simultaneous binding of a set of nodes. An active node occupies exactly one of the binding slots by firing during the part of the cycle corresponding to that slot.

Thus in the give example above, the first slot is occupied by the give-obj/book binding, the second slot by recipient/Mary, etc. We will call the first slot the first *formal role*; this formal role is occupied by both give-obj, the *semantic role*, and by book, the *semantic filler*. These two elements which occupy the first formal role comprise the *formal filler* of that role. The second formal role is occupied by the formal fillers recipient and Mary; one of these is a semantic role, the other a semantic filler. The crux of our analysis is summarized in Table 1.

Tensor Product Representations

On the face of it, phase synchrony binding seems a completely new method of solving the variable binding problem, totally unrelated to other techniques. One of these techniques, which generalizes many more specialized methods, is *tensor product representation*. In this technique, a structure is viewed as a set of bindings of formal roles to formal fillers. For example, the string ABC may be regarded as a set of three bindings: the first formal role is 'first position', which has formal filler A; the fillers of the second and third roles are B and C, respectively.

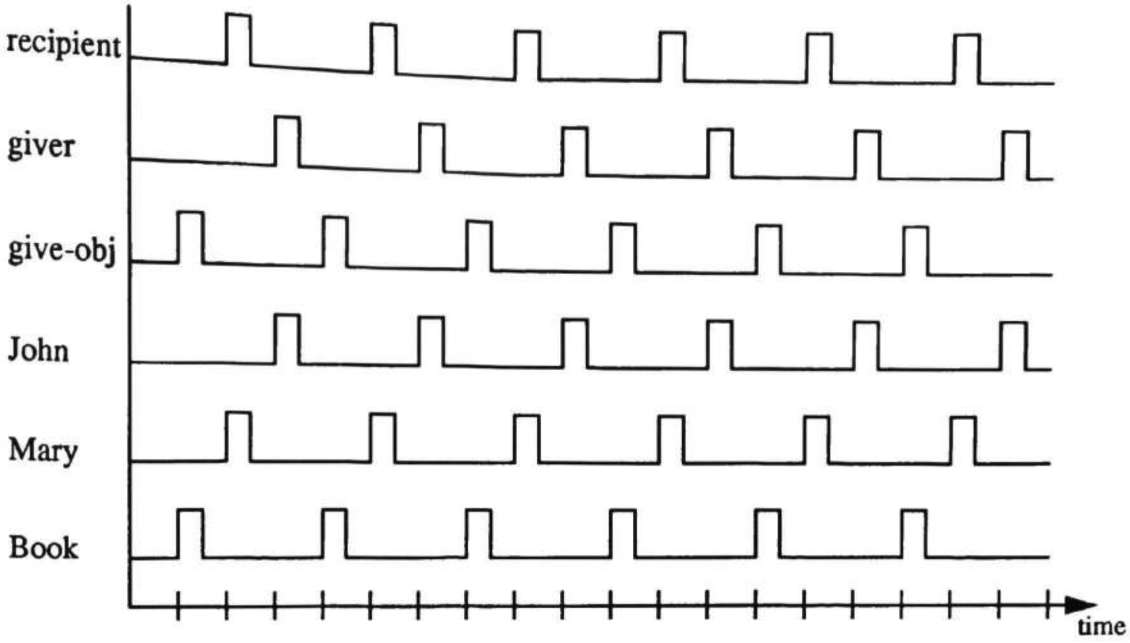

Fig. 1: Temporal synchrony representation of give(John, Mary, book), with N=3 binding slots per cycle (Shastri & Ajjanagadde 1990). Each label on the vertical axis denotes a single unit.

Table 1: Key to the Analysis

formal role = 1 or 2 or ... or N	*e.g.:* formal role = 1 *is occupied by*
formal filler = { a semantic role, its semantic filler }	formal filler = {give-obj, book}

Each formal role is bound to its formal filler with the tensor product operation, and the roles and fillers are both represented by patterns of activation with the structure of a tensor. The properties of tensor product representations, and their relations to symbolic structures in connectionist models generally, are discussed extensively in Smolensky (1990). Tensor products have been used in models of memory (e.g., Humphreys et al. 1989) and analogy (Halford et al., 1994) and in grammatical theory (Legendre, Miyata & Smolensky 1990 et seq.).

