A Symbolic-Connectionist Theory of Relational Inference and Generalization

John E. Hummel   Keith J. Holyoak
University of California, Los Angeles

Abstract

We present a theory of how relational inference and generalization can be accomplished within a cognitive architecture that is psychologically and neurally realistic. Our proposal is a form of *symbolic connectionism*: a connectionist system based on distributed representations of concept meanings, using temporal synchrony to bind fillers and roles into relational structures. We present a specific instantiation of our theory in the form of a computer simulation model, *LISA* (*Learning and Inference with Schemas and Analogies*). By using a kind of self-supervised learning, LISA can make specific inferences and form new relational generalizations, and hence acquire new schemas by induction from examples. We demonstrate the sufficiency of the model by using it to simulate a body of empirical phenomena concerning analogical inference and relational generalization.

A fundamental aspect of human intelligence is the ability to form and manipulate relational representations. Examples of relational thinking include our ability to appreciate analogies between seemingly different objects or events (Gentner, 1983; Holyoak & Thagard, 1995), our ability to apply abstract rules in novel situations (e.g., Smith, Langston & Nisbett, 1992), our ability to understand and learn language (e.g., Kim, Pinker, Prince & Prasada, 1991), and even our ability to appreciate perceptual similarities (e.g., Palmer, 1978; Goldstone, Medin & Gentner, 1991; Hummel, 2000a; Hummel & Stankiewicz, 1996). Relational inferences and generalizations are so commonplace that it is easy to assume that the psychological mechanisms underlying them are relatively simple. But this would be a mistake. The capacity to form and manipulate explicit relational representations appears to be a late evolutionary development (Robin & Holyoak, 1995), closely tied to the substantial increase in the size and complexity of the frontal cortex in the brains of higher primates, and most dramatically in humans (Stuss & Benson, 1986). Waltz et al. (1999) found that the ability to represent and integrate multiple relations is profoundly impaired in patients with severe degeneration of prefrontal cortex, and neuroimaging studies have implicated dorsolateral prefrontal cortex in various tasks that require relational reasoning (Christoff et al., 2001; Duncan et al., 2000; Kroger et al., in press; Prabhakaran et al., 1997), as well as in a variety of working-memory tasks (Cohen et al., 1997; Smith & Jonides, 1997).

Our goal in the present paper is to present a theory of how relational inference and generalization could be accomplished within a cognitive architecture that is psychologically and neurally realistic. We begin by reviewing the characteristics of relational thinking, and the logical requirements for realizing these properties in a computational architecture. We next present a theory of human relational reasoning motivated by these requirements, and present a specific instantiation of the theory in the form of a computer simulation model, *LISA* (*Learning and Inference with Schemas and Analogies*). LISA was initially developed as a theory of analog retrieval and mapping (Hummel & Holyoak, 1997), reflecting our assumption that the process of finding systematic correspondences between structured representations, which plays a central role in analogical reasoning, is fundamental to all relational reasoning (Hofstadter, 2001). Here we demonstrate that these same principles can be extended to account for both our ability to make specific inferences and our ability to form new relational generalizations, and hence to acquire new schemas by induction from examples. We demonstrate the sufficiency of the model by using it to simulate a body of empirical phenomena concerning analogical inference and relational generalization. (See Hummel & Holyoak, 1997, for simulations of data on retrieval and analogical mapping.)

Relational Reasoning

The power of relational reasoning resides in its capacity to generate inferences and generalizations that are constrained by the *roles* that elements play, rather than solely the properties of the elements themselves.   In the limit, relational reasoning yields universal inductive generalization from a finite and often very small set of observed cases to a potentially infinite set of novel instantiations. An important example of human relational reasoning is analogical inference. Suppose someone knows that Bill loves Mary, Mary loves Sam, and Bill is jealous of Sam (a *source* analog). The reasoner then observes that Sally loves Tom, and Tom loves Cathy (a *target* analog). A plausible analogical inference is that Sally will be jealous of Cathy. Mary corresponds to Tom based on their shared roles in the two *loves* (*x*, *y*) relations; and similarly, Bill corresponds to Sally and Sam to Cathy. The analogical inference depends critically on these correspondences (e.g., it is not a plausible analogical inference that Tom will be jealous of Cathy).

This same analysis, in which objects are mapped on the basis of common relational roles, can be extended to understand reasoning based on the application of schemas and rules. Suppose the reasoner has learned a schema or rule of the general form, "If person1 loves person2, and person2 loves person3, then person1 will be jealous of person3," and now encounters Sally, Tom and Cathy. By mapping Sally to person1, Tom to person2 and Cathy to person3, the inference that Sally will be jealous of Cathy can be made by extending the mapping to place Sally in the (unrequited) lover role (person1), Tom in the both-lover-and-beloved roles (person2), and Cathy in the beloved role (person3). When the schema has an explicit "if-then" form, as in this example,

the inference can readily be interpreted as the equivalent of matching the left-hand side ("if" clause) of a production rule and then "firing" the rule to create an instantiated form of the right-hand side ("then" clause), maintaining consistent local variable bindings across the "if" and "then" clauses in the process (Anderson, 1993; Newell, 1973).

There is also a close connection between the capacity to make relational inferences and the capacity to learn variablized schemas or rules of the above sort. If we consider how a reasoner might first acquire the "jealousy" schema, a plausible bottom-up basis would be induction from multiple specific examples. For example, if the reasoner observed Bill being jealous of Sam, and Sally being jealous of Cathy, and was able to establish the relational mappings and observe the commonalities among the mapped elements (e.g., Bill and Sally are both people in love, and Sam and Cathy are both loved by the object of someone else's affection), then inductive generalization could give rise to the jealousy schema.

The computational requirements for discovering these correspondences—and for making the appropriate inferences—include the following:

1. The underlying representation must bind fillers (e.g., Bill) to roles (e.g., lover). Knowing only that Bill, Mary and Sam are in various love relations will not suffice to infer that it is Bill who will be jealous of Tom.

2. Each filler must maintain its identity across multiple roles (e.g., the same Mary is both a lover and a beloved), and each role must maintain its identity across multiple fillers (e.g., *loves* (x, y) is the same basic relation regardless of whether the lovers are Bill and Mary or Sally and Tom). In turn, such maintenance of identities implies that the representation of a role must be independent of its fillers, and vice versa (Holyoak & Hummel, 2000; Hummel, 2002). If the representation of *loves* (x, y) depended on who loved whom (e.g., with one set of units for *Sally-as-lover* and a completely separate set for *Bill-as-lover*), then there would be no basis for appreciating that Sally loving Tom has anything in common with Bill loving Mary. And if the representation of Mary depended on whether she is a lover or is beloved, then there would be no basis for inferring that Bill will be jealous of Sam: it would be as though Bill loved one person, and someone else loved Sam.

3. Elements of the source and target can be mapped on the basis of their relational (i.e., role-based) correspondences. For example, we know that Sally maps to Bill because she also is a lover, despite the fact that she has a more direct similarity to Mary by virtue of being female.

4. The inference depends on an extension of the initial mapping. Because Sally maps to Bill on the basis of her role as an unrequited lover, it is inferred that she, like Bill, will also be jealous.

5. The symbols that compose these representations must have semantic content. In order to induce a general *jealousy* schema from the given examples, it is necessary to know what the people and relations involved have in common and how they differ. The semantic content of human mental representations manifests itself in countless other ways as well.

Together, requirements (1) - (4) imply that the human cognitive architecture is a symbol system, in which relations and their arguments are represented independently but can be bound together to compose propositions. We take (5) to imply that the symbols in this system are distributed representations that explicitly specify the semantic content of their referents. The capacity to satisfy these requirements is a fundamental property of the human cognitive architecture, and places strong constraints on models of that architecture (Hummel & Holyoak, 1997). Our proposal is a form of *symbolic connectionism*: a cognitive architecture that codes relational structures by dynamically binding distributed representations of roles to distributed representations of their fillers. We argue that the resulting architecture captures the meanings of roles and their fillers—along with the natural flexibility and automatic generalization enjoyed by distributed connectionist representations—in a way that no architecture based on localist symbolic representations can (see also Holyoak & Hummel, 2001). At the same time, the architecture proposed here captures the power of human relational reasoning in a way that traditional connectionist architectures (i.e., non-symbolic architectures, such as those that serve as the platform for back-propagation learning; see Marcus, 1998, 2001) cannot.

