LISA: A Computational Model of Analogical Inference and Schema Induction

John E. Hummel and **Keith J. Holyoak**
Department of Psychology
University of California, Los Angeles
405 Hilgard Ave.
Los Angeles, CA 90095-1563
jhummel@psych.ucla.edu holyoak@psych.ucla.edu

Abstract

The relationship between analogy and schema induction is widely acknowledged and constitutes an important motivation for developing computational models of analogical mapping. However, most models of analogical mapping provide no clear basis for supporting schema induction. We describe LISA (Hummel & Holyoak, 1996), a recent model of analog retrieval and mapping that is explicitly designed to provide a platform for schema induction and other forms of inference. LISA represents predicates and their arguments (i.e., objects or propositions) as patterns of activation distributed over units representing semantic primitives. These representations are actively (dynamically) bound into propositions by synchronizing oscillations in their activation: Arguments fire in synchrony with the case roles to which they are bound, and out of synchrony with other case roles and arguments. By activating propositions in LTM, these patterns drive analog retrieval and mapping. This approach to analog retrieval and mapping accounts for numerous findings in human analogical reasoning (Hummel & Holyoak, 1996). Augmented with a capacity for intersection discovery and unsupervised learning, the architecture supports analogical inference and schema induction as a natural consequence. We describe LISA's account of schema induction and inference, and present some preliminary simulation results.

Schemas, Induction and Analogy

Cognitive scientists have long appealed to the notion of *schemas* to explain many aspects of human thinking (see Rumelhart, 1980). A schema is a generalized knowledge structure that characterizes the relationships applicable to some class of objects or events. For example, a "permission schema" (Cheng & Holyoak, 1985) might describe the class of situations in which some precondition must be satisfied before permission to perform an act is granted (e.g., one must be over 21 to drink alcohol); a "combustion engine schema" might specify the general relationships among the parts and operation of a combustion engine. Schemas support inferences. For example, a reasoner could use the permission schema to infer that a teenage beer drinker would be in violation of the rule; a reasoner could use the combustion engine schema to anticipate that a Honda 1.6 liter engine will not run after the gas line has been cut (even if that person has never actually cut the gas line of a Honda 1.6 liter engine). An essential property of schemas is that they are *relational structures* rather than simple lists of features or properties. That is, they explicitly specify how the properties of a class are related to one another: the (legal) drinking of alcohol is *contingent upon* being over 21; the gas line *carries* the gasoline to the carburetor.

An important question regarding schemas concerns their origin: How do we induce a general schema from experience with specific objects and events? As Holland, Holyoak, Nisbett and Thagard (1986) have emphasized, induction cannot proceed by blind search. Rather, it entails discovering systematic correspondences among the elements of specific known instances (objects or events) and using those correspondences to guide the induction of generalized schemas. For example, consider inducing a simple schema describing situations in which a man loves a woman, the woman likes flowers, and the man gives the woman flowers, based on the examples: (1) Jim loves Mary, Mary likes roses, and Jim gives Mary roses, and (2) Bill loves Susan, Susan likes tulips, and Bill gives Susan tulips. To generate the schema from the examples, it is first necessary to appreciate that Jim corresponds to Bill rather than Mary, that loves corresponds to loves rather than gives, and so forth. Knowledge of these correspondences is crucial for knowing which elements to generalize over.

One way to discover the appropriate correspondences is to draw an analogy between the instances. For this reason, it has been argued that analogical reasoning plays an important role in schema induction (Gentner, 1989; Holyoak & Thagard, 1995). Analogical reasoning generally involves using a relatively well-understood *source* analog to guide inferences about a less familiar *target* analog. This process has four major components: (1) using the target to retrieve a potentially useful source from memory; (2) mapping elements of the source onto elements of the target to identify systematic correspondences; (3) using the mapping to draw inferences about the target; and (4) inducing a generalized schema that captures the commonalties between the source and target (e.g., Carbonell, 1983; Gentner, 1989; Gick & Holyoak, 1983).

