Indirect Analogical Mapping

John E. Hummel and Keith J. Holyoak
Department of Psychology
University of California, Los Angeles, CA 90024-1563

Abstract[1]

An Indirect Analogical Mapping Model (IMM) is proposed and preliminary tests are described. Most extant models of analogical mapping enumerate explicit units to represent all possible correspondences between elements in the source and target analogs. IMM is designed to conform to more reasonable assumptions about the representation of propositions in human memory. It computes analogical mappings indirectly -- as a form of guided retrieval -- and without the use of explicit mapping units. IMM's behavior is shown to meet each of Holyoak and Thagard's (1989) computational constraints on analogical mapping. For their constraint of pragmatic centrality, IMM yields more intuitive mappings than does Holyoak and Thagard's model.

Introduction

The central function of analogical thinking is to aid in creating coherent, structured representations of important novel situations. By finding a mapping -- that is, a set of correspondences -- between a known situation (the *source* analog) and a novel one (the *target* analog), the structure of the source can be used as a kind of blueprint for building a representation of the target. While analogy involves a number of component processes, the mapping process is pivotal because the correspondences it establishes constrain the inferences that can be generated about the target.

This paper presents our preliminary investigations into an Indirect Analogical Mapping Model (IMM). The primary goal of this effort is to develop an algorithm for analogical mapping consistent with reasonable assumptions about the representation of propositional information in human memory. Extant models of analogical mapping typically posit explicit processing units for all possible correspondences between the elements of the source and target analogs (e.g., Falkenhainer, Forbus & Gentner, 1989; Holyoak & Thagard, 1989). There are a number of serious problems associated with such enumeration of mapping units (Hofstadter & Mitchell, in press). Although it can be argued that mapping units are a notational convenience rather than a literal claim about the nature of mental representations, it is unclear how the critical processes posited by such models (e.g., parallel constraint satisfaction) would operate under more natural representational assumptions. A related difficulty with explicit mapping units is that they exist strictly for the purpose of analogical mapping and have no obvious usefulness for other cognitive processes. The primary goal of IMM is to simulate analogical mapping within an architecture more consistent with realistic assumptions about the representation of propositions in memory.

Theoretical Motivation

Representation of Propositions. The central problem in representing propositions involves encoding their internal structure. Representing a proposition entails creating a set of bindings between the arguments of the proposition and the case roles they fill. For example, to distinguish the representation of (chase Arnold Bill) from (chase Bill Arnold), Arnold must be bound to the agent role of "chase" in the first proposition and to the patient role in the second. A basic tenet of our approach is that active representation of propositional information (i.e., in working memory) and its long-term storage require different solutions to this binding problem.

Let us first consider the problem of case role-argument binding in an active representation. It is possible to imagine a representation for propositions in which dedicated units (or patterns of activation) represent each role-argument binding. For example, units could be created *de novo* each time a proposition enters working memory, or -- as proposed by Smolensky (1990) -- bindings could be represented by explicitly calculating a tensor product of the activation vectors representing the individual case roles and arguments (Halford, Wilson, Guo, Gayler, Wiles & Stewart, in press). In both these cases, the bindings are *static* because they are represented by units dedicated to specific conjunctions of elements.

This approach to the representation of attribute conjunctions suffers numerous limitations (cf. Hummel

[1]This research was supported by Contract MDA 903-89-K-0179 from the Army Research Institute, and by NSF Grant DIR-9024251 to the UCLA Cognitive Science Research Program.

& Biederman, 1992). The most serious is that by coding conjunctions of case roles and arguments, static binding units cannot represent the individual case roles (predicates) and arguments (objects); hence, the natural similarity structure of the predicates and objects is lost. For example, if separate static units represent (a) Arnold as the agent of chasing, (b) Arnold as the patient of chasing, (c) Bill as the agent, and (d) Bill as the patient, then the proposition (chase Arnold Bill), represented by units a and c, would be no more similar to (chase Bill Arnold), represented by b and d, than it is to (says My-doctor Caffeine-makes-me-nervous). And although this example assumed a localist representation, the underlying problem cannot be solved simply by postulating a more distributed representation. In Smolensky's (1987) tensor product representation, which uses distributed representations, the representation of a given object bound to one case role will not necessarily overlap *at all* with the representation of the identical object bound to a different case role.