Core Properties of a Biological Symbol System

Representational Assumptions

We assume that human reasoning is based on distributed symbolic representations that are generated rapidly (within a few hundred milliseconds per proposition) by other systems, including those that perform high-level perception (e.g., Hummel & Biederman, 1992) and language comprehension (e.g., Frazier & Fodor, 1978).[1]   The resulting representations serve both as the basis for relational reasoning in working memory (WM), and the format for storage in long-term memory (LTM).
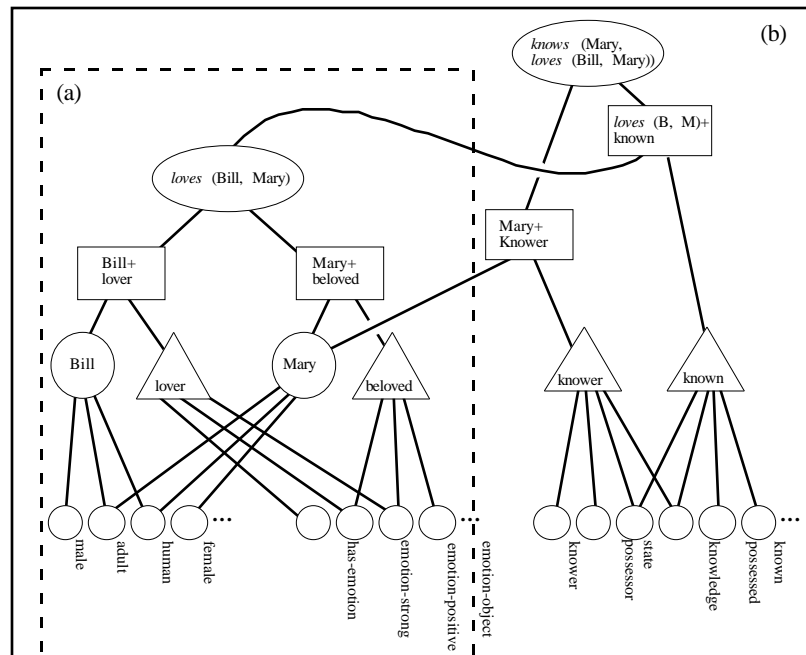

Figure 1. (a) A proposition is a four-tier hierarchy. At the bottom are the semantic features of individual object and relational roles (small circles). Objects and roles share semantic units to the extent that they are semantically similar (e.g., both Bill and Mary are connected to the semantic units human and adult.) At the next level of the hierarchy are localist units for individual objects (large circles; e.g., Bill and Mary) and relational roles (triangles; e.g., *lover* and *beloved*). At the third tier of the hierarchy, sub-proposition (SP) units (rectangles) bind objects to relational roles (e.g., Bill to the lover role), and at the top of the hierarchy, P units (ovals) bind separate role bindings into complete multi-place propositions(e.g., binding "Bill+*lover*" and "Mary+*beloved*" together to form *loves* (Bill, Mary)). (b) To represent a hierarchical proposition, such as *knows* (Mary, *loves* (Bill, Mary)), the P unit for the lower-level proposition (in the current example, *loves* (Bill, Mary)) serves in the place of an object unit under the appropriate SP of the higher-level proposition (in the current example, the SP for "*known+loves* (Bill, Mary)).

Individual propositions. In LISA, propositions play a key role in organizing the flow of information through WM. A proposition is a four-tier hierarchical structure (see Figure 1a). Each tier of the hierarchy corresponds to a different type of entity (semantic features, objects and roles, individual role bindings, and sets of role bindings comprising a proposition), and our claim is that the cognitive architecture represents all four tiers explicitly. At the bottom of the hierarchy are distributed representations based on *semantic* units, which represent the semantic features of

---

[1] This estimate of a few hundreds of milliseconds per proposition is motivated by work on the time-course of token-formation in visual perception (Luck & Vogel, 1997) and cognition (Kanwisher, 1991; Potter, 1976).

individual relational roles and their fillers. For example, the proposition *loves* (Bill, Mary) contains the semantic features of the *lover* and *beloved* roles of the *loves* relation (e.g., *emotion, strong, positive*), along with the features of Bill (e.g., *human*, *adult*, *male*) and Mary (e.g., *human*, *adult*, *female*). The three other layers of the hierarchy are defined over localist *token* units. At the second tier, the localist units represent objects (circles in Figure 1) and predicates (i.e., properties, as in *red* (*x*) or relational roles, as in the *lover* and *beloved* roles of *loves* (*x*, *y*); triangles in Figure 1). Each predicate unit represents one role of one relation (or property), and has bidirectional excitatory connections to the semantic units representing that role. Object units are just like predicate units except that they are connected to semantic units representing objects rather than roles.[2] At the third tier *sub-proposition* (*SP*) units bind individual roles to their fillers (either objects or other propositions). For example, *loves* (Bill, Mary) would be represented by two SPs, one binding *Bill* to *lover*, and the other binding *Mary* to *beloved* (see Figure 1A). SPs share bidirectional excitatory connections with their role and filler units. At the top of the hierarchy proposition (*P*) units bind role-filler bindings into complete propositions. Each P unit shares bidirectional excitatory connections with the SPs representing its constituent role-filler bindings.

This basic four-tier hierarchy can be extended to represent more complex embedded propositions, such as *know* (Mary, *loves* (Bill, Mary)) (see Figure 1B). When a proposition serves as an argument of another proposition, the lower-level P unit serves in the place of an object unit under the appropriate SP.

The localist representation of objects, roles, role-filler bindings (SPs) and propositions in the upper three layers of the hierarchy is important because it makes it possible to treat each such element *as a distinct entity* (cf. Hofstadter, 2001; Hummel, 2001; Hummel & Holyoak, 1993; Page, 2001). For example, if Bill in one analog maps to Sally in another, then this mapping will be represented as an excitatory connection linking the *Bill* unit directly to the *Sally* unit (as elaborated shortly). The localist nature of these units greatly facilitates the model's ability to keep this mapping distinct from other mappings. If the representation of Sally were distributed over a collection of object units (analogous to the way it is distributed over the semantic units), then any mapping learned for Sally (e.g., to Bill) would generalize automatically to any other objects sharing object units with Sally. Although automatic generalization is generally a strength of distributed representations, in this case it would be a severe liability: It simply does not follow that, just because Sally corresponds to Bill, all other objects therefore correspond to Bill to the extent that they are similar to Sally.

The need to keep entities distinct for the purposes of mapping is a special case of the familiar type-token differentiation problem (e.g., Kahneman, Treisman, & Gibbs, 1984; Kanwisher, 1991; Xu & Carey, 1996). Localist coding is frequently used to solve this and related problems in connectionist/neural network models of perception and cognition (e.g., Feldman & Ballard, 1982; Hummel & Biederman, 1992; Hummel & Holyoak, 1997; McClelland & Rumelhart, 1981; Poggio & Edelman, 1990; Shastri & Ajjanagadde, 1993). As in these other models, we assume that the localist units are realized neurally as small populations of neurons (as opposed to single neurons), where the members of each population code very selectively for a single entity. All tokens (localist units) referring to the same type are connected to a common set of semantic units. For example, if Bill is mentioned in more than one analog, then all the various object units representing Bill (one in each analog) would be connected to the same semantic units. The semantic units thus represent types.[3]

_____

[2] For the purposes of learning, it is necessary to know which semantic units refer to roles, and which refer to the objects filling the roles. For simplicity. LISA codes this distinction by designating one set of units to represent the semantics of predicate roles, and a separate set to represent the semantic properties of objects.

[3] We assume all tokens for a given object, such as Bill, are connected to a central core of semantic features (e.g., *human*, *adult*, and *male*, plus a unique semantic feature for the particular Bill in question—something such as *Bill#47*). However, different tokens of the same type may be connected to slightly different sets of semantics, as

It should be emphasized that LISA's representation of any complex relational structure, such as a proposition or an analog, is inevitably distributed over many units and multiple hierarchical levels. For example, the representation of *love* (Bill, Mary) in Figure 1A is not simply a single P unit, but the entire complex of token and semantic units. While the distributed semantic representation enables desirable forms of automatic generalization, the localist layers make it possible to treat objects, roles, and other structural units as discrete entities, each with potentially unique properties and mappings, and each with a fundamental identity that is preserved across propositions within an analog.