Numerous models of analogy have been developed that collectively address the stages of analog retrieval, mapping, and inference (e.g., Falkenhainer, Forbus & Gentner, 1989; Forbus, Gentner & Law, 1995; Halford et al., 1994; Hofstadter & Mitchell, 1994; Holyoak & Thagard, 1989; Thagard, Holyoak, Nelson & Gochfeld, 1990). On the face of it, such models provide a basis for modeling schema induction (because they provide a computational account of how to determine the correspondences between elements). However, this apparent connection, while widely recognized, has generally not been computationally realized. In part, this shortcoming reflects the way these models represent analog elements (Hummel & Holyoak, 1996). Most models of analogical mapping represent analogs either as collections of symbols composed into propositions (e.g., Falkenhainer et. al, 1989; Keane, 1995) or as localist units in a connectionist network (e.g., Holyoak & Thagard, 1989; Thagard et. al, 1990). Representations of this type can readily capture structure, making them very attractive as a basis for analogical mapping (an inherently structural problem). But lacking any detailed semantic decomposition, such representations are inadequate for generalization and building abstractions (basic components of schema induction). In general, the twin requirements of structure sensitivity and flexible generalization pose a serious challenge to the design of an architecture that aims to integrate analogical mapping with schema induction.

We have recently developed a computational model of analogy based on very different assumptions about the representation of analog elements and the operations that discover correspondences between them (Hummel & Holyoak, 1996; see Hummel & Holyoak, 1992, and Hummel, Meltz, Thompson, & Holyoak, 1994, for precursors). The heart of the model is an architecture for representing structured information in a distributed fashion, capturing both the structure-sensitivity of a localist or symbolic representation and the flexible generalization provided by a distributed connectionist representation. The model, called *LISA (Learning and Inference with Schemas and Analogies)*, is designed to provide an integrated account of all four major components of analogy use, from retrieval to schema induction. We have recently shown that LISA accounts for numerous findings concerning human analog retrieval and mapping (Hummel & Holyoak, 1996). This paper describes some preliminary results using LISA for schema induction and inference.

The LISA Model

Analog Representation, Retrieval and Mapping

We will briefly sketch the LISA model and its approach to analog retrieval and mapping. These operations are described in detail (along with simulation results) by Hummel and Holyoak (1996). The core of LISA's architecture is a system for actively (i.e., dynamically) binding roles to their fillers in working memory (WM) and encoding those bindings in LTM. LISA uses synchrony of firing for dynamic binding in WM (Hummel & Holyoak, 1992; Shastri & Ajjenagadde, 1993). Case roles and objects are represented in WM as distributed patterns of activation on a collection of *semantic units* (small circles in Figure 1); case roles and objects fire in synchrony when they are bound together and out of synchrony when they are not.

Every proposition is encoded in LTM by a hierarchy of *structure units* (see Figures 1 and 2). At the bottom of the hierarchy are *predicate* and *object* units. Each predicate unit locally codes one case role of one predicate. For example, *love1* represents the first (agent) role of the predicate "love", and has bidirectional excitatory connections to all the semantic units representing that role (e.g., *emotion1, strong1, positive1*, etc.); *love2* represents the patient role and is connected to the corresponding semantic units (e.g., *emotion2, strong2, positive2*, etc.). Semantically-related predicates share units in corresponding roles (e.g., *love1* and *like1* share many units), making the semantic similarity of different predicates explicit. Object units are just like predicate units except that they are connected to semantic units describing things rather than roles. For example, the object unit *Mary* might be connected to units for *human, adult, female*, etc., whereas *rose* might be connected to *plant, flower*, and *fragrant*.

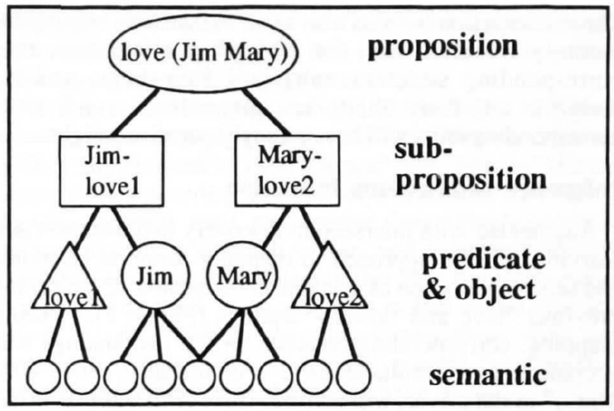

Figure 1: Illustration of the LISA representation of the proposition "love (Jim Mary)".

