Mapping Hierarchical Structures with Synchrony for Binding: Preliminary Investigations

John E. Hummel Eric R. Melz Jeff Thompson Keith J. Holyoak
Department of Psychology
University of California, Los Angeles
Los Angeles, CA 90024-1563
jhummel@psych.ucla.edu, melz@isi.edu, thompson@psych.ucla.edu, holyoak@psych.ucla.edu

Abstract

Synchrony of firing has recently become a popular technique for dynamic binding in neural networks, and has been applied to numerous problem domains. However, hierarchical structures are difficult to represent using synchrony for binding. This paper presents our progress toward a framework for representing hierarchies in a neural network using synchrony for dynamic binding. We illustrate the approach with a model of analogical mapping. The model (IMM2) uses synchrony to bind case roles to objects within propositions. Hierarchies are established by allowing units representing propositions to play a dual role, acting both as the argument of one proposition and as a pointer to another.

Introduction

Synchrony of firing has attracted attention as a useful way to create dynamic bindings in artificial neural networks. The idea is that a binding of two or more primitives can be represented by allowing separate units, each representing one primitive, to fire in synchrony. For example, to represent the proposition *own(Janet Book)*, units representing *own-agent* would fire in synchrony with units for *Janet* while units for *own-object* fire in synchrony with units for *book* (Hummel & Holyoak, 1992; Shastri & Ajjanagadde, 1990). Dynamic bindng is attractive because it can be created and destroyed on the fly, allowing a network to reuse the same units in multiple bindings. This capacity has many important implications, the most basic of which is that it makes compositionality possible in a neural network representation.

Despite the usefulness of synchrony for simple dynamic binding, it is not straightforward to extend it to represent hierarchical structures. Given the ubiquity of hierarchies in human cognition, the problem of how they may be represented in a general way is an extremely important one. This paper focuses on the use of synchrony to represent and process hierarchical structures.

The task domain within which this issue will be addressed is *analogical mapping*, the problem of finding a set of correspondences between the elements of two analogous situations or structures (a *source* analog and a *target*). For example, given the source *know(Clark Janet)*, and the target *know(Paul Elaine)*, a natural mapping places *Clark* into correspondence with *Paul* and *Janet* into correspondence with *Elaine*. Analogical mapping provides an ideal problem domain for assessing representations of structure because structural constraints, based in part on hierarchical relations, play a central role in establishing analogical mappings (Gentner, 1983; Holyoak & Thagard, 1989). In addition, mapping is a challenging special case of the basic cognitive process of comparing structured representations. Comparing representations (e.g., matching cases to schemas, or memory elements to antecedents of rules) is fundamental to reasoning and comprehension. Analogical mapping is particularly demanding because it requires finding novel constant-to-constant correspondences, which cannot be pre-wired in any simple fashion. In general, comparison of structured representations has proven difficult to accomplish in neural networks (Barnden, 1994).

Hummel and Holyoak (1992; Hummel, Burns & Holyoak, 1994) have proposed a model of analogical mapping that uses synchrony to dynamically bind case roles to objects in propositional statements and to establish correspondences between the elements of two analogs. This model, called the Indirect Mapping Model (IMM), is the point of departure for the current effort. IMM can solve simple analogies of the type illustrated above, as well as considerably larger analogies. However, like other models using synchrony for binding, it cannot represent or compare hierarchical structures, such as propositions that take other propositions as arguments.

Hierarchies and Synchrony

Hierarchies are difficult to represent using synchrony because synchrony provides only one degree of freedom (df) for binding: at any given instant, two units are either synchronized or they are not (but see McClurkin et. al., 1988). When units are synchronized, the entities they represent are interpreted as bound (members of a single group); conversely, if two entities are unbound (members of different groups), then their respective units must remain desynchronized. Accordingly, simple synchrony relations cannot represent that two units are members of the same group at one level of a hierarchy and members of separate groups at some lower level of the hierarchy. For example, in the proposition *know(Clark own(Janet book))*, *Janet* and *book* are members of the same group with respect to the patient role of the predicate *know*, so with respect to the top-level proposition they should be synchronized. But in the lower-level proposition, *Janet* and *book* are bound to different case roles and should therefore remain out of synchrony.