Here, a tensor may be thought of as a multi-dimensional array with multiple indices and real valued elements. The number of indices that a tensor possesses is called its *rank*. A rank 2 tensor is just a matrix; it has two indices, as in T_{ij}. A rank 1 tensor is a garden-variety vector: it has one index, as in T_i. Tensor product representations of *recursive* structures require tensors of rank higher than 2; here we need only tensors of rank 1 and 2 (vector and matrix).

Let $\mathbf{v}$ denote a vector (rank 1 tensor) of *dimensionality* $d_{\mathbf{v}}$: $\mathbf{v}$ consists of $d_{\mathbf{v}}$ elements, the ith element being v_i. Let $\mathbf{u}$ denote another vector, with dimensionality $d_{\mathbf{u}}$. The *tensor product* of $\mathbf{u}$ and $\mathbf{v}$, denoted $\mathbf{u}\otimes\mathbf{v}$, is defined to be a rank 2 tensor $\mathbf{T}$ with indices having the dimensionalities $d_{\mathbf{u}}$ and $d_{\mathbf{v}}$ (in that order), and with elements defined as $T_{ij} = u_i v_j$. The tensor product is not commutative ($\mathbf{u}\otimes\mathbf{v} \neq \mathbf{v}\otimes\mathbf{u}$), because $u_i v_j \neq v_i u_j$ (unless of course $\mathbf{u} = \mathbf{v}$). (For $\mathbf{u}\otimes\mathbf{v}$, the first index has dimensionality $d_{\mathbf{u}}$, the second $d_{\mathbf{v}}$; for $\mathbf{v}\otimes\mathbf{u}$, the dimensionalities reverse.) When $\mathbf{u}$ and $\mathbf{v}$ are rank 1 tensors—regard them as column vectors—the tensor product is the same as the *outer product* of matrix algebra: $\mathbf{u}\otimes\mathbf{v}$ reduces to $\mathbf{u}\mathbf{v}^{\mathbf{T}}$.

Addition of tensors is the straightforward generalization of addition of vectors: each element of the resulting tensor is the sum of the corresponding elements of the addend tensors. Two tensors may only be added together if they have the same number of indices and each pair of corresponding indices have the same dimensionality.

A tensor product representation binds a formal filler to a formal role by taking the tensor product of the tensor representing the filler and the tensor representing the role. Multiple bindings may be combined into one tensor representation by superimposing the representations of the individual bindings (that is, by summing the tensor representations): see Fig. 2.

Synchronized Firing as a Tensor Product

Figs. 1 and 2, depicting the representations using temporal synchrony and tensor product variable binding, suggest that the two schemes are entirely different. This is an illusion, however; dissolving this illusion is our main goal here.