Sub-proposition units (*SPs*) bind roles to objects in LTM. For example, "love (Jim Mary)" would be represented by two SPs, one binding Jim to the agent of loving, and the other binding Mary to the patient role (Figure 1). The *Jim+agent* SP has bidirectional excitatory connections with *Jim* and *love1*, and the *Mary+patient* SP has connections with *Mary* and *love2*. *Proposition* (*P*) units reside at the top of the hierarchy and have bidirectional excitatory connections with the corresponding SP units. P units serve a dual role in hierarchical structures (such as "Sam knows that Jim loves Mary"), and behave differently according to whether they are currently serving as the "parent" of their own proposition or the "child" (i.e., argument) of another (see Hummel & Holyoak, 1996). It is important to emphasize that structure units do not encode semantic content in any direct way. Rather, they serve only to store that content in LTM, and to generate (and respond to) the corresponding synchrony patterns on the semantic units.

The final component of LISA's architecture is a set of *mapping connections* between structure units of the same type in different analogs. Every P unit in one analog shares a mapping connection with every P unit in every other analog; likewise, SPs share connections across analogs, as do objects and predicates. For the purposes of mapping and retrieval, analogs are divided into two mutually exclusive sets: a *driver* and one or more *recipients*. Retrieval and mapping are controlled by the driver. (There is no necessary linkage between the driver/recipient distinction and the more familiar source/target distinction.) LISA performs mapping as a form of guided pattern matching. As P units in the driver become active, they generate (via their SP, predicate and object units) patterns on the semantic units (one pattern for each role-argument binding). The semantic units are shared by all propositions, so the patterns generated by one proposition will activate one or more similar propositions in LTM (analogical access) or in WM (analogical mapping). Mapping differs from retrieval solely by the addition of the modifiable mapping connections. During mapping, the weights on the mapping connections grow larger when the units they link are active simultaneously, permiting LISA to learn the correspondences generated during retrieval. These connection weights also serve to constrain subsequent memory access. By the end of a simulation run, corresponding structure units will have large positive weights on their mapping connections, and non-corresponding units will have strongly negative weights.

Inference and Schema Induction

Augmented with intersection discovery and unsupervised learning, LISA's approach to mapping supports inference and schema induction as a natural consequence. Consider the previous "love and flowers" analogs (Figure 2). During mapping, corresponding elements in the two analogs will become active simultaneously. For instance, "love (Jim Mary)" in the driver, will activate "love (Bill Susan)" in the recipient. Corresponding elements (such as *Jim* and *Bill*) will fire in synchrony with one another, and non-corresponding elements (*Jim* and *Susan*) will fire out of synchrony (Figure 3a). *Jim* shares *male* with *Bill*, and *Mary* shares *female* with *Susan*, so a natural proposition to induce from these correspondences is "loves (male, female)" (Figure 3b). To induce this part of the schema, it is necessary to (a) make explicit what corresponding elements have in common, and (b) encode those common elements into LTM as a new proposition.

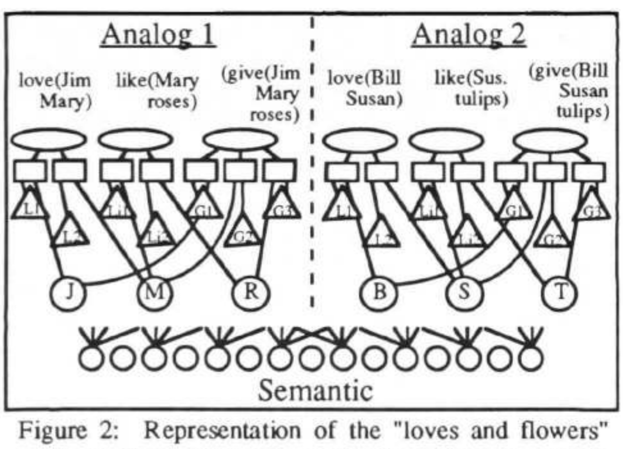

Figure 2: Representation of the "loves and flowers" analogy. Shapes (triangle, rectangle, etc.) correspond to classes of units as in Fig. 1. Not all connections are shown.

LISA performs (a) by means of a simple type of intersection discovery. Although we have described the activation of semantic units only from the perspective of the driver, the recipient analog also feeds activation to the semantic units. The activation of a semantic unit is a linear function of its inputs, so any semantic unit that is common to both the driver and recipient will receive input from both and become roughly twice as active as any semantic unit receiving input from only one analog. Common semantic elements are thus tagged as such by their activation values.