An alternative to static binding is *dynamic binding*, in which units representing case roles are *temporarily* bound to units representing the arguments of those roles. Following Shastri and Ajjanagadde (1990) and others, IMM represents dynamic case role-argument bindings as synchronized firing of units representing the bound elements. For example, (chase Arnold Bill) is represented by units for the agent role of "chase" firing in synchrony with units for Arnold, while units for the patient role of "chase" fire in synchrony with units for Bill. Naturally, the agent/Arnold set must fire out of synchrony with the patient/Bill set.

Dynamic binding permits a small set of units to be reused in an unlimited number of specific bindings. The capacity to reuse units allows the representation of case roles and objects to be completely independent of one another. The theoretical and practical advantages of this independence are vast, but the most important is that it preserves similarity across different bindings. For example, all propositions in which Bill serves as an argument will be similar by virtue of their sharing the units that represent Bill; likewise, all propositions involving the predicate "chase" will employ the same "chase" units. As such, the independence afforded by dynamic binding permits essentially complete isomorphism between the meaning of a proposition and its representation: the representation of two propositions will overlap exactly to the extent that their meanings overlap. Some practical advantages of this isomorphism will become clear when IMM's operation is described.

Although dynamic binding affords critical benefits in the active representation of propositions, it is of course completely impractical as a solution to the storage of role-argument bindings in long-term memory. In long-term memory, bindings must be represented in a static form (e.g., as "synaptic" strengths) that can remain dormant until the proposition is reactivated. Importantly, the long-term representation must be capable of reinstating the original dynamic bindings of arguments to case roles when it is reactivated[2]. To this end, IMM encodes propositions into its long-term memory as connections from units representing objects and predicates to semantically empty units called ***sub-proposition*** (SP) units. A proposition is retrieved from long-term memory by activating the SP units that encode it. When an SP unit fires, it activates and synchronizes the object and predicate units to which it is connected. Separate SPs within a proposition remain out of synchrony (desynchronized) with one another. Together, a proposition's SPs reconstruct the synchronized firing of predicate and object units that represents the structured semantic content of the proposition.

Computational Constraints on Mapping. The computational theory underlying IMM as a model of analogical mapping is borrowed from Holyoak and Thagard's (1989) Analogical Constraint Mapping Engine (ACME). ACME posits three broad classes of constraints on natural correspondences between the elements of analogs. (1) The structural constraint of *isomorphism* has two components: (a) *structural consistency* implies that if a particular source and target element correspond in one context, they should do so in all others; (b) *one-to-one mapping* implies that each element should have a unique correspondent in the other analog. (2) *Semantic similarity* implies that elements with some prior semantic similarity (e.g., by virtue of joint membership in a taxonomic category) should tend to map to each other. (3) *Pragmatic centrality* implies that a mapping should give preference to elements that are deemed especially important to goal attainment, and should maintain correspondences that can be presumed on the basis of prior knowledge.

The Indirect Mapping Model

ACME implements the mapping constraints directly, via parallel constraint satisfaction on explicit mapping units of the type described previously. Our goal is to achieve analogical mapping according to these constraints, but to do so indirectly -- i.e., without directly implementing the constraints as connections among mapping units. Rather, IMM treats analogical mapping as a form of guided retrieval: propositions in a source analog drive the activation of propositions in a target analog. This process is mediated by a set of predicate units that are shared by the propositions in

[2]This is true by definition; any long-term representation that could not reproduce the active representation of a binding would not encode that binding in any meaningful way.

both analogs. When a proposition in the source becomes active, its SPs create a synchronized pattern of firing across the predicate units. This pattern then activates the proposition(s) in the target analog to which it most closely matches. The resulting match (i.e., mapping) is then learned by updating modifiable connections between units across the analogs. The asymptotic strengths of these connections are interpreted as the model's preferred mappings.

The implementation described here was designed to test IMM's basic capacity for this type of mapping. To unconfound the properties of the architecture from the properties of any specific units of which it might be composed, we have made a number of strong simplifying assumptions that idealize IMM's operation. These assumptions will be relaxed in future implementations.