   Complete analogs. A complete *analog* (e.g., situation, story, or event) is encoded as a collection of token units, along with the connections to the semantic units. For example, the simple story, "Bill loves Mary, Mary loves Sam, and Bill is jealous of Sam," would be represented by the collection of units shown in Figure 2. Within an analog, a single token represents a given object, predicate, SP or proposition, no matter how many times that structure is mentioned in the analog. Separate analogs are represented by non-overlapping sets of tokens (e.g., if Bill is mentioned in a second analog, then a separate object unit would represent Bill in that analog). Although separate analogs do not share token units, all analogs are connected to the same set of semantic units. The shared semantic units allow separate analogs to communicate with one another, and thus bootstrap memory retrieval, analogical mapping, and all the other functions LISA performs.
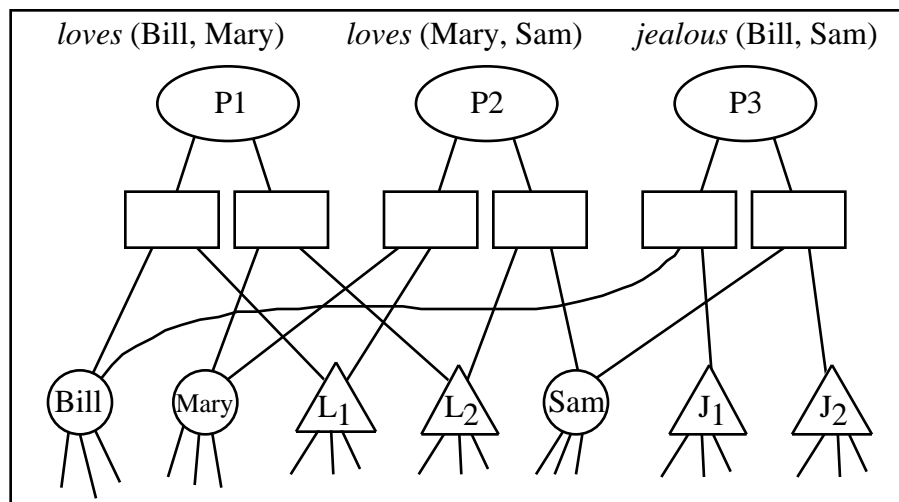

Figure 2. A complete analog is a collection of localist structure units representing the objects, relations and propositions stated in that analog. Semantic units are not shown. See text for details.

   Mapping connections. The final element of LISA's representations is a set of *mapping connections* that serve as a basis for representing analogical correspondences (mappings) between tokens in separate analogs. Every P unit in one analog has the potential to learn a mapping connection with every P unit in every other analog; likewise, SPs can learn connections across analogs, as can objects and predicates. As detailed below, mapping connections are created whenever two units of the same type in separate analogs are in WM simultaneously; the weights of existing connections are incremented whenever the units are coactive, and decremented whenever one unit is active and the other inactive. By the end of a run, corresponding units will have mapping connections with large positive weights, and non-corresponding units will either lack mapping connections altogether, or have connections with weights near zero. The mapping connections keep

---

dictated by the context of the particular analog. For instance, in the context of Bill loving Mary, *Bill* might be connected to a semantic feature specifying that he is a bachelor; in the context of a story about Bill going to the beach, *Bill* might be connected to a semantic feature specifying that he is a surfer.

track of what corresponds to what across analogs, and allow the model to use known mappings to constrain the discovery of additional mappings.

Processing Assumptions

A fundamental property of a symbol system is that it specifies the binding of roles to their fillers. For example, *loves* (Bill, Mary) explicitly specifies that Bill is the lover and Mary the beloved, rather than vice versa. For this purpose, it is not sufficient simply to activate the units representing the various roles and fillers making up the proposition. The resulting pattern of activation on the semantic units would create a bizarre concatenation of Bill, Mary and loving, rather than a systematic relation between two separate entities. It is therefore necessary to represent the role-filler bindings explicitly. The manner in which the bindings are represented is critical. Replacing the independent role and filler units (in the first two tiers of the hierarchy) with units for specific role-filler *conjunctions* (e.g., like SPs, with one unit for Bill-as-lover, another unit for Bill-as-traveler, etc.) would specify how roles are bound to fillers, but it would necessarily violate role-filler independence (Hummel, 2002; Hummel & Holyoak, 1997). And simply adopting a combination of conjunctive and independent units would result in a system with a combination of their limitations (Hummel & Biederman, 1992).

Binding is something a symbol system *does to* symbols for roles and fillers; it is not a property of the symbols themselves. What is needed is an explicit "tag" that specifies how things are bound together, but which permits the bound entities to remain independent of one another. The most common and neurally-plausible approach to binding that satisfies this criterion is based on synchrony of firing. The basic idea is that when a filler is bound to a role, the corresponding units fire in synchrony with one another, and out of synchrony with the units representing other role-filler bindings (Hummel & Biederman, 1992; Hummel & Holyoak, 1993, 1997; Shastri & Ajjanagadde, 1993; Strong & Whitehead, 1991; von der Malsburg, 1981). As illustrated in Figure 3, *loves* (Bill, Mary) would be represented by units for Bill firing in synchrony with units for lover, while units for Mary fire in synchrony with units for beloved. The proposition *loves* (Mary, Bill) would be represented by the very same units, but the synchrony relations would be reversed.

There is a growing body of neurophysiological evidence for binding by synchrony in visual perception (e.g., in striate cortex; Eckhorn et al., 1988; Gray, 1994; König & Engel, 1995; Singer, 2000; see also Alais & Blake, 1998; Lee & Blake, 2001; Usher & Donnelly, 1998) and in higher-level processing dependent on frontal cortex (Desmed & Tomberg, 1994; Vaadia et al., 1995). Numerous neural network/connectionist models use synchrony for binding. It has been applied in models of perceptual grouping (e.g., Eckhorn, et al., 1990; Horn, Sagi & Usher, 1992; von der Malsburg & Buhmann, 1992), object recognition (Hummel, 2001; Hummel & Biederman, 1992; Hummel & Saiki, 1993; Hummel & Stankiewicz, 1996, 1998), rule-based reasoning (Shastri & Ajjanagadde, 1993), episodic storage in the hippocampal memory system (Shastri, 1997), and analogical reasoning (Hummel & Holyoak, 1992, 1997).

In contrast to bindings represented by conjunctive coding, bindings represented by synchrony of firing are *dynamic* in the sense that they can be created and destroyed on the fly, and they permit the same units to participate in different bindings at different times. The resulting role-filler independence permits the meaning, interpretation or effect of a role to remain invariant across different (even completely novel) fillers, and vice versa (see Holyoak & Hummel, 2001). As noted previously, it is precisely this capacity to represent relational roles independently of their fillers that gives relational representations their expressive power, and that permits relational generalization in novel contexts. Hummel (2002) demonstrates mathematically that among the neurally-plausible approaches to binding in the computational literature, synchrony of firing is unique in its capacity to represent bindings both explicitly and independently of the bound entities.
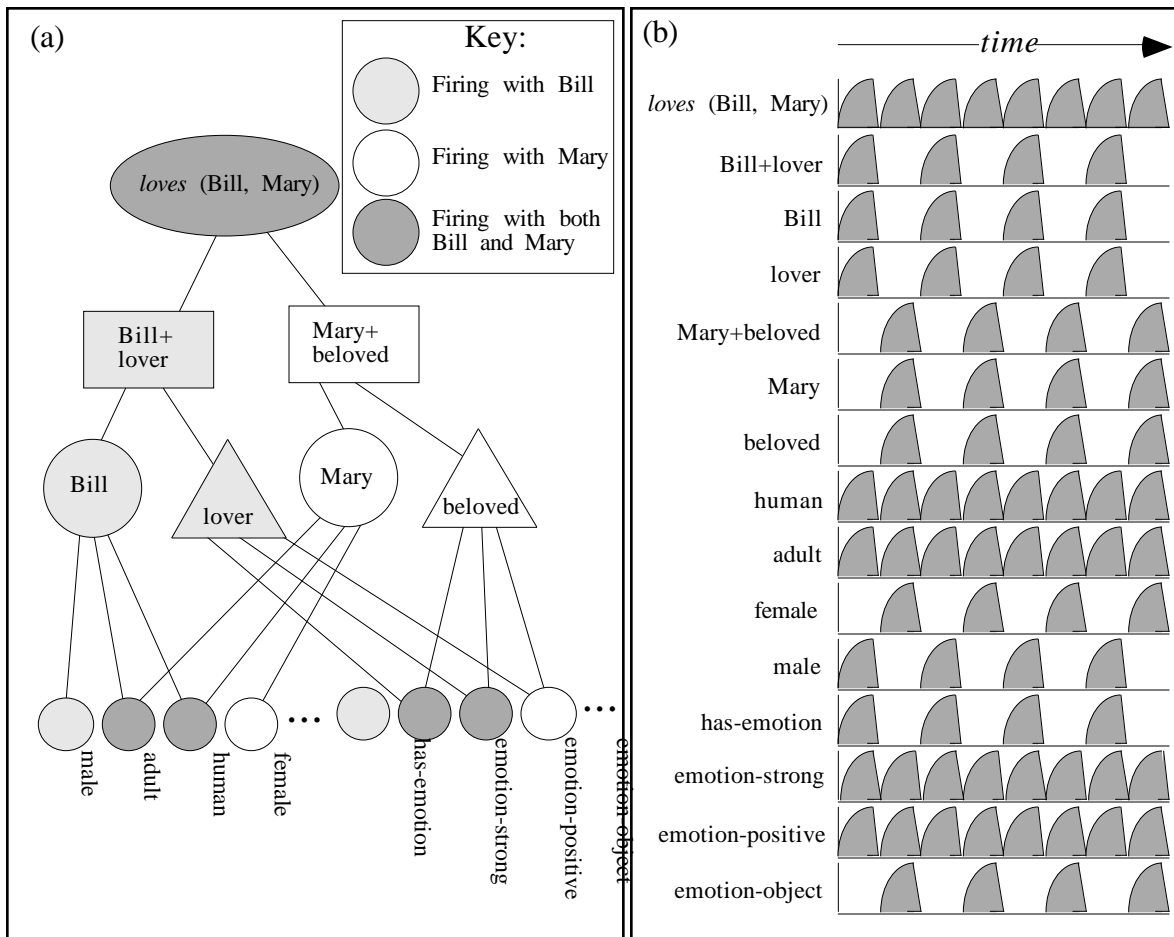
Figure 3. (a) Illustration of the representation of *loves* (Bill, Mary) in LISA's WM. Specked units are firing with Bill and *lover*, white units with Mary and *beloved*, and gray units with both. (b) A time-series illustration of the representation of *loves* (Bill, Mary) in LISA's WM. Each graph corresponds to one unit in Figure 3a—specifically, the unit with the same name as the graph. The abscissa of the graph represents time and the ordinate represents the corresponding unit's activation. Note that units corresponding to Bill and *lover* fire in synchrony with one another and out of synchrony of units for Mary and *beloved*; units common to both fire with both.