In principle, this dilemma could be resolved by any number of schemes for squeezing extra df out of the temporal characteristics of a unit's firing. For example, units might fire in synchrony once if they are bound at the first level of a hierarchy, twice if they are bound at both the first and second, etc. However, such schemes are likely to be cumbersome and error prone (e.g., what temporal resolution would be required to exploit such a code?). Moreover, because they rely on complex codes, such schemes are prone to hard and arbitrary limits on embedding (determined by the number of codes in the scheme).

We propose an approach to representing hierarchies that operates within synchrony's single df, by allowing only one level of a hierarchy to be active at a time. Hierarchical embedding is accomplished by means of units that represent complete propositions (*proposition units*). A proposition unit can serve both as the argument of one proposition and as a pointer to another (in general, such units could point to any complex structure, but the current discussion is restricted to the representation of propositions). Because a single unit cannot itself "contain" all the semantic content of a proposition, it must instead be able to stand for that content in some other meaningful way. The manner in which proposition units perform this function becomes the central issue in this approach. In general terms, pointing is accomplished by having a proposition unit learn connections that allow it to mimic the causal properties of the propositional structure to which it points (as elaborated shortly).

The Indirect Mapping Model, II (IMM2)

Architecture

IMM2 encodes propositions as arrangements of five classes of units. Units are coupled by one or both of two types of links: *connections*, which propagate excitation and inhibition, and *Fast Enabling Links* (*FELs*; Hummel & Biederman, 1992), which allow units to synchronize and desynchronize oscillations in their outputs. The encoding of *own(Janet book)* is illustrated in Figure 1. Each proposition, *i*, is encoded locally by one proposition unit, P_i. P_i shares recriprocal excitatory connections with one to three *sub-proposition* units, $SP_{i.j}$. $SP_{i.j}$ encodes the bindings defining the *j*th case role of the *i*th proposition. For example, if P_1 is *own(Janet book)*, then $SP_{1.1}$ encodes the binding of *Janet* to the agent role of *own*, and $SP_{1.2}$ encodes the binding of *book* to the object role of *own*. SPs under the same proposition share positive connections and negative FELs (the latter keep the SPs out of synchrony with one another). $SP_{i.j}$ encodes its case role-argument bindings by means of reciprocal excitatory connections and positive FELs with one *predicate* unit and one *object* unit. In the current example, $SP_{1.1}$ is linked to the predicate unit for *own-agent* (denoted O1) and the object unit for *Janet* (denoted J). Predicate and object units are functionally equivalent, and are distinguished only for clarity.

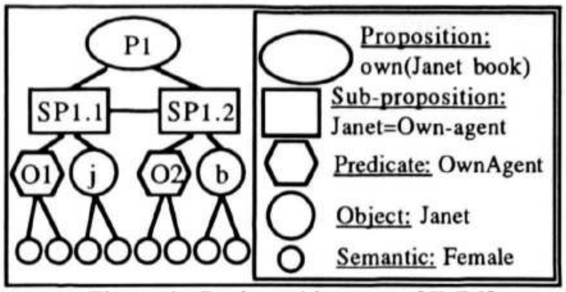

Figure 1. Basic architecture of IMM2.

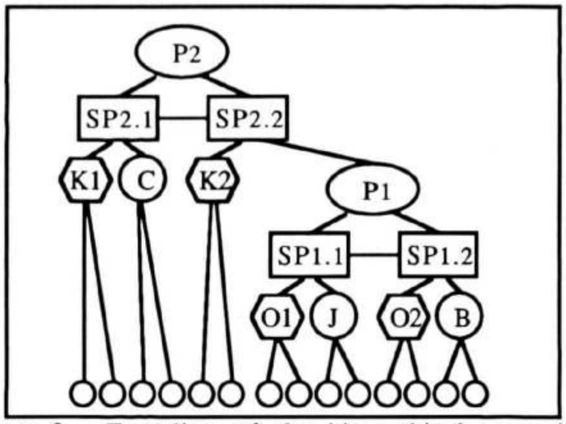

Figure 2. Encoding of the hierarchical proposition *know(Clark* P1*)*, where P1 = *own(Janet book)*.