Architecture

Figure 1 illustrates IMM's basic architecture using the following analogs:

<u>Source</u>	<u>Target</u>
(chase Arnold Bill)	(eat fox goose)
(chase Bill Charles)	(eat goose corn)

IMM is composed of three types of units: predicate units, object units, and sub-proposition (SP) units. Predicate units represent the semantic content of predicates in a distributed fashion. For example, the predicate "chase" is represented by one pattern of activity over these units, and the predicate "pursue" would be represented by a different but overlapping pattern. (These patterns are not detailed in the figure.) Similarly, objects such as Arnold and Bill share some predicates (e.g., both are human and male) and differ on others. The similarity between two objects or two predicates is defined by their degree of overlap on the predicate units. The precise content of these representations is less important for our current purposes than the architecture in which they reside.

Propositions are encoded into IMM's long-term memory by symmetrical, excitatory connections from predicate and object units to SP units. Each SP permanently encodes a binding of one object to some number of single-place predicates and to *one role* of one multi-place predicate. For example, (chase Arnold Bill) is represented by two sub-propositions. The first encodes Arnold as the agent of chasing, and is denoted chase(Arnold _). Chase(Arnold _) has excitatory links to (a) the object unit for Arnold, (b) each single-place predicate unit that describes Arnold (e.g., person, male, etc.), and (c) the units for agent role of the two-place predicate "chase". The second SP, chase(_ Bill), has excitatory links to Bill, Bill's single-place predicates, and the patient role of "chase". Predicate units do not directly communicate; their only connections are to SP units. Predicate and object units are temporally yoked to SP units -- i.e., they fire only when they receive excitatory inputs from SPs. Therefore, when an SP fires, all predicate and object units to which it is connected also fire.

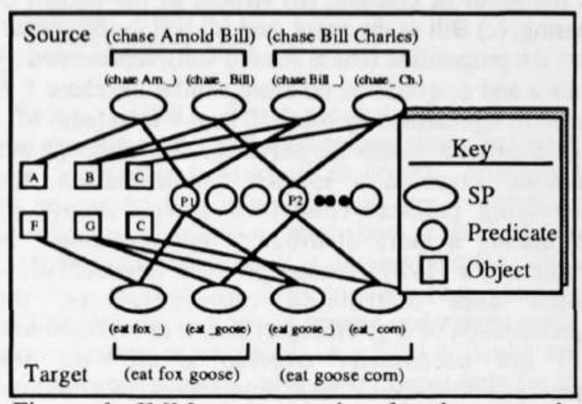

Figure 1. IMM representation for the example above. Not shown: Modifiable connections; connections between SPs within an analog; full SP-predicate connectivity.

Because predicate and object units are yoked to SPs, it is critical that separate SPs fire out of synchrony with one another. If chase(Arnold _) and chase(_ Bill) fired in synchrony, all their predicates and objects would also fire in synchrony, and it would be impossible to tell who was chasing whom. Therefore, SPs in the same proposition are assumed to share links that *desynchronize* their outputs. In the current implementation, all SPs within an analog are forcibly desynchronized (i.e., they are forced to fire one at a time).

Modifiable connections (of initial strength zero) exist between SP units across analogs and between object units across analogs; their function is described below.

The IMM Algorithm

The state of the network is updated in discrete cycles. The following sequence of operations is performed on each cycle:

1) One SP in the source analog fires; its output is set to 1.0 and propagated to the predicates, objects, and target SPs to which it is connected.
2) The SPs in the target update their activations (A_i) based on their excitatory inputs (E_i) and their lateral inhibitory inputs from one another. Lateral inhibition is implemented by the equation:

$$A_i = E_i^3 / \Sigma_j E_j^3.$$

3) The object units in the target update their activations based on their excitatory inputs from

the target SPs and their lateral inhibitory inputs from one another. Lateral inhibition is implemented by the above equation.

4) The SPs in the target recalculate their activations based on their excitatory inputs from the predicates, source SPs, and target objects, and their lateral inhibitory inputs from one another.

5) SPs in the target update their connections to SPs in the source, and objects in the target update their connections to objects in the source by the Hebbian rule

$$\Delta W_{ij} = A_i A_j,$$

where W_{ij} is the connection weight from source element j to target element i. Reflecting the one-to-one mapping constraint, connections to a target unit (both SP and object) and from a source unit are constrained to add to 1.0. This constraint is enforced by normalizing the modifiable connections at the end of each cycle by the ratio:

$$W_{ij} = W_{ij}/(W_{ij} + \Sigma_k W_{kj} + \Sigma_l W_{il}), i = k, l = j.$$

This normalization resets each connection according to its weight and the weights of all other connections to the same target SP and from the same source SP.