The need to represent roles independently of their fillers does not imply that the interpretation of a predicate must never vary as a function of its arguments. For example, the predicate *loves* suggests somewhat different interpretations in *loves* (Bill, Mary) versus *loves* (Bill, rib roast). Rather, the point is that the architecture of the symbol system must *permit* roles and fillers to remain invariant across different bindings. Effects of the filler on the interpretation of a role, as in the case of loving a person versus a food, must stem from the system's prior knowledge, rather than being an inevitable side effect of the way it binds roles to fillers. Thus if Bill were a cannibal, then he might indeed love Mary and rib roast in the same way; but a symbol system that was incapable of representing *loves* (*x*, *y*) in the same way regardless of the argument bound to *y* would be incapable of understanding the similarity of Bill's fondness for rib roast and his fondness for the unfortunate Mary.

The Hierarchical Structure of Memory

Using synchrony for dynamic binding imposes a hierarchical temporal structure on LISA's knowledge representations. This structure is analogous to Cowan's (1988, 1995) embedded-

processes model, with four levels of memory (see Figure 4). *Active memory* (cf. *long-term working memory*; Ericsson & Kintsch, 1995) is the active subset of LTM, including the various modality-specific slave systems proposed by Baddeley and Hitch (1974; Baddeley, 1986). During analogical mapping, the two analogs and the emerging mapping connections between them are assumed to reside in active memory. Following Cowan (1995), we assume that active memory has no firm capacity limit, but rather is time-limited, with activation decaying in roughly 20 sec unless it is reactivated by allocation of attention. When reasoning about large structures (e.g., mapping complex analogs), most of the relevant information will necessarily be first encoded into LTM, and then placed in active memory as needed.
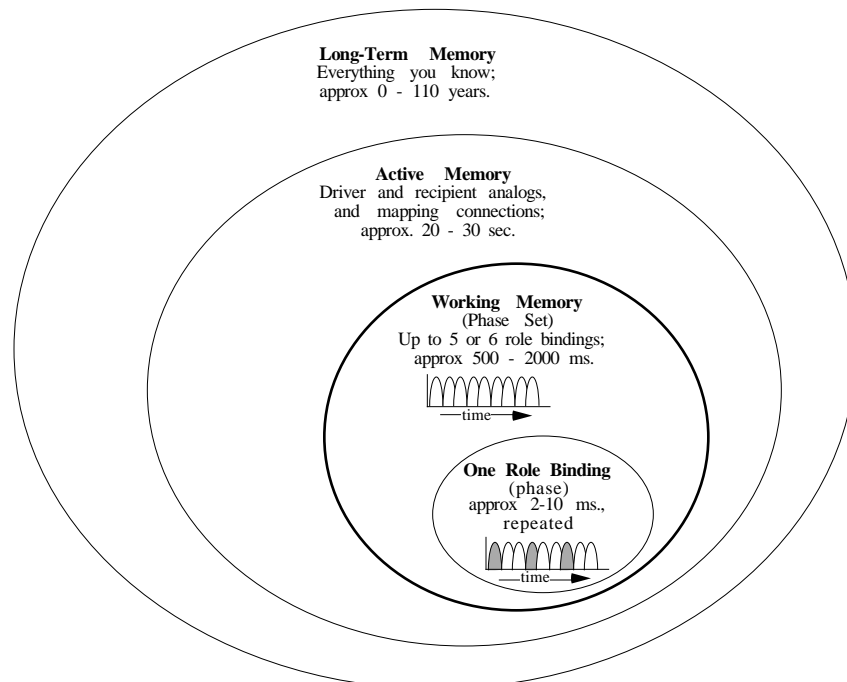

Figure 4. Illustration of the hierarchical structure of memory in LISA. We assume that *long-term memory* lasts a lifetime (mostly) and has, for all practical purposes, unlimited capacity. *Active memory* (Cowan 1988, 1995; cf. *Long-term working memory*; Ericsson & Kintsch, 1995) is the active subset of LTM. The two analogs and the emerging mapping connections between them are assumed to reside in active memory. We assume that active memory has no firm capacity limit, but is time-limited, with activation decaying in roughly 20 sec unless it is reactivated by attention (Cowan, 1995). *Working memory* (WM) consists of the set of role bindings (*phases*) that are simultaneously active and mutually *out* of synchrony with one another. Its capacity is limited by the number of phases it is possible to keep mutually desynchronized (roughly 5 or 6). We assume that role bindings reside in WM for a period of time ranging from roughly half a second to a few seconds. The smallest unit of memory in LISA is the individual role binding (phase), which we assume oscillates in WM (although not necessarily in a regular periodic manner), with each peak lasting a few milliseconds.

Within active memory, a very small number of role bindings can enter the phase set—the set of mutually desynchronized role bindings. We will use the term "working memory" to refer specifically to the phase set. Each "phase" in the set corresponds to one role binding; a single phase corresponds to the smallest unit of WM. The phase set corresponds to the current focus of attention—what the model is thinking about "right now"—and is the most significant bottleneck in the system. The size of the phase set—i.e., the capacity of WM—is determined by the number of role-filler bindings (phases) it is possible to have simultaneously active but mutually out of synchrony. This number is necessarily limited, and its value is proportional to the length of time between successive peaks in a phase (the period of the oscillation) divided by the duration of each

phase (at the level of small populations of neurons) and/or temporal precision (at the level of individual spikes; see also Lisman & Idiart, 1995).

There is evidence that binding is accomplished by synchrony in the 40 hz (gamma) range, meaning a neuron or population of neurons generates one spike (or burst) approximately every 25 ms (see Singer & Gray, 1995, for a review). According to Singer and Gray (1995), the temporal precision of spike timing is in the range of 4 - 6 ms. With a 5 ms duration/precision and a 25 ms period, these figures suggest a WM capacity of approximately 25/5 = 5 role bindings (Singer & Gray, 1995; see also Luck & Vogel, 1997). LISA's architecture for dynamic binding thus yields a principled estimate of the maximum amount of information that can be processed together during analogical mapping: 4-6 role bindings, or roughly 2-3 propositions. (As detailed in Appendix A, the algorithm LISA uses to allow multiple role-bindings to time-share efficiently is inherently capacity-limited.) Other researchers (e.g., Hummel & Biederman, 1992; Hummel & Holyoak, 1993, 1997; Hummel & Stankiewicz, 1996; Lisman & Idiart, 1995; Shastri & Ajjanagadde, 1993) have suggested similar explanations for the capacity limits of visual attention and working memory. Behavioral work with human subjects suggests a remarkably similar figure for the capacity of visual working memory (four to five items; Bundesen, 1998; Luck & Vogel, 1997; Sperling, 1960) and working memory for relations (three to five items; Broadbent, 1975; Cowan, 1995, 2001).