Proposition, sub-proposition, predicate, and object units serve the purely structural function of encoding the binding relations among the components of a proposition. This function is enhanced by their strictly localist implmentation (see Hummel & Holyoak, 1992). However, encoding semantics is better served by a distributed representation over units representing attributes. Each object and predicate unit is connected to a set of *semantic* units that code the attributes of the corresponding object or predicate. For example, *Janet* might be connected to semantic units for *human*, *adult*, and *female*, while *book* might be connected to *artifact*, *paper*, and *small*. Similarly, predicates, such as *own-agent* and *own-object* are connected to semantic units for their attributes. Semantic units for predicates are distinguished according to place, so that units encoding the agent role do not overlap with those for the patient role. Otherwise, all predicate and object representations are allowed to overlap on the semantic units.

Extending this architecture to encode hierarchical propositions is straightforward. Let P_2 be *know(Clark own(Janet book))*. P_2 is encoded as shown in Figure 2, where P_1 simply takes the place of an object unit under $SP_{2.2}$ (which encodes the binding P_1 to *know-object*). Otherwise, the general structure encoding P_2 is exactly the same as that encoding P_1.

For the purposes of analogical mapping, propositions are encoded in two mutually exclusive sets, propositions P^s in the source analog, and P^t in the target. Source and target propositions share semantic units (see Figure 3), and it is primarily through the semantic units that propositions in the source activate propositions in the target. In addition to the connections described above, there are modifiable connections between units of the same type across the analogs (henceforth, *Hebb* connections, for their Hebbian learning rule). In the initial state of the network, the weights of all Hebb connections are zero. As the network runs, these connections update their weights to (a) keep a record of the mappings the network has established, and (b) allow past mappings to constrain future mappings.

Operation

In general, IMM2 performs analogical mapping in the same way as the original IMM. One at a time, propositions in the source become active and create patterns of activity on the semantic units. The semantic units excite and synchronize units in the target analog. Lateral inhibition between units of the same class (i.e., predicate, object, SP, and proposition) and recurrent excitation between consistent units (e.g., SPs with their objects and predicates) results in the best-fitting target proposition's growing active at the expense of poorly-fitting propositions. Once the pattern of activation in the target settles, active units strengthen their Hebb connections from the active units (of the same class) in the source. The pattern of Hebb connections that evolves over several such cycles is the model's representation of the best source-target mapping.

Active propositions. Active propositions are represented as synchronized patterns of activity over the network's units. Proposition i is recalled from long-term memory by activating unit P_i. P_i excites all SP units under it. The SPs establish the pattern of synchronized firing that represents the active proposition. By virtue of the negative FELs between them, SPs under a given P_i rapidly establish a pattern of mutually desynchronized firing. Each SP excites and synchronizes the predicate and object (or proposition) under it. Therefore, predicates and objects belonging to the same case role fire in synchrony with one another and out of synchrony with those belonging to other case roles. The predicates and objects likewise synchronize the semantic units to which they are connected. Consider an example. Say P_1 in Figure 2, *own(Janet book)*, becomes active. It will activate $SP_{1.1}$ and $SP_{1.2}$, which will begin to fire out of synchrony with one another. $SP_{1.1}$ will activate and synchronize the predicate *own-agent* and the object *Janet*, and these will activate and synchronize their respective semantic units. $SP_{1.2}$ will similarly activate *own-object* and *book*, which will activate their respective semantic units. The result is that the semantic units for *Janet* and *own-agent* will fire in synchrony with one another and out of synchrony with those for *book* and *own-object*.