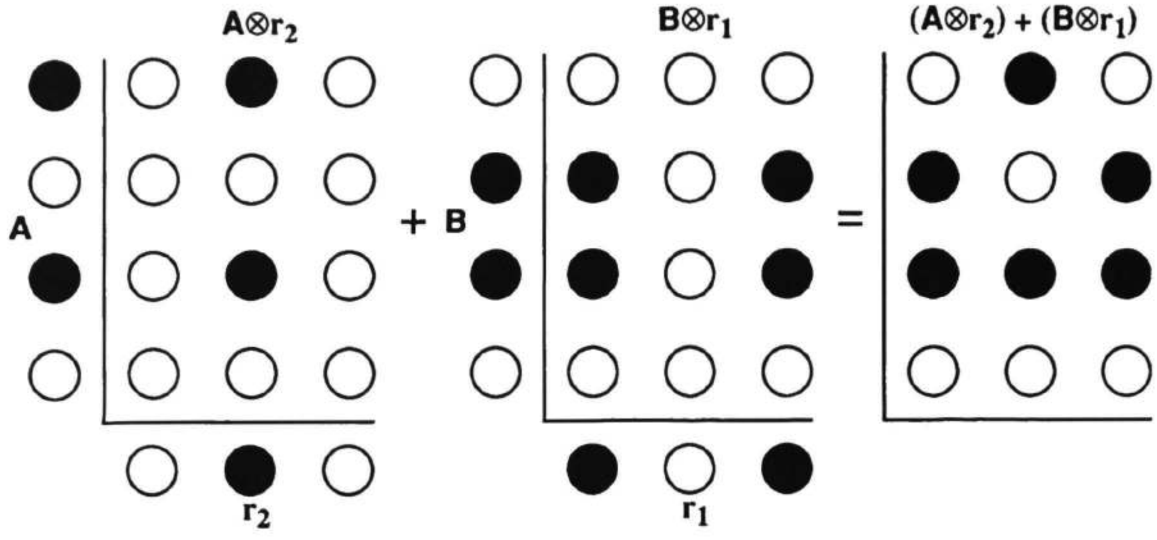

Fig. 2. A tensor product representation of BA. The formal roles are r_1, r_2—first and second position—represented by the vectors $\mathbf{r_1}$ = (1 0 1), $\mathbf{r_2}$ = (0 1 0); the fillers are **A** = (1 0 1 0), **B** = (0 1 1 0). The representation of BA is **BA** = $\mathbf{A} \otimes \mathbf{r_2} + \mathbf{B} \otimes \mathbf{r_1}$, a rank-2 tensor.

Fig. 3a illustrates our analysis of temporal synchrony binding as a kind of tensor product representation. For each 'real' unit, e.g., John, in the temporal synchrony scheme, there is now an entire *row* of 'virtual' units, each showing the activity of John during one time slot during one time cycle. We've simply replaced a bumpy activity trace with a row of virtual units showing the same activation values. Fig. 3a is a *space-time diagram* of an activity pattern: the vertical axis is space, the horizontal axis time. A row of virtual units in Fig. 3 shows the activity history over time of a single real unit in Fig. 1; a column of units in Fig. 3 shows the activity pattern over the whole Fig. 1 network at a single moment of time.

Fig. 3b shows one of the bindings in the full proposition give(John,Mary,book) shown in Fig. 3a. This binding, book/give-obj, is the tensor product of the formal filler vector $\mathbf{f_1}$ shown along the right edge and the formal role vector $\mathbf{r_1}$ shown along the bottom. The formal role vector $\mathbf{r_1}$ has activity value 1 during the first 'slot' of each time cycle. The formal filler vector $\mathbf{f_1}$ has activity value 1 in the locations corresponding to the units for give-obj and book; the subtlety is that the *formal* filler includes both the *semantic* filler (book) *and* the *semantic role* (give-obj). The *formal* role has no relation to the *semantic* role, which is part of the formal filler. Rather than using the tensor product to directly bind the semantic role and semantic filler, these are implicitly bound together in virtue of both being explicitly bound (via the tensor product) to a common formal role, i.e. time slot. In equations: rather than **book⊗give-obj** we have:

$$[\text{book} + \textbf{give-obj}] \otimes \mathbf{r_1} = \textbf{book} \otimes \mathbf{r_1} + \textbf{give-obj} \otimes \mathbf{r_1}.$$

The first alternative (**book⊗give-obj**) instantiates the general tensor product binding scheme **formal-filler⊗formal-role** by setting the formal role = semantic role and the formal filler = semantic filler; this method, which we'll dub the *formal=semantic approach*, is one way to use the tensor product technique to represent a proposition. Another way, illustrated in Fig. 3, is to set the formal filler to be the superposition of a semantic role and its corresponding semantic filler, and set the formal role to be an arbitrary pattern, independent of the other formal roles used in the other bindings. We'll call this the *formal≠semantic approach*. (Other connectionist representational schemes also instantiate this approach: see the Conclusion section.)