Simulations

Six tests of IMM are reported here, five based on small examples designed to test specific capacities of the model, and one based on a larger example. All tests were run ten times. The modifiable (SP-to-SP and object-to-object) connection weights were initialized to zero at the beginning of each run. Each run consisted of 20 iterations through the source analog, and each iteration consisted of one cycle (as described above) for each SP in the source analog. The firing order of the SPs was randomized at the beginning of each iteration. Mapping results are reported below in terms of the mean modifiable connection weights (object-to-object and, in one case, SP-to-SP) developed across analogs over the ten runs.

Test 1 was based on the analogs depicted in Figures 1 and 2. In this example, the predicates and objects are assumed to have no semantic overlap across the analogs, so the mapping must be solved purely on the basis of structural isomorphism. The most natural solution maps Bill to goose because they share the structural property of appearing in both the second place of the first proposition and the first place of the second. Because of the one-to-one mapping constraint, Arnold should then map to fox, and Charles to corn.

A detailed illustration of IMM's operation on the first cycle of this test is given in Figure 2. (1) The SP chase(Arnold _) fires and sends activation to the object unit for Arnold and the predicate P1 (shaded cells in Figure 2). P1 is a structural predicate indicating that its argument (Arnold) appears in the first place of some multi-place predicate, i.e., that its argument is the subject of some proposition. (2) The target SPs eat(fox _) and eat(goose _) each receive an excitatory input of 1.0 from P1. Since this is the first cycle through the source, all SP-to-SP connections and object-to-object connections are zero. After lateral inhibition, both target SPs' activations are 0.5. (3) The target objects fox and goose each receive inputs of 0.5 from their respective SPs, and after lateral inhibition, their activations are 0.5. (4) The target SPs recalculate their activations, again settling on 0.5 each. (5) SP-to-SP and object-to-object connections are updated. The weights from chase(Arnold _) to both eat(fox _) and eat(goose _) become 0.5, and the weights from Arnold to both fox and goose become 0.5.

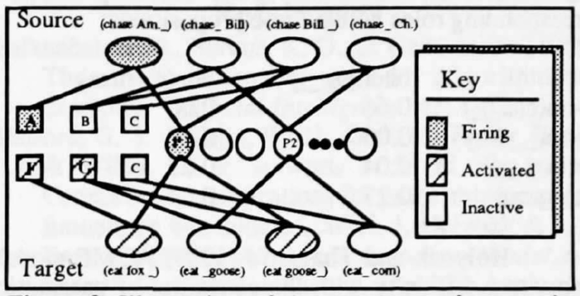

Figure 2. Illustration of the sequence of events in one cycle with the example above.

On the second, third and fourth cycles, these steps are repeated for chase(_ Bill), chase(Bill _) and chase(_ Charles), respectively. The second and third cycles are the most critical for finding the mapping from Bill to goose. On the second, Bill updates its connections both to goose and corn; on the third, it updates it connections to goose and fox. Over repeated iterations, Bill updates its connections to goose twice as often as it does to fox or corn, and -- due to the normalization of connection weights -- the Bill-goose connection eventually overpowers all other connections involving either Bill or goose. This reduction of other connections involving goose allows Arnold and Charles to map less strongly to goose, and more strongly to fox and corn, respectively. IMM successfully mapped these analogs on the basis of their structure alone. Over 10 runs, the mean object-to-object connection weights (after 20 iterations per run) were: Arnold --> fox = 0.99; Bill --> goose = 0.98; Charles --> corn = 0.99. All other object-to-object connections were zero.

Our second example tests IMM's sensitivity to the structure of information within propositions. It is based on the analogs:

<u>Source</u>	<u>Target</u>
(bite dog man)	(pet boy cat)
	(sit-on cat boy).