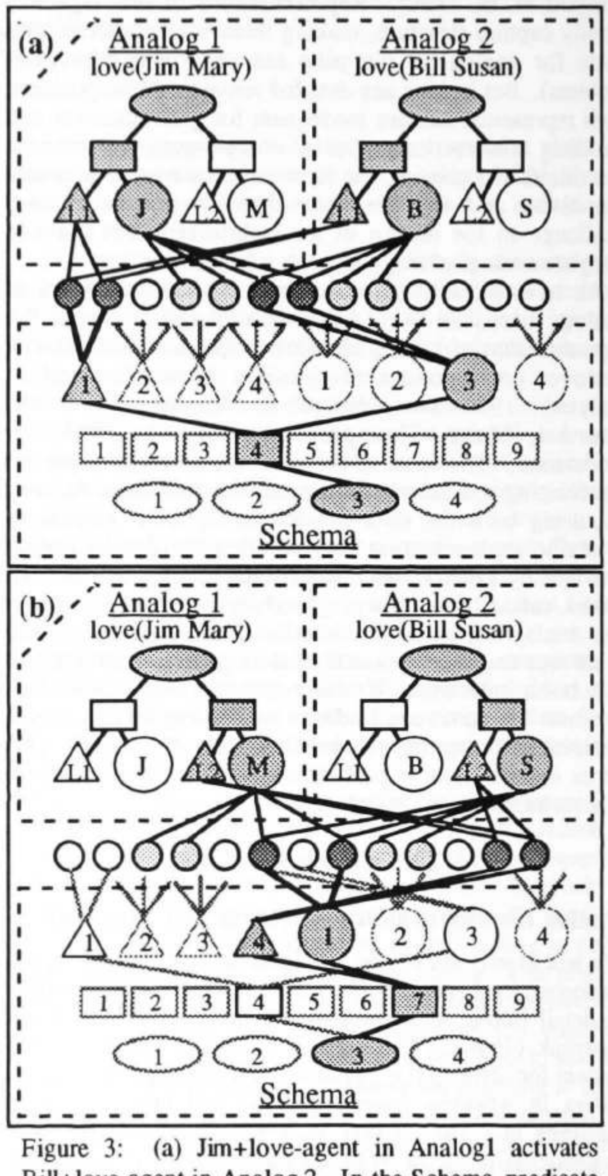

Figure 3: (a) Jim+love-agent in Analog1 activates Bill+love-agent in Analog 2. In the Schema, predicate unit 1 is recruited for love agent, and object unit 3 is recruited for the intersection of Jim and Bill ("human" and "male"). SP 4 is recruited for human male (object 3) bound to love-agent (predicate 1). Proposition unit 3 begins to be recruited. (b) Mary+love-patient in Analog 1 activates Susan+love-patient in Analog 2. Predicate 4 is recruited for love-patient; object 1 is recruited for "human" and "female". SP 7 is recruited for the binding of predicate 4 and object 1. Proposition unit 3 now codes "love(human male, human female)".

These common elements are encoded into LTM by means of an unsupervised learning algorithm. In addition to structure units representing the known source and target analogs, LISA has a collection of *unrecruited* structure units (i.e., units with random connections to one another and to the semantic units) that reside together in a third "schema analog" (Figure 3). Unrecruited predicate and object units have input thresholds that only allow them to receive input from highly active semantic units -- that is, semantic units that are common to both the driver and recipient analogs. Such semantic units are depicted in dark gray in Figure 3. Without the aid of an external teacher, these unrecruited schema units learn to respond to these common elements of the known analogs. Simultaneously, unrecruited SP units learn to respond to specific conjunctions of predicate, object, and (in the case of hierarchical propositions) P units, and unrecruited P units learn to respond to specific combinations of SP units. The result is that propositions describing the common elements of the known analogs are encoded into LTM as a third analog -- a schema. Figure 3 illustrates this process for one proposition in the "love and flowers" analogy.

As we will describe more fully below, LISA accomplishes analogical inference by the same unsupervised learning algorithm as used for schema induction, except that the unrecruited units reside not in a completely separate analog (the to-be-induced schema), but in the target itself.