Fig. 3b shows only one of the three bindings present in Fig. 3a; the other two are analogous. Just as prescribed by the general tensor product scheme, these three bindings are combined by superposition (i.e., summation); this yields exactly Fig. 3a. In Fig. 3a, we have distinguished the three bindings by using different shading patterns for their active units; in all cases, regardless of pattern, the shaded units have activity 1.

The real resource measure of a representation, we claim, is the number of activation values it requires; here, we'll call this *tensor element complexity*, TEC for short. The TEC of the formal=semantic and formal≠semantic representations differ. For the formal=semantic approach, it's (#semantic roles)(#semantic fillers); for the other, it's N(#semantic roles + #semantic fillers). (Recall that N is the number of time slots in each cycle. We count only the activation values in a single cycle, since multiple cycles contain no more information.)

As Shastri and Ajjanagadde (1993) point out, temporal synchrony allows book to be bound to both give-obj and, say, own-obj in a single time slot; the tensor product analysis of this is simply:

$$\textbf{book} \otimes \mathbf{r_1} + \textbf{give-obj} \otimes \mathbf{r_1} + \textbf{own-obj} \otimes \mathbf{r_1}.$$

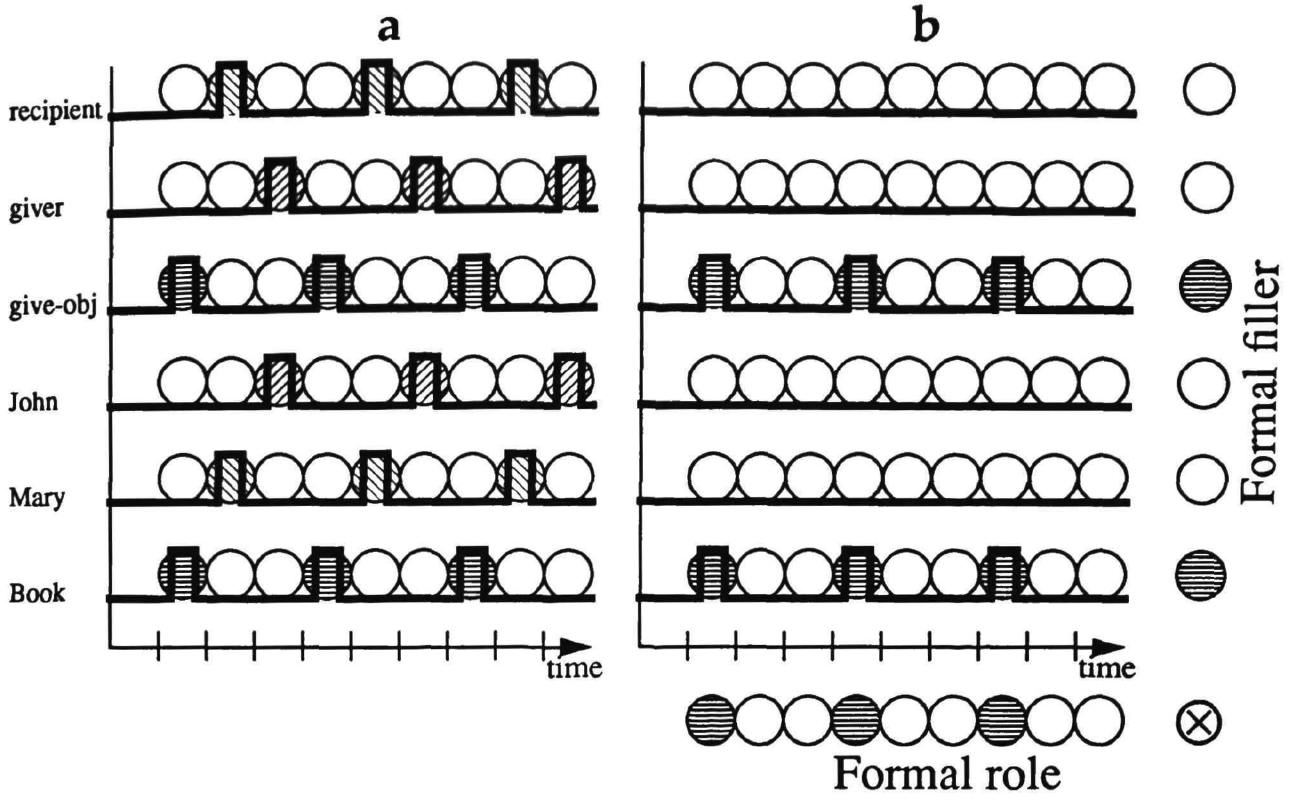

Fig. 3a: Temporal synchrony binding as a spatio-temporal tensor product representation.
Fig. 3b: One of the three bindings which are superimposed in Fig. 3a, book/give-obj, the tensor product of the indicated formal filler and formal role vectors.

A Two-Way Formal/Implementational Representation Typology

Viewing the sychronized firing model as a kind of tensor product has a number of advantages. Crucially, it allows us to clearly separate issues lying at Marr's lowest, implementational, level from those at the next highest, representational/algorithmic, level (Marr 1982).

Fig. 3 is a space-time diagram of a network. If the horizontal axis is changed from time to space, it becomes a normal diagram of a network, where each circle represents a real rather than a virtual unit. The number of activity values—the TEC—does not change, of course; we have only the standard trade-off between space and time. In the purely-spatial interpretation of Fig. 3, we have a pattern of activity distributed over two space dimensions, but unvarying in time; in the space-time interpretation, we have a temporally varying pattern of activity across a one-space-dimensional