Capacity Limits and Sequential Processing

Because of the strong capacity limit on the phase set, LISA's processing of complex analogs is necessarily highly sequential: LISA can only hold at most three propositions in WM simultaneously, so it must map large analogies in small pieces. The resulting algorithm—which serves as the basis of analog retrieval, mapping, inference and schema induction—is analogous to a form of guided pattern recognition. At any given moment, one analog is the focus of attention and serves as the *driver*. One (or at most three) at a time, propositions in the driver become active, generating synchronized patterns of activation on the semantic units (one pattern per SP). In turn, these patterns activate propositions in one or more *recipient* analogs in LTM (during analog retrieval) or in active memory (during mapping, inference and schema induction). Units in the recipient compete to respond to the semantic patterns in much the same way that units for words or objects compete to respond to visual features in models of word recognition (e.g., McClelland & Rumelhart, 1981) and object recognition (e.g., Hummel & Biederman, 1992).

For example, consider a simplified version of our "jealousy" analogy in which males map to males and females to females. (In the analogy discussed previously, males map to females and vice versa. As described later, LISA can map such analogies. But for our current purposes, the example will be much clearer is we allow males to map to males and females to females.) As in the previous example, let the source analog state that Bill loves Mary, Mary loves Sam, and Bill is jealous of Sam. Let the target state that Tom loves Cathy and Cathy loves Jack (Figure 5). Let the source be the driver and the target the recipient. When the proposition *loves* (Bill, Mary) fires in the driver, the patterns of activation on the semantic units will tend to activate the units representing *loves* (Tom, Cathy) in the recipient. Specifically, the semantics activated by *Bill* (which include *human*, *adult* and *male*) will tend to activate *Tom* and *Jack* equally. The semantics of *lover* (activated by *lover* in the driver) will activate *lover* in the recipient. *Lover* will activate the SPs for both *Tom+lover* and *Cathy+lover* equally. *Tom* will excite *Tom+lover*, and *Jack* will excite *Jack+beloved*. Note that the SP *Tom+lover* is receiving two sources of input, whereas *Cathy+lover* and *Jack+beloved* are each receiving one source of input. *Tom+lover* will therefore inhibit *Cathy+lover* and *Jack+beloved* to inactivity.

Similar operations will cause *Cathy+beloved* to become active in the target when *Mary+beloved* fires in the source. Together, *Tom+lover* and *Cathy+beloved* will activate the P unit for *loves* (Tom, Cathy). All these operations cause *Tom* to become active whenever *Bill* is active (i.e., *Tom* fires in synchrony with *Bill*), *Cathy* to become active whenever *Mary* is active, and the roles of *loves* (x, y) in the recipient to become active whenever the corresponding roles are active in the driver. These synchrony relations between the driver and the recipient drive the discovery of mappings: LISA assumes that units that fire together map to one another (accomplished by a simple

form of Hebbian learning, detailed shortly). Thus, *Bill* maps to *Tom*, *Mary* to *Cathy*, *lover* to *lover* and *beloved* to *beloved*. By the same operations, LISA learns that *Sam* maps to *Jack*, the only exception being that the mappings discovered previously aid in the discovery of this latter mapping.
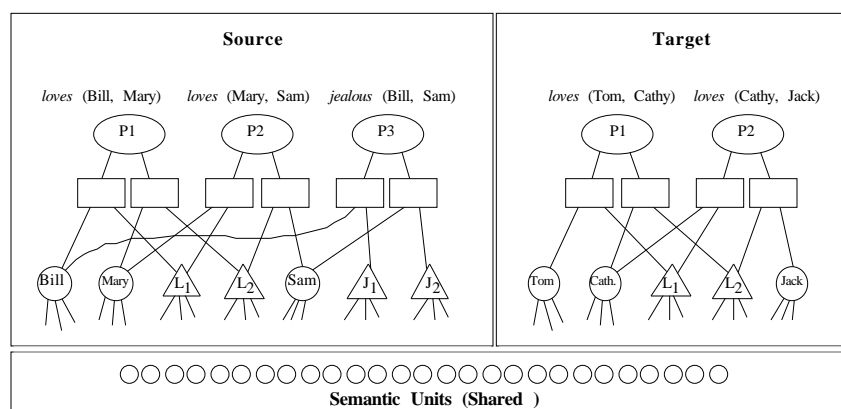

Figure 5. LISA's representations of two analogs. The source states that Bill loves Mary (P1), Mary loves Sam (P2), and Bill is jealous of Sam (P3). The target states that Tom loves Cathy (P1) and Cathy loves Jack (P2). See text for details.

The sequential nature of mapping, as imposed by LISA's limited-capacity working memory, is analogous to the sequential scanning of a visual scene, as imposed by the limits of visual working memory (see, e.g., Irwin, 1992). Just as the visual field typically contains more information than can be simultaneously foveated, so active memory will typically contain more information than can be simultaneously placed in WM. And just as eye movements systematically guide the flow of information through foveal vision, LISA's attentional mechanisms guide the flow of propositions through the phase set.

As a general default, we assume that propositions in the driver fire in roughly the order they would naturally be stated in a text—that is, in accord with the principles of text coherence (see Hummel & Holyoak, 1997)—and that the most important propositions (i.e., pragmatically central propositions, such as those on the causal chain of the episode, and those pertaining to goals; see Spellman & Holyoak, 1996) tend to fire earlier and more often than less important propositions. Also by default, propositions are fired one at a time (i.e., with only one proposition in a phase set), reflecting the assumption that holding multiple propositions in working memory requires greater cognitive effort than thinking about only one fact at a time. If multiple propositions need to be considered in the phase set simultaneously (e.g., in cases when the mapping would be structurally ambiguous based on only a single proposition), we assume that propositions are more likely to be placed together into WM if they are connected in ways that create textual coherence (Kintsch & van Dijk, 1978). In particular, propositions are more likely to enter WM together if they are arguments of the same higher-order relation (especially a causal relation), or if they share arguments or predicates, than if they are unrelated (Hummel & Holyoak, 1997; Kubose, Holyoak & Hummel, in press). In the original LISA (Hummel & Holyoak, 1997), the firing order and grouping of propositions had to be specified by hand, and the constraints stated above served as guidelines for doing so. As elaborated under *The Operation of LISA*, the current model contains several mechanisms that allow it to set its own firing order and groupings in a manner that conforms to the above constraints.

LISA's ability to discover a mapping between two analogs is strongly constrained by the order in which it fires propositions in the driver, and by which propositions it puts in WM simultaneously. This is especially true for structurally complex or semantically ambiguous analogies (Kubose et al., in press). Mappings discovered early in the mapping process constrain the discovery of later mappings. If the first mappings to be discovered correspond to structurally "correct" mappings, then they can aid significantly in the discovery of subsequent mappings. But if the first mappings correspond to structurally incorrect mappings, then they can lead LISA down a

garden path that causes it to discover additional incorrect mappings. Mapping performance is similarly affected by which propositions LISA puts together into WM. For example, recall the original analogy between two pairs of lovers: In one analog, Bill loves Mary, and Mary loves Sam; in the other, Sally loves Tom, and Tom loves Cathy. Although it is readily apparent that Bill corresponds to Sally, Mary to Tom and Sam to Cathy, these mappings are in fact structurally ambiguous unless one considers both statements in a given analog *simultaneously*: By itself, *loves* (Bill, Mary) bears as much structural similarity to *loves* (Tom, Cathy) (the incorrect mapping) as it does to *loves* (Sally, Tom) (the correct mapping). (Indeed, the genders of the characters involved tend to favor mapping Bill to Tom and Mary to Cathy.) Only by considering *loves* (Bill, Mary) simultaneously with *loves* (Mary, Sam) is it possible to discover the structurally correct mapping.

Analogical Inference and Schema Induction

Our final set of core theoretical claims concern the relationship between analogicalmapping, analogical and rule-based inference, and schema induction. We assume that both schema induction and analogical (and rule-based) inference depend on structure mapping, and that exactly the same algorithm is used for mapping in these tasks as is used for mapping analogies. A related claim is that the processes of rule- and schema-based inference are governed by exactly the same algorithm as the process of analogical inference. That is, analogical, rule- and schema-based inferences are all special cases of the same algorithm. These claims imply that brain damage and other factors (e.g., WM load imposed by a dual task) that affect reasoning in one of these domains should have similar effects on reasoning in all the others.