Simulations

Schema Induction

To simulate the induction of the "love and flowers" schema, we gave LISA the following analogs (schematized in Figures 2 and 3):

Analog 1:	Analog 2:
love (Jim Mary)	love (Bill Susan)
like (Mary roses)	like (Susan tulips)
give (Jim Mary roses)	give (Bill Susan tulips)

Every object (person or flower) was represented by five semantic units. All people shared the features *person*, *Jim* and *Bill* shared *male1* and *male2*, and Mary and Susan shared *female1* and *female2*. Each person also had two unique features, so that no two people were identical. The flowers (*roses* and *tulips*) were each connected to five semantic units, three of which they shared. The predicates were represented by four semantic units each per case role. *Love* shared two units with *like* (*emotion* and *positive*) but only one with *give* (*positive*). There was also a third analog containing only "unrecruited" units -- i.e., units with initially random connections to one another and (in the case of object and predicate units) to the semantic units. This analog served as the schema-learning analog. It had 10 object units, 15 predicate units, 15 SP units, and 10 P units.

Every proposition in Analogs 1 and 2 was selected (activated) twice during the simulation run. As indices of mapping and schema induction, we recorded both the final values of the cross-analog mapping weights and the final values of the (initially random) connections in the third, schema analog. By the end of the run, there was one object unit in the schema analog that had learned large positive weights (i.e., > 0.7) to the semantic units *person*, *male1* and *male2*, and very small weights (< 0.2) to all other semantic units. This unit had been recruited to represent "male person" and accordingly had developed strong mapping weights to *Jim* and *Bill* and negative weights to all other objects. A different unit had been recruited to represent "female person" and a third to represent "generic flower." These units had strong mapping weights to *Mary* and *Susan* and to *roses* and *tulips*, respectively. The predicates recruited units in an analogous fashion, as did the SP and P units. Although this was only a "toy" example, LISA's performance with it suggests that it can induce a relational schema given specific analogs as examples.

As a more challenging test of LISA's ability to induce schemas from examples, we gave it simplified descriptions of Gick and Holyoak's (1980) "tumor" and "fortress" stories. These stories describe, respectively, situations in which a doctor uses many weak rays (rather than a single powerful one) to destroy a stomach tumor, and a general deploys several small groups of soldiers (rather than one large group) to capture a fortress. Presented with stories of this type and given a task in which they must map them onto one another, people will induce a more general "convergence schema" describing what the stories have in common (Gick & Holyoak, 1983).

We gave LISA these stories in a simplified eight-proposition format and ran them in the same general manner as for the "loves and flowers" analogy. As was the case for the previous example, LISA induced a schema by recruiting one structure unit for each element of the two analogs, abstracting over common elements (e.g., recruiting a single new unit for both "tumor" and "fortress", which play analogous roles in the two stories), and mapping the abstracted (schema) elements to the corresponding original story elements.

Analogical Inference

The same unsupervised learning algorithm that supports schema induction can be used in LISA to perform inductive inference by a form of "copy with substitution and generation" (Falkenhainer et al., 1989; Holyoak, Novick & Melz, 1994). Here, known elements or relations in one analog are used to "fill gaps" in a less familiar analog. Consider, for example, this "uncle" analogy, which we gave LISA:

Analog 1:	Analog 2:
father (Abe Bill)	father (Adam Bob)
brother (Charles Abe)	brother (Cary Adam)
uncle (Charles Bill)	

In Analog 1, Charles is Bill's uncle, a fact that is explicitly stated. In Analog 2, Cary is likewise the uncle of Bob, but this fact is not explicitly stated. We allowed LISA to map the two propositions in Analog 2 onto Analog 1, establishing the correspondences (as mapping connection weights) between Adam and Abe, Bob and Bill, and Cary and Charles.

We then allowed LISA to map Analog 1 back onto Analog 2. When the *father* and *brother* propositions in Analog 1 became active, they simply activated the corresponding propositions in Analog 2, reinforcing the established mappings. But when the *uncle* proposition became active in Analog 1, there was no corresponding proposition in Analog 2. Instead, Analog 2 had a collection of unrecruited units of the type used for schema induction in the previous examples. Because there were no predicate units pre-dedicated for the uncle relation, two unrecruited predicate units learned (without supervision) to respond to the two places of the uncle relation. The unit recruited for *uncle1* (the agent of the uncle relation) fired in synchrony with the *Cary* unit (because *Cary* was being driven by *Charles*, which was firing in synchrony with *uncle1* in Analog 1). As a result, an SP unit was recruited to respond to the conjunction of *Cary* and *uncle1*. Similarly, a predicate unit was recruited for *uncle2* and an SP unit was recruited for the conjunction *Bob-uncle2*. Finally, a P unit was recruited to respond to these two new SPs. Each of these units developed strong mapping connection weights to the corresponding units in Analog 1. The result of these operations was that LISA "inferred" that Cary is the uncle of Bob and stored this inference in LTM as a new proposition in Analog 2.

