
Toward an Understanding of Analogy within a Biological Symbol System

Keith J Holyoak and John E Hummel

Knowledge Representation in Models of Analogy

The past two decades have seen extraordinary growth in the investigation of analogical thinking. This work has included extensive experimental study as well as naturalistic observation of analogy use (e.g., Dunbar, chap. 9, this volume), theoretical analyses of the component processes involved in analogical transfer, and detailed simulations that instantiate theories of analogy in running computer programs (e.g., Forbus, chap. 2, and Kokinov and Petrov, chap. 3, this volume). Although many issues remain controversial, a remarkable degree of consensus has been reached about the basic nature of analogical thinking. A number of component processes have been distinguished (retrieval of

working memory, generation and evaluation of inferences, and induction of relational schemas). All these component processes depend in part on representations with a predicate argument (or role filler) structure, especially relations (predicates with multiple arguments) and higher order relations (predicates that take propositions as arguments). The centrality of relational representations to analogy was first clearly articulated in Gentner's (1983) structure mapping theory of mapping, and has been acknowledged in all major models of analogy. In addition to the role of structure, it is generally accepted that analogy use is guided by semantic similarity of concepts and by the goals of the analogist. The combined influence of structural, semantic, and pragmatic constraints was emphasized by the multiconstraint model of Holyoak and Thagard (1989, 1995).

Particularly given the interdisciplinary nature of work on analogy, the field can be viewed as one of the exemplary products of modern cognitive science. Analogy provides an important example of what appears to be a highly general cognitive mechanism that takes specific inputs from essentially any domain that can be represented in explicit propositional form, and operates on them to produce inferences specific to the target domain. At its best, analogy supports transfer across domains between analogs that have little surface resemblance but nonetheless share relational structure, and generates both specific inferences and more general abstractions from as few as two examples. These properties make analogy a central example of the power of explicit symbolic thought, which distinguishes the cognition of humans and perhaps other primates from that of other species. They also provide a major challenge for computational models, particularly types of neural network models that depend on huge numbers of training examples and that exhibit severely restricted transfer to novel inputs (Holyoak and Hummel 2000, Marcus 1998).

Because analogical thinking depends on representations that can explicitly express relations, all major computational models of analogy have been based on knowledge representations that express the internal structure of propositions, binding values to the arguments of predicates (or equivalently, fillers to relational roles). Such representations constitute symbol systems (Newell 1980, 1990). Most models have used traditional symbolic representations based on variants of predicate calculus, in which localist symbols are bound to roles by virtue of their positions in ordered lists. (This is true not only of models of analogical reasoning, but of symbolic models of cognitive processes, generally, cf. Holyoak and Hummel 2000.) For example, the proposition "John loves Mary" might be represented by the code *loves (John, Mary)*. Models based on such representations include SME (Falkenhainer, Forbus, and Gentner 1989, Forbus, chap. 2, this volume), ACME (Holyoak and Thagard 1989), ABMR (Kokinov 1994, chap. 3, this volume), and IAM (Keane, Ledgeway, and Duff 1994). These systems are broadly similar in that they take symbolic, propositional representations as inputs, use localist representations of individual concepts, and perform complex symbolic operations to generate plausible sets of candidate mappings.

Despite the considerable success such models have achieved in simulating aspects of human thinking, several reservations have been expressed about their psychological plausibility and potential for extension (Hummel and Holyoak 1997). First, the algorithms used in these models have typically ignored the capacity limits of human working memory, which their processing requirements appear to exceed. Although the working memory requirements for mapping can be reduced by using incremental algorithms to serialize processing (Keane, Ledgeway, and Duff 1994, Forbus, Ferguson, and Gentner 1994), these algorithms provide no principled basis for estimating the maximum working memory capacity available for mapping; that decision is left to the intuitions of the modeler. Second, none of these models have achieved a full integration of the major steps in analogical thinking (access, mapping, inference, and learning, but see Forbus chap. 2, and Kokinov

modeling efforts). The major models perform mapping and inference, and some have been combined with models of access (ACME with ARCS, Thagard et al. 1990, and SME with MAC/FAC, Forbus, Gentner, and Law 1995). However, none of these models have fully integrated a domain independent learning component capable of inducing new abstractions from analogical mappings (including mappings between nonidentical predicates).

For several years, we and our colleagues (Holyoak and Hummel 2000, Hummel and Holyoak 1992, 1993, 1996, 1997, forthcoming, Hummel, Burns, and Holyoak 1994, Hummel et al. 1994) have been working to develop a new architecture for analogical thinking and other forms of relational reasoning. Our aim is to develop a model with greater psychological and ultimately biological fidelity than previous efforts. Although we are far from achieving this goal, we believe cognitive science can benefit from a new focus on *biological symbol systems*—knowledge representations that capture the symbolic nature of human (and other primate) cognition, and might potentially be realized in the brain. Our models are high level abstractions of possible neural representations and processes, being based on densely connected networks of local computing elements. Our aim is to relate models of analogy to cortical functions, particularly the role of the prefrontal cortex in relational reasoning.