Summary

Our approach is based on a handful of core theoretical claims (see Table 1 for a summary of these claims and their instantiation in the model): (1) The mind is a symbol system that represents relationships among entities as explicit propositions. (2) A proposition is a four-tier hierarchical structure that represents the semantic properties of objects and relational roles in a distributed fashion, and simultaneously represents roles, objects, role-filler bindings and propositions in a sequence of progressively more localist units. (3) These hierarchical representations are generated rapidly (i.e., in a few hundreds of milliseconds per proposition) in working memory, and the resulting structure serves both to represent the proposition for the purposes of relational reasoning and to store it in LTM. (4) In the lowest layers of the hierarchy, roles are represented independently of their fillers, and must therefore be bound together dynamically when the proposition enters WM. (5) The need for dynamic role-filler binding gives rise to the capacity limits of WM. (6) Relational representations are generated and manipulated in a hierarchy of temporal scales, with individual role-filler bindings (phases) at the bottom of the hierarchy, combining to form complete propositions (phase sets) at the next level of the hierarchy, and multiple propositions forming the representation of an analog in active and long-term memory. (7) Due to the capacity limits of WM, structure-mapping is necessarily a sequential process. (8) Also due to capacity limits, mapping is directional, proceeding from a driver analog to the recipient(s). (9) As a result of (7) and (8), mapping is sensitive to factors that affect the order in which propositions are mapped, and which propositions enter WM together. (10) Principles of text coherence, pragmatic centrality, and expertise determine which propositions a reasoner is likely to think about together, and in what order. (11) Analog retrieval, mapping, inference and schema induction, as well as rule-based reasoning, are all special cases of the same sequential, directional algorithm for mapping.

Table 1.

Theoretical Claim	Instantiation in LISA
1) The mind is a symbol system that represents relationships among entities as explicit propositions.	LISA represents knowledge as propositions that explicitly represent relations as predicates, and explicitly represents the binding of relational roles to their arguments.
2) A proposition is a four-tier hierarchical structure that represents the semantic properties of objects and relational roles in a distributed fashion, and simultaneously represents roles, objects, role-filler bindings and propositions in a sequence of progressively more localist units.	LISA represents each tier of the hierarchy explicitly, with (a) units that represent the semantic features of roles and objects in a distributed fashion at the bottom of the hierarchy, and localist structure units to represent tokens for (b) relational roles and objects (c) role-filler bindings, and (d) complete propositions.
3) These hierarchical representations are generated rapidly (i.e., in a few hundreds of milliseconds per proposition) in working memory, and the resulting structure serves both to represent the proposition for the purposes of relational reasoning and to store it in LTM.	The same hierarchical collection of units represents a proposition in WM for the purposes of mapping and for storage in LTM (with the exception that synchrony of firing dynamically binds roles to their fillers in WM but not in LTM).
4) In the lowest layers of the hierarchy, roles are represented independently of their fillers, and must therefore be bound together dynamically when the proposition enters WM.	At the level of semantic features and object/predicate units, LISA represents roles independently of their fillers, in the sense that the same units represent a role regardless of its filler, and the same units represent an object regardless of the role(s) to which is bound. LISA uses synchrony of firing to dynamically bind distributed representations of relational roles to distributed representations of their fillers in WM.
5) The need for dynamic role-filler binding gives rise to the capacity limits of WM.	The capacity of LISA's WM is given by number of SPs it can fit into the phase set simultaneously, and this number is necessarily limited.
6) Relational representations are generated and manipulated in a hierarchy of temporal scales, with individual role-filler bindings (phases) at the bottom of the hierarchy, combining to form complete propositions (phase sets) at the next level of the hierarchy, and multiple propositions forming the representation of an analog in active and long-term memory.	In LISA, roles fire in synchrony with their fillers and out of synchrony with other role-filler bindings. Role-filler bindings belonging to the same phase set oscillate in closer temporal proximity than role-filler bindings belonging to separate phase sets.
7) Due to the capacity limits of WM, structure-mapping is necessarily a sequential process.	A single mapping may consist of many separate phase sets, which must be processed one at a time.
8) Mapping is directional, in the sense that it proceeds from the driver to the recipient(s).	Activation starts in the driver and flows into the recipient, initially only through the semantic units, and later through both the semantic units and the emerging mapping connections.
9) Mapping is sensitive to factors that affect the order in which propositions are mapped, and which propositions enter WM together.	LISA's mapping performance is highly sensitive to the order and groupings of propositions in the phase set.
10) Principles of text coherence, pragmatic centrality, and expertise determine which propositions a reasoner is likely to think about together, and in what order.	Importance (pragmatic centrality) and support relations among propositions (reflecting constraints on firing order) determine the order in which LISA fires propositions in WM, and which propositions (if any) it puts together in the phase set.
11) Analog retrieval, mapping, inference and schema induction, as well as rule-based reasoning, are all special cases of the same sequential, directional algorithm for mapping.	LISA uses the same mapping algorithm and self-supervised learning algorithm to perform all these functions.

The Operation of LISA

Overview

Figure 6 illustrates our core claims about the relationship between analogical access, mapping, inference and schema induction. (1) Analog *access* (using a target analog to retrieve an appropriate source from LTM) is accomplished as a form of guided pattern matching. Propositions in a target analog generate synchronized patterns of activation on the semantic units, which in turn activate propositions in potential source analogs residing in LTM. (2) The resulting coactivity of driver and recipient elements, augmented with a capacity to learn which structures in the recipient were coactive with which in the driver, serves as the basis for analogical *mapping* (discovering the correspondences between elements of the source and target). (3) Augmented with a simple algorithm for self-supervised learning the mapping algorithm serves as the basis for *analogical inference* (using the source to draw new inferences about the target). (4) Finally, augmented with a simple algorithm for intersection discovery, self-supervised learning serves as the basis for *schema induction* (inducing a more general schema or rule based on the common elements of the source and target).
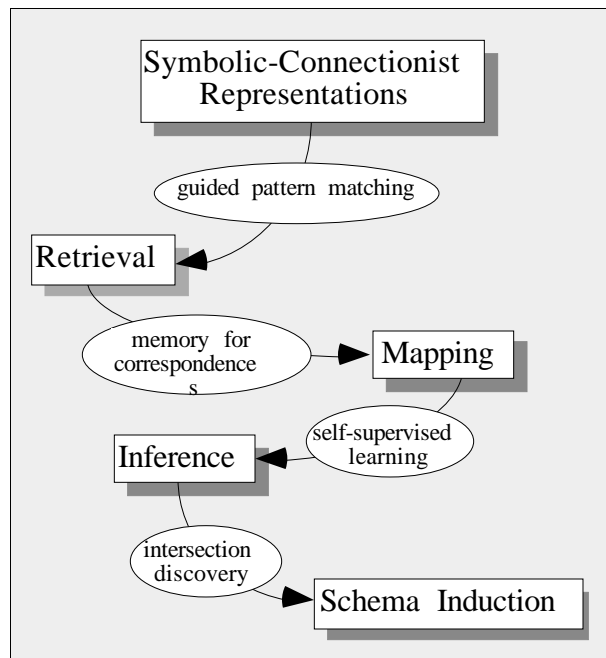
Figure 6. Illustration of our core claims about the relationship between analogical access, mapping, inference and schema induction. Starting with distributed symbolic-connectionist representations (Figures 1 – 3), analog *retrieval* (or *access*) is accomplished as a form of guided pattern matching. Augmented with a basis for remembering which units in the driver were active in synchrony with which units in the source (i.e., by a form of Hebbian learning), the retrieval algorithm becomes a mapping algorithm. Augmented with a simple basis for self-supervised learning, the mapping algorithm becomes an algorithm for analogical inference. And augmented with a basis for intersection discovery, the inference algorithm becomes an algorithm for schema induction.

Flow of Control in LISA

There is no necessary linkage between the driver/recipient distinction and the more familiar source/target distinction. However, a canonical flow of control involves initially using the target as the driver to access a familiar source analog in long-term memory. Once a source is in active memory, mapping can be performed in either direction (see Hummel & Holyoak, 1997). After a

stable mapping is established, the source is used to drive both the generation of inferences in the target and schema induction.

When directed to set its own firing order, LISA randomly chooses propositions in the driver to fire (although it is still possible, and often convenient, to set the firing order by hand). The probability that a proposition will be chosen to fire is proportional to the proposition's importance and its *support* from other propositions, and is inversely proportional to the recency with which it fired. Support relations (defined quantitatively below) allow propositions to influence the probability with which one another fire. To the extent that proposition i supports proposition j, then when i fires, it increases the probability that j will fire in the near future. The role of text coherence in firing order is captured by support relations that reflect the order in which propositions would be stated in a text (i.e., the first proposition supports the second, the second supports the third, etc.). The role of higher-order relations (in both firing order and grouping of multiple proposition in WM) is captured by support relations between higher-order propositions and their arguments, and between propositions that are arguments of a common higher-order relation. The role of shared predicates and arguments is captured by support relations between propositions that share predicates and/or arguments.