Conclusion

LISA provides a solution to the problem (forcefully posed by Fodor & Pylyshyn, 1988) of representing knowledge over a distributed set of units while preserving systematic relational structure. Like previous models based on symbolic or localist-connectionist representations, LISA is able to retrieve and map analogs based in large part on structural constraints. But in addition, LISA is able to capitalize on its distributed representations of meaning to integrate analogical mapping with a flexible generalization mechanism. This induction engine can make analogical inferences about a specific target analog; the same basic mechanism can create new schemas by finding and coding the structured intersection between multiple analogs. LISA thus provides an explanation of why people appear to induce generalized schemas as a natural consequence of using analogies (e.g., Novick & Holyoak, 1991; Ross & Kennedy, 1990). Analogical reasoning provides both the input and the trigger for inductive learning.

Achnowledgements

This work was supported by NSF Grant SBR-9511504.

References

Carbonell, J. G. (1983). Learning by analogy: Formulating and generalizing plans from past experience. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (pp. 137-161). Palo Alto, CA: Tioga Press.

Cheng, P. W., & Holyoak, K. J. (1985). Pragmatic reasoning schemas. *Cognitive Psychology, 17*, 391-416.

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

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.

Forbus, K. D., Gentner, D., & Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science, 19*, 141-205.

Gentner, D. (1989). The mechanisms of analogical learning. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 199-241). New York: Cambridge University Press.

Gick, M. L., & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology, 12*, 1-38.

Gick, M. L., & Holyoak, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology, 15*, 306-355.

Grossberg, S. (1971). Pavlovian pattern learning by nonlinear neural networks. *Proceedings of the National Academy of Science, 68*, 828-831. Washington, D.C.: National Academy of Science.

Halford, G. S., Wilson, W. H., Guo, J., Gayler, R. W., Wiles, J., & Stewart, J. E. M. (1994). Connectionist implications for processing capacity limitations in analogies. In K. J. Holyoak & J. A. Barnden (Eds.), *Advances in connectionist and neural computation theory, Vol. 2: Analogical connections* (pp. 363-415). Norwood, NJ: Ablex.

Hofstadter, D. R., & Mitchell, M. (1994). 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* (pp. 31-112). Norwood, NJ: Erlbaum.

Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. (1986). *Induction: Processes of inference, learning, and discovery*. Cambridge, MA: MIT Press.

Holyoak, K. J., Novick, L. R., & Melz, E. R. (1994). Component processes in analogical transfer: Mapping, pattern completion, and adaptation. In K. J. Holyoak & J. A. Barnden (Eds.), *Advances in connectionist and neural computation theory, Vol. 2: Analogical connections* (pp. 130-180). Norwood, NJ: Ablex.

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

Holyoak, K. J., & Thagard, P. (1995). *Mental leaps: Analogy in creative thought*. Cambridge, MA: MIT Press.

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

Hummel, J. E., & Holyoak, K. J. (1996). Distributed representations of structure: A theory of analogical access and mapping. Manuscript submitted for publication.

Hummel, J. E., Melz, E. R., Thompson, J., & Holyoak, K. J. (1994). Mapping hierarchical structures with synchrony for binding: Preliminary investigations. In A.

Ram & K. Eiselt (Eds.), *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* (pp. 433-438). Hillsdale, NJ: Erlbaum.

Keane, M. T. (1995). On order effects in analogical mapping: Predicting human error using IAM. In J. D. Moore & J. F. Lehman (Eds.), *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society* (pp. 449-454). Hillsdale, NJ: Erlbaum.

Novick, L. R.., & Holyoak, K. J. (1991). Mathematical problem solving by analogy. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 17*, 398-415.

Ross, B. H., & Kennedy, P. T. (1990). Generalizing from the use of earlier examples in problem solving. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 16*, 42-55.

Rumelhart, D. E. (1980). Schemata: The building blocks of cognition. In R. Spiro, B. Bruce, & W. Brewer (Eds.), *Theoretical issues in reading comprehension* (pp. 33-58). Hillsdale, NJ: Erlbaum.

Shastri, L., & 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*, 417-494.

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