(see Benson 1993, Grafman, Holyoak, and Boiler 1995, Shallice and Burgess 1991 for reviews of prefrontal functions) These models also constitute relatively concrete proposals about the possible interface between perception and cognition (e.g., Hummel and Holyoak, forthcoming)

We will provide an overview of our current model of analogical thinking and show how it provides a unified account of the major stages of analogical processing. We will also sketch our initial effort at integrating abstract reasoning about relations with a perception-like module that can be used to make transitive inferences based on linear orderings. Finally, we will describe some recent behavioral and neuropsychological studies that connect relational reasoning with human working memory and the functions of prefrontal cortex.

Symbolic Connectionism. The LISA Model

At the heart of our effort is a neural network model of analogy called LISA (Learning and Inference with Schemas and Analogies), and the principles of symbolic processing on which it is based. LISA represents an approach to building symbolic representations in a neurally inspired computing architecture—an approach that we term *symbolic connectionism* (Hummel and Holyoak 1997, Holyoak and Hummel 2000).

Dynamic Binding in a Symbolic-Connectionist Model

We have argued that one basic requirement for relational reasoning is the ability to represent roles (relations) independently of their fillers (arguments), which makes it possible to appreciate what different symbolic expressions have in common, and therefore to generalize flexibly from one to the other. What gives a symbolic representation its power is precisely this capacity to bind roles to their fillers dynamically (i.e., to create the bindings as needed), and to represent the resulting bindings independently of the roles and fillers themselves (i.e., without changing the representation of the roles or fillers, Fodor and Pylyshyn 1988, Holyoak and Hummel 2000).

The symbolic connectionist framework that we have been developing seeks to realize these properties of symbol systems using the tools avail-

able neurally plausible computing architectures. Specifically, symbolic connectionist models (Holyoak and Hummel 2000; Hummel and Holyoak 1997) and their precursors (Hummel and Biederman 1992; von der Malsburg 1981) use synchrony of firing to bind representations of roles to representations of their fillers. The basic idea is that if two elements are bound together, then the neurons (or units in an artificial neural network) representing those elements fire in synchrony with one another; elements that are not bound together fire out of synchrony. For example, to represent "Jim loves Mary," the units for Jim would fire in synchrony with the units for lover, while Mary fires in synchrony with beloved. To represent "Mary loves Jim," the very same units would be placed into the opposite synchrony relations, so that Mary fires in synchrony with lover, while Jim fires in synchrony with beloved. Binding by synchrony satisfies the computational requirements for dynamic binding in a symbol system in that it permits the system to express binding in for both explicitly and independently of the representation of the bound elements (synchronizing "Jim" with "lover" on one occasion and "Mary" with it on another need not affect the representation of "lover" at all). As a basis for a biological theory of symbolic processing, binding by synchrony has the additional advantage of neural plausibility: there is growing evidence that real nervous systems use synchrony of firing as a basis for binding (see e.g., Singer 1999 for a recent review). As discussed later, symbolic connectionism—as an algorithmic theory of symbol systems—also provides a natural account of the fact that humans have a limited working memory capacity.

Analog Representation, Retrieval, and Mapping

We will now 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 (1997). 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 long-term memory (LTM). Case roles and objects are represented in WM as distributed patterns of activation on a collection of semantic units (small circles in figure 5.1); case roles and objects fire in synchrony when they are bound together and out of synchrony when they are not.

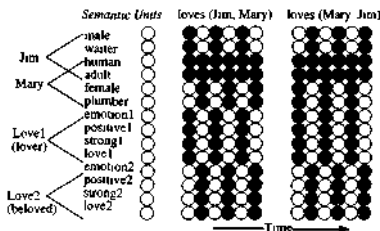


Figure 5.1
Representation of propositions in LISAs working memory (WM). Objects and

lection of semantic units. Objects are bound to relational roles in LISAs WM by synchrony of firing. Active semantic units are depicted in dark gray and inactive units in white.

Every proposition is encoded in LTM by a hierarchy of *structure units* (see figures 5.1 and 5.2). 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*, and so on, whereas *rose* might be connected to *plant*, *flower*, and *fragrant*.