The predicates bite, pet, and sit-on are assumed to have no semantic overlap. The objects dog and cat share the single-place predicate unit for "animal", and the objects boy and man share the predicate "human". Both target propositions share four predicate units with the source proposition (p1, p2, animal, and human). However, (bite dog man) should map to (sit-on cat boy) because, in each case, an animal appears in the agent role, and a human in the patient role. In (pet boy cat), the human and animal are bound to opposite roles. Thus, this example constitutes a test of semantic similarity in which successful mapping depends on sensitivity to structure. Run with this example, IMM unambiguously mapped man to boy and dog to cat: dog --> cat = 0.99; man --> boy = 0.99; all other object-to-object connections were zero. As indicated by the the SP-to-SP connections, IMM also correctly mapped (bite dog man) to (sit-on cat boy), and correctly mapped corresponding roles within those propositions:

	bite(dog__)	bite(__ man)
sit-on(cat __)	0.66	0.00
sit-on(__ boy)	0.00	0.66
pet(boy __)	0.04	0.27
pet(__ cat)	0.27	0.04

Holyoak and Thagard's (1989) ACME model encounters difficulty with the constraint of pragmatic centrality. It does not respond appropriately to source elements that are marked as "important" (Spellman & Holyoak, in preparation; Hummel, Burns & Holyoak, in press). Therefore, IMM's treatment of important elements is a particularly critical test. Tests 3 - 5 examine the effect of object importance on mapping. Importance is implemented in IMM by allowing SPs containing important objects to fire more often than SPs containing objects not given extra importance. This convention is based on the assumptions that (1) firing rate reflects the activation of a unit, and (2) more important elements are more active than elements not given special importance. In these simulations, important SPs were allowed to fire twice (rather than only once) on each iteration through the source analog.

Tests 3 - 5 were based on the following analogs:

<u>Source</u>	<u>Target</u>
(chase coyote roadrunner)	(chase pig rabbit)
(eat Popeye spinach)	(eat rabbit carrot).

On every test with this example, IMM correctly mapped coyote exclusively to pig (connection weight 0.99) and spinach exclusively to carrot (connection weight 0.99). The interesting question concerns the degree to which Popeye vs. roadrunner will map to rabbit based on which (Popeye or roadrunner) is deemed "important". With neither given importance (Test 3), IMM mapped both equally to the rabbit (connection weights were 0.49), reflecting the ambiguity of the mapping. With special importance given to roadrunner (Test 4), IMM mapped roadrunner to rabbit more strongly than Popeye to rabbit (0.65 vs. 0.34). Similarly, with special importance given to Popeye (Test 5), it mapped Popeye to rabbit more strongly than it did roadrunner (0.66 vs. 0.33). Thus, IMM was able to adjudicate between ambiguous mappings on the basis of the relative importance of an element. In contrast, ACME produces less clear mappings for these simple examples (Hummel et. al., in press).

How does IMM differ from ACME so that the former succeeds on these simple tests of pragmatic centrality? In ACME, the success of a particular mapping depends upon the activity of the corresponding mapping unit relative to its competitors. An element (object or predicate) is marked as important by increasing the activities of all units representing mappings involving it. The increased activity associated with an important element's mapping units has the effect of increasing the tendency for those mappings to dominate other mappings. As such, the important element tends to map more to *everything*, rather than selectively mapping more to those other elements with which it already matches well.

By contrast, consider how an element in the source analog (SE) establishes a mapping with a target element (TE) in IMM. Each time an SE fires, its tendency to map to a specific TE is a function of (1) how closely the pattern of which the SE is a part matches the pattern(s) of which the TE is a part (as determined by the number of predicate units they share) and (2) how often and how strongly the SE has mapped to that TE in the past (as captured in the modifiable object-to-object and SP-to-SP connections). Like ACME, IMM implements increased importance as increased activity. In IMM, increased activity results in an increased firing rate. But note that an SE's *tendency* to map to any given TE, as defined by (1), has nothing to do with how often either unit fires; rather it is strictly a function of how well they match when they do fire. Therefore, increasing an SE's firing rate simply increases the *number of opportunities* that the SE has to map to those TEs for which it already has a preference. Each time an SE maps to a TE, they strengthen the connection between them at the expense of their other connections. Thus, a greater firing rate (i.e., more importance) means more opportunities for an SE to monopolize its preferred TE's connections.