The above describes the activation of simple, non-hierarchical propositions. More challenging is the representation of active hierarchical propositions. Due to the one df limitation on synchrony, it is critical that semantic units representing a lower proposition (such as P_1 in Figure 2) not become active when the higher level proposition (P_2) is active. However, it is necessary to activate some representation of the lower proposition. Otherwise, in the case of, say, *know(Clark own(Janet book))*, there would be no explicit representation of what it is that Clark knows. IMM2 solves this problem by allowing the lower level proposition unit (P_1) to become active but not to propagate its activation to the SPs below itself ($SP_{1.1}$ and $SP_{1.2}$). This gating is accomplished by means of a "mode"[1] within which each P_i operates. If P_i receives more excitatory input from SPs above itself (i.e., SPs with respect to which it serves as an object filling a role) than from SPs below itself, then it enters "child" mode. If it receives more excitation from SPs below than above, then it enters "parent" mode. In parent mode, a P_i propagates activation only to SPs below itself; in child mode, it propagates activation only to SPs above itself.

Consider what happens when P_2, *know(Clark* P_1*)*, becomes active. It will activate $SP_{2.1}$ and $SP_{2.2}$, which will fire out of synchrony with one another. $SP_{2.1}$ will activate and synchronize *know-agent* and *Clark*, which will activate and synchronize their respective semantic units. Similarly, $SP_{2.2}$ will activate *know-object*, which will activate its semantic units. But when $SP_{2.2}$ activates P_1, P_1 will enter child mode, firing in synchrony with $SP_{2.2}$ (and, therefore, with *know-object*), but not propagating any activity to $SP_{1.1}$ or $SP_{1.2}$. The resulting pattern thus nominally represents the embedding of one proposition within another. For this nominal representation to be a functional one, P_1 must function in lieu of the entire proposition to which it points. P_1 learns to perform this function during the analogical mapping process.

Analogical mapping. Like the original IMM, IMM2 performs analogical mapping as a form of guided pattern matching. One at a time, P_i in the source are selected (at random) to become active. As each is activated, it creates a synchronized pattern of activation on the semantic units as described above. In turn, these patterns activate and synchronize predicate, object, SP, and proposition units in the target analog. In the target, units of the same type compete via lateral inhibition, and those that remain active update their Hebb connections to the active units in the source. After several iterations, the Hebb connections grow toward asymptotic values. For example, run on the analogy in Figure 3, the connection from P^s_1 in the source (henceforth *S1*) to P^t_1 in the target (T1) grows toward 1.0, and those from S1 to T2 and from S2 to T1 grow toward -1.

Hebb connections allow earlier mappings to constrain later mappings. For example, if S1 is coactive with T1, then the connection between them will become positive. Therefore, whenever S1 becomes active, it will pass activation directly to T1, favoring it in its competition with the other target propositions. Similar biasing results from the Hebb connections that develop between predicate, object and SP units. In every case, the biasing serves to constrain the later mappings on the basis of earlier mappings. But in the case of the proposition units, the biasing serves the additional function of allowing embedded propositions to behave as hierarchical structures.

---

[1] Although they might appear "non-neural," such modes are straightforward to implement with two auxillary units and multiplicative synapses.

Consider the following simple analogy (*Example 1*):

<u>Source</u>	<u>Target</u>
S1: F(a b)	T1: F(f g)
S2: G(c S1)	T2: G(h T1)
	T3: F(i j)

The representation of Example 1 is illustrated in Figure 3. Upper case letters are predicate units (distinguished according to place) and lower case are objects. The desired solution maps S1 to T1, S2 to T2, and nothing to T3, along with the corresponding predicate and object mappings. This example is semantically impoverished (except that identical letters represent identical predicates), and therefore is solvable only on the basis of its structure. If the hierarchical structure of the analogs is ignored, then there is no basis for determining whether S1 should map to T1 or T3. However, if hierarchial structure is considered, then S1 maps to T1 because they are embedded within S2 and T2, respectively.

How does IMM2 solve this mapping? Suppose S2 is the first proposition to become active. It will switch into parent mode and activate its SPs. They will desynchronize and activate G1 (the agent role of the predicate *G*) and c (synchronized) and G2 and S1 (synchronized). S1 will enter child mode because it is receiving excitation from the SP above itself. G1 and G2 in the source will activate G1 and G2 in the target (via their shared semantic units), and G1 and G2 in the target will excite T2. T2 will (a) enter parent mode, because it is receiving excitation from below itself, and (b) activate T1, which will enter child mode. All Hebb connections initially have strengths of zero. Active units update their Hebb connections. Proposition units strengthen their Hebb connections only with other proposition units in the same mode. Thus, the connection from S1 to T1 will grow stronger, the connection from S2 to T2 will grow stronger, and all the other Hebb connections involving those units will grow proportionately weaker (the learning rule is competitive).