Subproposition units (SPs) bind roles to fillers (objects or other propositions) in LTM. For example, *love (Jim, Mary)* would be represented

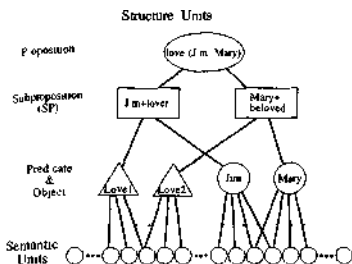


Figure 5.2
 Representation of a proposition in LISA's long term memory (LTM). Predicate units (triangles) and object units (circles) have bidirectional excitatory connections to semantic units describing the corresponding relational role (or object) and bind those semantics together in LTM. Role-filler bindings are encoded into LTM by subproposition (SP) units (rectangles) which have excitatory connections with the corresponding role and filler. Collections of role-filler bindings are excitatory connections with the corresponding SPs.

by two SPs: one binding Jim to the agent of loving, and the other binding Mary to the patient role (figure 5.1). 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 (Hummel and Holyoak 1997). 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.⁴ 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) synchronized patterns of activation on the semantic units (one pattern for each role argument binding). The semantic units are shared by all analogs so the patterns generated by a proposition in One analog will tend to activate one or more similar propositions in other analogs 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 permitting LISA to learn the correspondences generated during retrieval. These connection weights serve to constrain subsequent memory access and mapping. By the end of a simulation run corresponding structure Units will have large positive weights on their mapping connections and noncorresponding units will have strongly negative weights.

This algorithm accounts for a large number of findings in human analog retrieval and mapping including the role of working memory in mapping the effects of analog similarity and size on analog retrieval and some complex asymmetries between retrieval and mapping (see Hummel and Holyoak 1997). Of particular interest is LISA's account of the role of working memory—and the implications of its limitations—in analogical mapping. Binding by synchrony is inherently capacity limited in the sense that it is possible to have only a finite number of separate role-filler bindings simultaneously active and mutually out of synchrony with one another. Let us refer to the set of currently active but mutually desynchronized bindings as the *phase set* (the set of bindings that are firing

out of phase with one another) The phase set is the set of bindings LISA can process all at once," so the maximum size of the phase set corresponds to the capacity of LISA's working memory. A reasonable estimate for this capacity in humans is about four to six bindings (i.e., two or three propositions, see Hummel and Holyoak 1997). That is, under plausible estimates of the capacity of human working memory, LISA can "think about at most two or three propositions at once. As a result, analog retrieval and mapping are largely serial operations in LISA, in which propositions are mapped in packages consisting of at most three (and more often one).

The serial nature of LISA's mapping algorithm matters because mappings discovered early on can, by virtue of the mapping connections thereby acquired, strongly influence the mappings discovered later. Mapping performance therefore depends heavily on the order in which LISA maps the propositions, and on which propositions it considers together in the same phase set² (Putting two or more propositions into the same phase set effectively allows LISA to consider them in parallel, so that the mappings established for one do not influence those established for others). This property leads to some counterintuitive predictions about human performance, a number of which we are actively testing. For example, as elaborated later in this chapter, LISA predicts that any manipulation that leads human reasoners to group propositions in working memory in a way that aids mapping—including simply drawing a box around a pair of statements, and instructing subjects to think about these statements together—will improve performance on difficult mapping tasks. Preliminary tests (Kubose, Holyoak, and Hummel 1999) show that this prediction is correct. Importantly, this prediction derives directly from the intrinsic limitations on LISA's working memory.

Inference and Schema Induction

Augmented with unsupervised learning and intersection discovery, LISA's approach to mapping supports inference and schema induction as a natural extension (Holyoak and Hummel 2000; Hummel and Holyoak 1996). Consider an analogy between two pairs of lovers: Jim loves Mary,

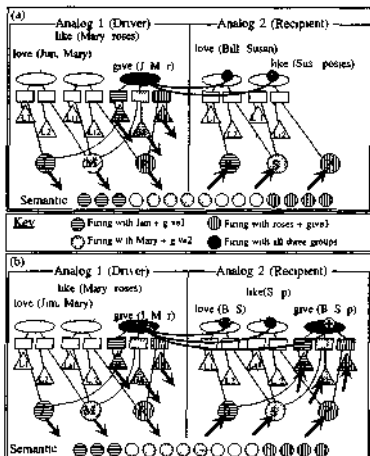


Figure 5.3

Illustration of analogical inference in LISA. (a) When *give (Jim Mary roses)* fires in the driver, it inhibits (via mapping connections, arcs terminating in dark circles) all the existing propositions in the recipient. (b) This situation serves as a cue to build new structure units (here, Give1, Give2, Give3, the P unit for *give (Bill Sally posies)*), and the corresponding SPs.

LISA connects these newly the object units *Bill Sally*, and *posies*). Structure units are connected together when they fire in synchrony.