The first five examples were designed to test specific capacities of the IMM architecture, and were deliberately kept small. The sixth test was designed to reveal IMM's capacity to deal with larger analogies. Test 6 is based on the "radiation to lightbulb" problem from Holyoak and Thagard (1989, Table 3). Space limitations prohibit full elaboration of the analogy, but it can be summarized as follows: The source analog states that there is a lightbulb with a broken filament that can be fused back together by a laser beam. The laser can generate either strong or weak beams. The strong beam would break the glass bulb surrounding the filament, but a single weak beam is too weak by itself to fuse the filament. The goal is to fuse the filament without breaking the glass bulb. The target analog states that there is a tumor surrounded by healthy tissue, and there is a radiation machine that can destroy the tumor. The radiation machine can generate either strong or weak rays. A strong ray would damage the healthy tissue surrounding the tumor, but the weak ray is too weak to destroy the tumor by itself. The goal is to destroy the tumor without damaging the healthy tissue. The intuitively correct mapping between these analogs generates the following object correspondences: laser --> radiation machine; strong laser beam --> strong rays; weak laser beam --> weak rays; tumor --> filament; glass bulb --> healthy tissue. IMM discovered all the correct mappings (mean modifiable connection strengths corresponding to correct mappings were all greater than 0.97) and did not discover any incorrect mappings (mean modifiable connection strengths corresponding to incorrect mappings were all zero).

Discussion

The initial simulations reported here, although run with a highly idealized version of IMM, have yielded encouraging results. IMM clearly demonstrates sensitivity to all the mapping constraints postulated by ACME: isomorphism, semantic similarity, and pragmatic centrality. It also scaled well to the larger analogy on which it was tested. Importantly, this behavior emerges from an architecture exploiting deliberately general principles for the representation of propositional information.

One strength of the IMM representation that we have not yet discussed is its capacity to scale with larger knowledge bases. Each proposition is encoded by a small number of SP units (typically three or fewer, depending on the number of argument places in the proposition). Therefore, the number of SP units required to represent an analogy grows linearly with the size of the analogs, and the number of modifiable connections between SPs across analogs grows linearly with the product of the number of propositions in the source and target.

The modifiable weights on object-object and SP-SP connections allow a relatively stable representation of the mapping between source and target elements to emerge. These modifiable connections are analogy-specific, making it possible for the system to learn contextually constrained correspondences between analogs without necessarily altering the structure of semantic memory. For example, the fact that a tumor maps to a filament in the context of the radiation/lightbulb analogy need not imply that these two concepts should now be closely related in semantic memory. At the same time, the asymptotic weights on the modifiable connections may provide inputs to post-mapping mechanisms that support the generation of analogical inferences about the target, as well as induction of relational generalizations based on the mapping between the source and target analogs.

It remains to be seen how IMM will perform with more realistic processing assumptions. The current implementation works largely because the sequence of events is globally and tightly controlled. If IMM proves highly sensitive to imperfections in the timing of events, it could be difficult to make it work with locally-controlled mechanisms for dynamic binding (i.e., for maintaining synchrony). Nonetheless, the indirect approach to analogical mapping seems sufficiently promising as to merit further exploration.

References

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence, 41,* 1-63.

Halford, G. S., Wilson, W. H., Guo, J., Gayler, R. W., & Wiles, J., & Stewart, J. E. M. (in press). Connectionist implications for processsing capacity limitations in analogies. In K. J. Holyoak & J. A. Barnden (Eds.), *Advances in connectionist and neural computation theory, Vol. 2: Analogical connections.* Norwood, NJ: Ablex.

Hofstadter, D. & Mitchell, (in press). An overview of the Copycat project. In K. J. Holyoak & J. A. Barnden (Eds.), *Advances in connectionist and neural computation theory, Vol. 2: Analogical connections.* Norwood, NJ: Ablex.

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

Hummel, J. E., & Biederman, I. (1992). Dynamic binding in a neural network for shape recognition. *Psychological Review,* in press.

Hummel, J. E., Burns, B. & Holyoak, K. J. (in press). 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.* Norwood, NJ: Ablex.

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, Computer and Information Science Department, University of Pennsylvania.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence, 46,* 159-216.

Spellman, B. A., & Holyoak, K. J. (in preparation). Pragmatics in analogical mapping. Research in progress, Department of Psychology, University of California, Los Angeles.