network. We can choose to expend N time units on our representation, and (#semantic roles + #semantic fillers) real units; or we can expend 1 time unit and N(#semantic roles + #semantic fillers): the resulting TEC is the same. The choice here is clearly an implementation-level one: at the representational level, Fig. 3 characterizes the same representation whether the horizontal axis is implemented in space or in time.

On the other hand, Fig. 3, illustrating the formal≠semantic approach, differs at the representational level from the formal=semantic alternative (they have different TECs, for example). Thus we can see in the temporal synchrony proposal two separate ideas residing at two different levels; these are shown along the two axes of the two-by-two typology of representations shown in Table 2 (in which 'f-' and 's-' abbreviate 'formal' and 'semantic,' respectively. The TEC depends on the representational but not the implementational axis.

A larger and more general typology can be generated by considering additional representational and implementational issues, including those listed in Table 3.

Conclusion

The temporal synchrony representational scheme of Shastri & Ajjanagadde (1993) and others provides an elegant synthesis of two logically separate ideas. The first is the implementation-level idea of using time as a representational dimension: in addition to using space as a resource for holding the activation values of a connectionist representation, we can also use time as such a resource. This is potentially quite useful for designing efficient artificial or biologically faithful networks. As our analysis shows, there is nothing inherently *temporal* about the role played by time in the synchrony approach; this role could be played by various other implementational resources, e.g., space, without changing the structure of the representations.

Table 2: A Typology of Representations

Typology		Representational Axis

Spatial Semantic Roles
time-complexity = 1
space-complexity =
(#s-roles)(#s-fillers)
	Temporal Semantic Roles
	time-complexity = #s-roles
	space-complexity =
	#s-fillers
		formal = semantic
		#f-fillers = #s-fillers
		#f-roles = #s-roles
		TEC = (#s-roles)(#s-fillers)

Spatial 'Synchrony'
time-complexity = 1
space-complexity =
N(#s-fillers + #s-roles)
	Temporal Synchrony
	time-complexity = *N*
	space-complexity =
	#s-fillers + #s-roles
		formal ≠ semantic
		#f-fillers = #s-fillers + #s-roles
		#f-roles = *N*
		TEC = *N*(#s-fillers + #s-roles)