Whereas support relations affect the conditional probability that proposition i will be selected to fire given that j has fired, *pragmatic centrality*—a proposition's importance in the context of the reasoner's goals (Holyoak & Thagard, 1989; Spellman & Holyoak, 1996)—is captured in the base rate of i's firing: Pragmatically important propositions have a higher base rate of firing than less important propositions, so they will tend to fire earlier and more often than less important propositions (Hummel & Holyoak, 1997). When LISA is directed to set its own firing order the probability that P unit i will be chosen to fire, is given by the Luce choice axiom:

$$p_i = \frac{c_i}{\sum_j c_j},\tag{1}$$

where p_i is the probability that i will be chosen to fire, c_i is a function of i's support from other propositions (s_i), its pragmatic importance (m_i), and its "readiness" to fire (r_i), which is proportional to the length of time that has elapsed since it last fired):

$$c_i = r_i(s_i + m_i).\tag{2}$$

A proposition's support from other propositions at time t is:

$$s_i(t) = s_i(t-1) - \delta^s s_i(t-1) + \sum_j \sigma_{ij} f_j(t-1),\tag{3}$$

where δ^s is a decay rate, σ_{ij} is the strength of support from j to i (σ_{ij} - …), and $f_j(t-1)$ is 1 if unit j fired at time $t-1$ (i.e., j was among the P units last selected to fire) and zero otherwise. If i is selected to fire at time t, then $s_i(t)$ is initialized to zero (becausew the unit has fired, satisfying its previous support); otherwise it decays by half its value on each iteration until i fires (i.,e., $\delta^s = 0.5$). A P unit's readiness at time t is:

$$r_i(t) = \begin{cases} r_i(t-1) + \gamma^r & \text{if } i \text{ did not fire at time } t \\ \dfrac{r_i(t-1)}{2} & \text{if } i \text{ did fire at time } t. \end{cases}\tag{4}$$

Together, Eq. (1) – (4) make a proposition more likely to fire to the extent that it is either pragmatically central or supported by other propositions that have recently fired, and less likely to fire to the extent that it has recently fired.

Analog Retrieval

In the course of retrieving a relevant source analog from LTM, propositions in the driver activate propositions in other analogs (as described below), but the model does not keep track of which propositions activate which. (Given the number of potential matches in LTM, doing so would require it to form and update an unreasonable number of mapping connections in WM.) Hummel and Holyoak (1997) show that this single difference between access and mapping allows LISA to simulate observed differences in human sensitivity to various types of similarity in these two processes (e.g., Ross, 1989). The current paper is concerned only with analogical mapping and the subsequent processes of inference and schema induction. In the discussions and simulations that follow, we assume that access has already taken place, so that both the driver and recipient analogs are available in active memory.

Establishing Dynamic Role-Filler Bindings: The Operation of SPs

Each SP unit, *i*, is associated with an inhibitory unit, I_i, which causes its activity to oscillate. When the SP becomes active, it excites the inhibitor, which after a delay, inhibits the SP to inactivity. While the SP is inactive, the inhibitor's activity decays, allowing the SP to become active again. In response to a fixed excitatory input, an SP's activation will therefore oscillate between 0 and (approximately) 1. (See Appendix A for details.) Each SP excites the role and filler to which it is connected and strongly inhibits other SPs. The result is that roles and fillers that are bound together oscillate in synchrony with one another, and separate role-filler bindings oscillate out of synchrony. SPs also excite P units "above" themselves, so a P unit will tend to be active whenever any of its SPs is active. Activity in the driver starts in the SPs of the propositions in the phase set. Each SP in the phase set receives an external input of 1.0, which we assume corresponds to attention focused on the SP. (The external input is directed to the SPs, rather than the P units, to permit the SPs to "time-share", rather than one proposition's SPs dominating all the others, when multiple propositions enter WM together.)

The Operation of the Recipient

Due to the within-SP excitation and between-SP inhibition in the driver, the proposition(s) in the phase set generate a collection of distributed, mutually desynchronized patterns of activation on the semantic units, one for each role-filler binding. In the recipient analog(s), object and predicate units compete to respond to these patterns. In typical winner-take-all fashion, units whose connections to the semantic units best fit the pattern of activation on those units tend to win the competition. The object and predicate units excite the SPs to which they are connected, which excite the P units to which they are connected. Units in the recipient inhibit other units of the same type (e.g., objects inhibit other objects, etc.) After all the SPs in the phase set have had the opportunity to fire once, P units in the recipient are permitted to feed excitation back down to their SPs, and SPs feed excitation down to their predicate and argument units.[4] As a result, the recipient acts like a winner-take-all network, in which the units that collectively best fit the synchronized patterns on the semantic units become maximally active, driving all competing units to inactivity (Hummel & Holyoak, 1997). All token units update their activations as:

_____

[4]If units in the recipient are permitted to feed activation downward before all the SPs in the phase set have had the opportunity to fire, then the recipient can fall into a local energy minimum based on the first SPs to fire. For example, imagine that P1 in the recipient best fits the active driver proposition overall, but P2 happens to be a slightly better match based on the first SP to fire. If the recipient is allowed to feed activation downward based on the first SP to fire, it may commit incorrectly to P2 before the second driver SP has had the opportunity to fire. Suspending the top-down flow of activation in the recipient until all driver SPs have fired helps to prevent this kind of premature commitment (Hummel & Holyoak, 1997).

$$a_i = \gamma n_i (1 - a_i) - \delta a_i, \qquad\qquad (5)$$

where γ is a growth rate, n_i is the net (i.e., excitatory plus inhibitory) input to unit i, and δ is a decay rate. Equation (5) embodies a standard "leaky integrator" activation function.

Mapping Connections

Any recipient unit whose activation crosses a threshold at any time during the phase set is treated as "retrieved" into WM for the purposes of analogical mapping. When unit i in the recipient is first retrieved into WM, *mapping hypotheses*, h_{ij}, are generated for it and every selected unit, j, of the same type in the driver (i.e., P units have mapping hypotheses for P units, SPs for SPs, etc.).[5] Units in one analog have mapping connections only to units of the same type in other analogs—e.g., object units do not entertain mapping hypotheses to SP units. In the following, this type-restriction is assumed: When we refer to the mapping connections from unit j in one analog to all units i in another, we mean all i *of the same type* as j. Mapping hypotheses are units that accumulate and compare evidence that a given unit, j, in the driver maps to a given unit, i, in the recipient. We assume that the mapping hypotheses, like the mapping connection weights they will ultimately inform, are not literally realized as synaptic weights, but rather correspond to neurons in prefrontal cortex with rapidly modifiable synapses (e.g., Asaad, Rainer & Miller, 1998; Fuster, 1997). For example, the activations of these neurons might gate the communication between the neurons whose mapping they represent.

The mapping hypothesis, h_{ij}, that unit j in the driver maps to unit i in the recipient grows according to the simple Hebbian rule:

$$h_{ij} = a_i a_j, \qquad\qquad (6)$$

where a_i and a_j are the activations of units i and j. That is, the strength of the hypothesis that j maps to i grows as a function of the extent to which i and j are active simultaneously.

At the end of the phase set, each mapping weight, w_{ij}, is updated competitively based on the value of the corresponding hypothesis, h_{ij}, and based on the values of other hypotheses, h_{ik}, referring to the same recipient unit, i (where k are units other than unit j in the driver), and hypotheses, h_{lj}, referring to the same driver unit, j (where l are units other than unit i in the recipient). Specifically, the value of a weight goes up in proportion to the value of the corresponding mapping hypothesis, and goes down in proportion to the value of other hypothesis in the same "row" and "column" of the mapping connection matrix. This kind of competitive weight updating serves to enforce the 1:1 mapping constraint (i.e., the constraint that each unit in the source analog maps to no more than one unit in the recipient; Falkenhainer, Forbus & Gentner, 1989; Holyoak & Thagard, 1989).

In contrast to the algorithm described by Hummel and Holyoak (1997), mapping connection weights in the current version of LISA are constrained to take values between 0 and 1. That is, there are no inhibitory mapping connections in the current version of LISA. In lieu of inhibitory mapping connections, each unit in the driver transmits a global inhibitory input to all recipient units. The value of the inhibitory signal is proportional to the driver unit's activation, and to the value of the largest mapping weight leading out of that unit; the effect of the inhibitory signal on any given recipient unit is also proportional to the maximum mapping weight leading into that unit (see Eq. 10 in Appendix A). This global inhibitory signal (see also Wang, Buhmann & von der Malsburg, 1992) has exactly the same effect as inhibitory mapping connections (as in the 1997

---

[5] If a connection already exists between a driver unit and a recipient unit, then that connection is simply used again; a new connection is not created.

version of LISA), but is more "economical" in the sense that it does not require every driver unit to have a mapping connection to every recipient unit.