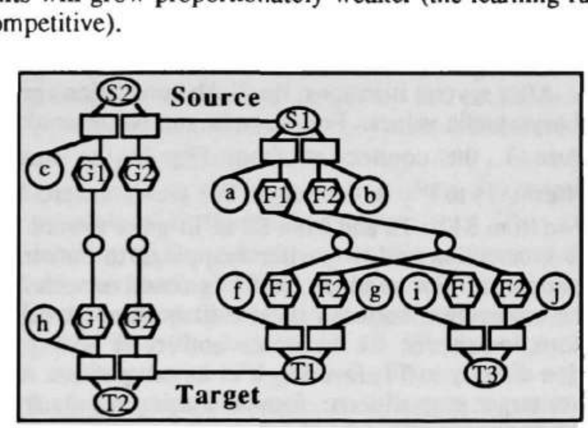

Figure 3. The IMM2 representation of Example 1.

Next suppose that S1 becomes active. It will activate the predicates F1 and F2, which (via the semantic units and the target's predicate units) will excite both T1 and T3, causing them to enter parent mode. On the basis of their inputs from the semantic units, T1 and T3 are equally good matches to S1. But by virtue of the Hebb connection from S1 to T1 (established last iteration), T1 will have an advantage over T3 in the inhibitory competition. It will therefore achieve a higher level of activation and further strengthen its Hebb connection from S1. After several iterations, the Hebb connection from S1 to T1 will approach 1.0 and the other Hebb connections involving those units will grow negative. The model will have successfully mapped the two analogs.

This illustration necessarily glossed over many details of the model's operation. In particular, it does not make the role of synchrony obvious. Synchrony represents case role-object bindings, so it becomes particularly important when a mapping depends upon these bindings. The original IMM demonstrates how synchrony can be used to map such analogies, so we shall not illustrate that aspect of IMM2 here.

Details of Operation

Establishing Synchrony: Every unit in IMM2 is described by three primary state variables: Activation (a_i), phase (ϕ_i), and output (o_i). Each unit's phase varies cyclically between 0 and 2π. Together, a unit's phase and activation determine when it will fire (set o_i to 1):

$$o_i = \begin{cases} 1, \text{ if } 0 < \phi_i < a_i\pi / 2 \\ 0, \text{ otherwise.} \end{cases} \tag{1}$$

The greater a_i, the longer the interval $0...a_i\pi/2$, and so the greater the number of times i will fire during the cycle; IMM2 codes activation as firing frequency.

Units synchronize their outputs by locking their phases, and desynchronize their outputs by keeping their phases as far apart as possible (the maximum difference between phases is 180°):

$$\Delta\phi_{ij} = \kappa e_{ij} f(\phi_i, \phi_j, FEL_{ij}), \tag{2}$$

where $\Delta\phi_{ij}$ is the change in ϕ_i due to ϕ_j, FEL_{ij} is the weight on the FEL from unit j to unit i, κ is a scaling constant ($0<\kappa<0.5$), e_{ij} is a random binary variable ($p(e_{ij}=1) = a_j|FEL_{ij}|$), and f is a function that returns the angular distance, α_{ij}, between ϕ_i and ϕ_j. For $FEL_{ij} > 0$, $f(\phi_i,\phi_j;FEL_{ij}) = -\alpha_{ij}$, and and for $FEL_{ij} < 0$, $f(\phi_i,\phi_j;FEL_{ij}) = -\alpha_{ij} + \pi$. Thus, positive FELs encourage units to have similar phases (and therefore to fire in synchrony), while negative FELs encourage units to have phases 180° apart (and therefore to fire out of synchrony). Unit i computes the total change in its phase, $\Delta\phi_i$, as the mean over j of all $\Delta\phi_{ij}$ (plus a constant phase increment, which causes the units' phases to change cyclically over time). The FEL-based updating of phase occurs on a faster time scale than the cyclical incrementing.