Spatial Formal Roles	**Temporal Formal Roles**	TEC = time-complexity × space-complexity
Implementational Axis		

Table 3: Representational- *vs.* Implementational-Level Issues

Representational-Level Issues
Distinction of formal and semantic roles and fillers
Localist *vs.* distributed representations
Dimensionality of the role and filler spaces (both semantic and formal)
Continuous *vs.* discrete dimensional indexing
Range of tensor component values used (e.g., real, binary)
Separate *vs.* overlapping subspaces for semantic roles and semantic fillers
Implementational-Level Issues
Complexity in space (number of neural units)
Complexity in time (intrinsic to neural units)
Complexity of activation values of units (real *vs.* binary, complex, 'label-passing')

The second, higher-level, idea embodied in the temporal synchrony scheme is that a semantic role/filler pair like give-obj/book can be *implicitly* bound together by *explicitly* binding each of give-obj and book to a common formal role (using the tensor product). This kind of implicit binding has been used by others, e.g., Pattern Similarity Association of Barnden & Srinivas (1991). The explicit binding can be analyzed within the tensor product representational framework, like the explicit binding in virtually all other connectionist representation schemes.

As our typology of representations spells out, the second idea has a significant bearing on the abstract structure and complexity of representations, whereas the first, implementational idea, does not.

The tensor product analysis of temporal synchrony offers a number of other contributions which we have insufficient space to discuss here; for example, it renders completely straightforward the seemingly impossible generalizations of the synchrony technique to distributed representations and to overlapping firing patterns, and it allows direct formal analysis of inference over these representations, exploiting tensor calculus (Tesar & Smolensky, in preparation).

Acknowledgements

The first author is supported by an NSF Graduate Fellowship and both authors by NSF grant IRI-9213894. We thank Lokendra Shastri and John Hummel for stimulating conversations, and the Society reviewers for helpful comments.

References

Barnden, J. A. and Srinivas, K. 1991. Encoding techniques for complex information structures in connectionist systems. *Connection Science* **3**, 263–309.

Eckhorn, R., Reitboeck, H.J., Arndt, M., and Dicke, P. 1990. Feature linking via synchronization among distributed assemblies: Simulations of results from cat visual cortex. *Neural Computation* **2**, 293–307.

Gray, C.M., Konig, P., Engel, A.K., and Singer, W. 1989. Oscillatory responses in cat visual cortex exhibit intercolumnar synchronization which reflects global stimulus properties. *Nature* (London) **338**, 334–7.

Halford, G.S., Wilson, W.H., Guo, J., Wiles, J., and Stewart, J.E.M. 1994. Connectionist implications for processing capacity limitations in analogies. In K.J. Holyoak & J. Barnden, eds., *Analogical Connections*. Norwood, NJ.: Ablex.

Hummel, J.E., and Biederman, I. 1992. Dynamic binding in a neural network for shape recognition. *Psychological Review* **99**, 480–517.

Humphreys, M.S., Bain, J.D., and Pike, R. 1989. Different ways to cue a coherent memory system: A theory of episodic, semantic and procedural tasks. *Psych. Review* **96**, 208–33.

Legendre, G., Miyata, Y., and Smolensky, P. 1990. Harmonic Grammar—A formal multi-level connectionist theory of linguistic well-formedness: Theoretical foundations. *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, 884–91. Cambridge, MA. LEA.

Marr, D. 1982. *Vision*. San Francisco: Freeman.

Shastri, L., and Ajjanagadde, V. 1993. From simple associations to systematic reasoning: A connectionist representation of rules, variables, and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences* **16**, 417–94.

Smolensky, P. 1990. Tensor Product Variable Binding and the Representation of Symbolic Structures in Connectionist Systems. *Artificial Intelligence* **46**, 159–216.

Tesar, B. & Smolensky, P. In preparation. Synchronous firing variable binding and inference: A tensor product analysis.

von der Malsburg, C., and Schneider, W. 1986. A neural cocktail-party processor. *Biological Cybernetics* **54**, 29–40.