Flow of Activation: Propositions, SPs, predicates and objects compute their net inputs, n_i, by:

$$n_i = \sum_j o_j w_{ij} m_{ij}, \tag{3}$$

where o_j is the output of unit j, and w_{ij} is the connection weight (Hebb or fixed) from j to i. m_{ij} is 0 for proposition units in different modes, and 1 for all other units. All lateral inhibitory weights are set to -0.98. Units update their activations by:

$$\Delta a_i = \begin{cases} \gamma n_i(1-a_i) - \delta a_i, n_i > 0 \\ n_i a_i, n_i < 0, \end{cases} \tag{4}$$

where γ and δ are growth and decay parameters, respectively. Target units update their Hebb connections from source units by:

$$\Delta w_{ij} = \mu(1-w_{ij})o_i o_j. \tag{5}$$

μ is a learning rate. Learning is competitive: each time w_{ij} is updated, the quantity $\Delta w_{ij}/n$ is subtracted from the n other connections from source unit i and the n other connections to target unit j. As a result, $\Sigma_j w_{ij}$ and $\Sigma_i w_{ij}$ are always 0.

Simulations

As a first test, we ran IMM2 on Example 1 (Figure 3) for two complete cycles. Each source proposition was chosen to be active once during each cycle and was allowed to remain active for 600 time slices, during which the model ran as described above. The mappings it established (in terms of the final values of the Hebb weights) conformed exactly to the desired mappings. IMM2 mapped S1 to T1 (the desired mapping) with strength 1.00 (the maximum value of Hebb weight), both S1 and S2 to T3 with strength -0.5, and S1 to T2 and S2 to T1 with strength -1.00. Likewise, all corresponding sub-proposition, predicate and object units mapped with weights greater than 0.95, and all non-corresponding predicates and objects mapped with weights less than zero. Although this analogy is extremely simple, it demonstrates that IMM2 can map propositions based on their hierarchical structure alone.

To test IMM2's ability to cope with more deeply nested hierarchies, we next tested it with a variant of Example 1 with one additional level of embedding. This analogy (Example 2) is:

Source	Target
S1: F(a b)	T1: F(f g)
S2: G(c S1)	T2: G(h T1)
S3: H(w S2)	T3: H(v T2)
	T4: F(d e)
	T5: G(k T4)

Here, the correct mapping maps S1 to T1, S2 to T2, and S3 to T3, but this time, the model must respect two levels of hierarchical structure to get the correct mappings: On the basis of predicates alone, S1 is as good a match to T4 as to T1; and on the basis of predicates and one level of hierarchical embedding, S3 is as good a match to T5 as it is to T3. Successfully mapping S3 to T3 (rather than to T5) thus requires the model to honor hierarchical embedding at two levels. The model successfully mapped these analogs: After three cycles, all Hebb connections corresponding to correct mappings had values greater than 0.75, and all those corresponding to incorrect mappings had values less than zero.

IMM2 also successfully mapped all the non-hierarchical analogies with which we tested the original IMM (see Hummel & Holyoak, 1992; Hummel et. al., 1994).

The most substantial test of IMM2 to date is based on a derivative of Gentner and Toupin's (1986) "Jealous Animal" story. The basic form of this story states that a cat was friends with a walrus, which played with a seagull, and made the cat jealous, causing the cat to become angry. Its anger caused the cat to become reckless, which caused the cat to be in danger. The seagull saved the cat, which caused the cat to befriend the seagull. Gentner and Toupin tested children's comprehension of stories like this by having them act them out with different characters. The finding of primary interest here is that comprehension was better for *systematic* stories, which included higher-level statements about the causal relationships between events, than for *unsystematic* stories, which did not include such statements (the magnitude of this effect varied as a function of a number of other variables).

Following Holyoak and Thagard (1989), we adapted the jealous animal story for IMM2 by generating source and target analogs from it, and testing the model's ability to map them onto one another as a function of whether the analogs were systematic or unsystematic (i.e., whether they contained the higher-level propositions). If IMM2 can use hierarchical statements (e.g., *cause(A B)*) to constrain analogical mapping, then mapping should be faster and/or more accurate with systematic than with unsystematic analogies. The source and target analogs for the systematic version of this analogy each contain 13 propositions, and are too large to display here. The target story was the same as the source except that the cat, walrus and seagull were replaced with a dog, walrus, and penguin, respectively. Each object was represented by six semantic units, of which similar objects in the source and target shared five. The model was allowed to run until the Hebb weights stabilized.

IMM2's performance on these simulations was mixed. Although it succeeded in finding intuitively correct mappings in the structured case, it also succeeded in doing so in the unstructured case. In one representative unstructured run, it found all the correct object and proposition mappings (and no incorrect mappings) within six cycles through the source (Hebb weights corresponding to correct mappings were greater than 0.9, and those corresponding to incorrect mappings were less than zero). On an otherwise equivalent run with the structured version of the analogy, the model settled on the correct mappings (with no incorrect mappings) after a single cycle through the source. Thus, IMM2 is faster to settle on the correct mappings in the structured than the unstructured cases, but it eventually succeeds in both cases. Although these results show that IMM2 is capable of finding the correct mappings in a complex analogy, they only show a modest advantage for structured over unstructured versions of the analogy.

Discussion

We have yet to fully explore IMM2's properties, but preliminary results suggest that it can represent and compare simple hierarchies. Central to this capacity is the proposition unit's ability to act both as an object filling a role in one proposition and as a pointer to another. With this extension, the one df synchrony provides for binding suffices to capture structural relations within multi-level propositions. IMM2 exploits both local and distributed representations and both serial and parallel processing. The units that encode structural relations are strictly localist, but the meanings of individual concepts are distributed over multiple semantic units. During mapping, source propositions are activated serially, and at a finer time scale the firing of elements associated with distinct roles are desynchronized, hence serial. This serial processing is crucial in representing the bindings of objects to roles. At the same time, target propositions respond in parallel to the activation triggered by the firing of a source proposition. The integrated system provides distributed representations of meaning and decision-making by parallel constraint satisfaction while maintaining systematicity of knowledge (Fodor & Pylyshyn, 1988).

Acknowledgments

Supported by NSF Grant SBR-9310614 to K.H.

References

Barnden, J.A. (1994). On the connectionist implementation of analogy and working memory matching. In J. A. Barnden & K. J. Holyoak (Eds.), *Advances in connectionist and neural computation theory, Vol. 3: Analog, metaphor, and reminding.* (pp. 327-374). Norwood, NJ: Ablex.

Fodor, J.A., & Pylyshyn, Z.W. (1988). Connectionism and cognitive architecture: A critical analysis. In S. Pinker & J. Mehler (Eds.), *Connections and symbols* (pp. 3-71). Cambridge, MA: MIT Press.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science, 7,* 155-170.

Gentner, D., & Toupin, C. (1986). Systematicity and surface similarity in the development of analogy. *Cognitive Science, 10,* 277-300.

Holyoak, K. J., & Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science, 13,* 295-355.

Hummel, J. E., Burns, B., & Holyoak, K. J. (1994). Analogical mapping by dynamic binding: Preliminary investigations. In K. J. Holyoak & J. A. Barnden (Eds.), *Advances in connectionist and neural computation theory, Vol. 2: Analogical connections.* (pp. 416-445). Norwood, NJ: Ablex.

Hummel, J. E., & Holyoak, K. J. (1992). Indirect analogical mapping. *Proceedings of the 14th Annual Conference of the Cognitive Science Society.* (pp 516 - 521). Hillsdale, NJ: Erlbaum.

McClurkin, J., Gawne, B., Richmond, B., Optican, L. & Robinson, D. (1988). Lateral geniculate nucleus neurons in awake behaving primates: I. Responses to B&W 2-D patterns. *Abstracts of the Society for Neuroscience, 14,* 309.

Shastri, L. & Ajjanagadde, V. (1990). From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings. Technical report MS-CIS-90-05, Dept of Computer and Information Sciences, University of Pennsylvania.

Thagard, P., Holyoak, K. J., Nelson, G., & Gochfeld, D. (1990). Analog retrieval by constraint satisfaction. *Artificial Intelligence, 46,* 259-310.