After the mapping hypotheses are used to update the weights on the mapping connections they (the hypotheses) are initialized to zero. Mapping weights are thus assumed to reside in active memory (long-term working memory), but mapping hypotheses—which can only be entertained and updated while the driver units to which they refer are active—must reside in WM. The capacity of WM therefore sharply limits the number of mapping hypotheses it is possible to entertain simultaneously. In turn, this limits the formal complexity of the structure mappings LISA is capable of discovering (Hummel & Holyoak, 1997; Kubose et al., in press).

Working Memory and Strategic Processing

Recall the original jealousy example from the Introduction, which requires the reasoner to map males to females and vice versa, and recall that finding the correct mapping requires LISA to put two propositions into the same phase set. (See Figure 7 for an illustration of placing separate propositions into separate phase sets [Figure 7a] vs. placing multiple propositions into the same phase set [Figure 7b].) Using a similarity-judgment task that required participants to align structurally complex objects (such as pictures of "butterflies" with many parts), Medin, Goldstone and Gentner (1994) showed that people place greater emphasis on object features than on inter-feature relations early in mapping (and similarity judgment), but greater emphasis on relations than on surface features later in mapping (and similarity judgment). More generally, as emphasis on relations increased, emphasis on features decreased, and vice versa. Accordingly, whenever LISA places multiple propositions into WM, it ignores the semantic features of their arguments. This convention is based on the assumption, noted previously, that people tend by default to think about only a single proposition at a time. In general, the only reason to place multiple propositions into WM simultaneously is to permit higher-order (i.e., inter-propositional) structural constraints to guide the mapping: That is, placing multiple propositions into WM together represents a case of extreme emphasis on relations over surface features in mapping.[6] (One limitation of the present implementation is that the user must tell LISA when to put multiple propositions together in the phase set, although the model can select for itself *which* propositions to consider jointly.)

When LISA places *loves* (Bill, Mary) and *loves* (Mary, Sam) into the same phase set and ignores the semantic differences between objects, it correctly maps Bill to Sally, Mary to Tom and Sam to Cathy. It does so as follows (see Table 2). When *lover+Bill* fires in the driver, it activates *lover+Sally* and *lover+Tom* equally, strengthening the mapping hypotheses for $lover^D \rightarrow lover^R$, *Bill→Sally* and *Bill→Tom* (Table 2, row 1). (The superscripted *D*s and *R*s on the names of units indicate that the unit belongs to the driver or recipient, respectively.) When *beloved+Mary* fires, it activates *beloved+Tom* and *beloved+Cathy*, strengthening the hypotheses for $beloved^D \rightarrow beloved^R$, *Mary→Tom* and *Mary→Cathy* (Table 2, row 2). At this point, LISA knows that Bill maps to either Sally or Tom, and Mary maps to either Tom or Cathy. When *lover+Mary* fires, it activates *lover+Sally* and *lover+Tom*, strengthening *Mary→Sally* and *Mary→Tom* (as well as $lover^D \rightarrow lover^R$) (Table 2, row 3). Finally, when *Sam+beloved* fires, it activates *Tom+beloved* and *Cathy+beloved*., strengthening *Sam→Tom* and *Sam→Cathy* (and $lover^D \rightarrow lover^R$) (Table 2, row 4). At this point, the strongest hypothesis (aside from $lover^D \rightarrow lover^R$ and $beloved^D \rightarrow beloved^R$) is *Mary→Tom*, which has a strength of 2; all other object hypotheses have strength 1 (Table 2, row

_____

[6] Ignoring semantic differences between objects entirely in this situation is doubtless oversimplified. It is probably more realistic to assume that emphasis on object semantics and structural constraints increase and decrease in a graded fashion. However, implementing graded emphasis would introduce additional complexity (and free parameters) to the algorithm without providing much by way of additional functionality.

5). When the mapping weights are updated at the end of the phase set, *Mary   Tom* therefore takes a positive mapping weight and the remainder of the object mapping weights remain at zero.
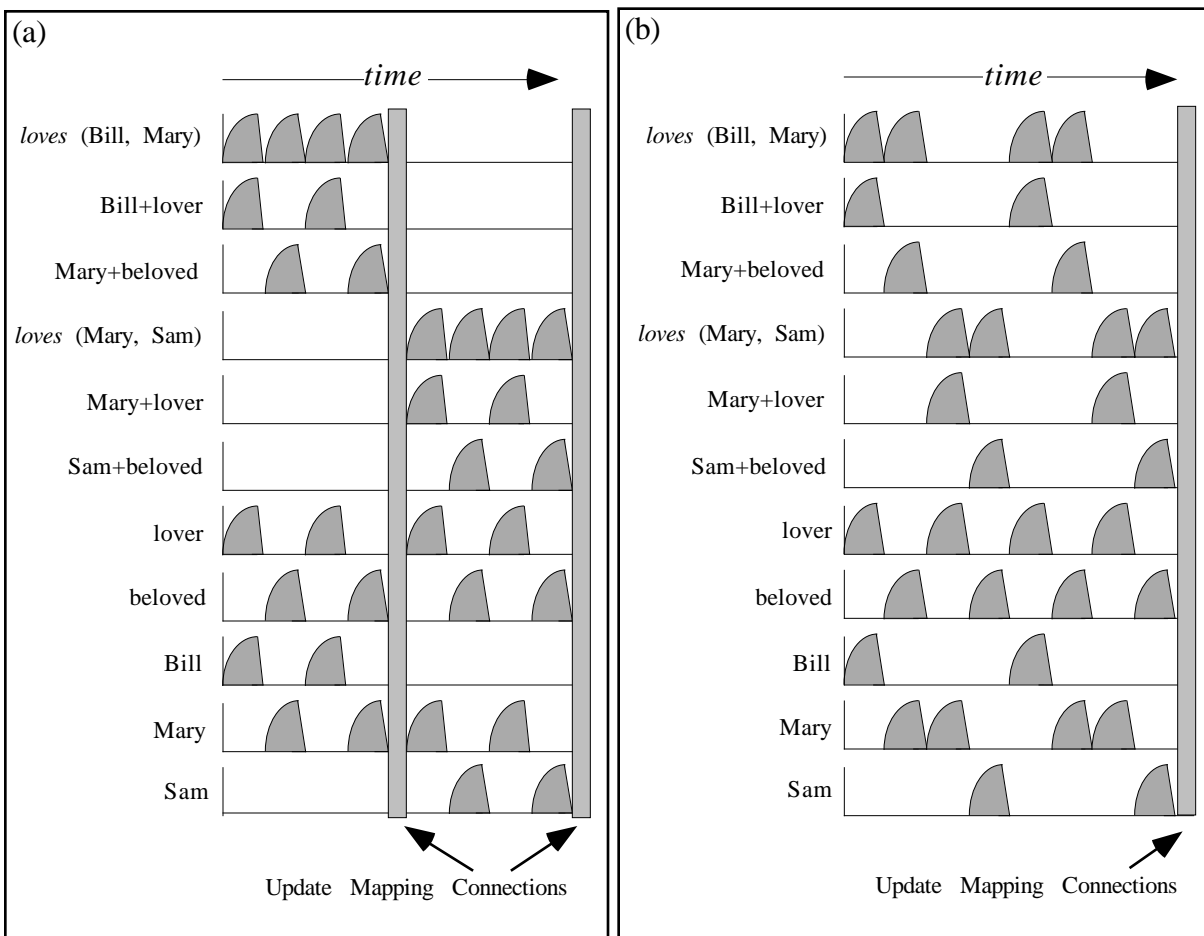

Figure 7. (a) Two propositions, *loves* (Bill, Mary) and *loves* (Mary, Sam), placed into separate phase sets. SPs belonging to one proposition are not temporally interleaved with SPs belonging to the other, and the mapping connections are updated after each proposition fires. (b) The same two propositions placed into the same phase set. The propositions' SPs are temporally interleaved, and the mapping connections are only updated after both propositions have finished firing.

Table 2

Driver SP	Bill→Sally		Bill→Tom		Mary→Sally		Mary→Tom		Mary→Cathy		Sam→Tom		Sam→Cathy	
	/	h	/	h	/	h	/	h	/	h	/	h	/	h
Bill+lover	1	1	1	1	0	0	0	0	0	0	0	0	0	0
Mary+beloved	0	1	0	1	0	0	1	1	1	1	0	0	0	0
Mary+lover	0	1	0	1	1	1	1	2	0	1	0	0	0	0
Sam+beloved	0	1	0	1	0	1	0	2	0	1	1	1	1	1
Total:	**1**		**1**		**1**		**2**		**1**		**1**		**1**	

Note: "   " Indicates the change in the value of the mapping hypothesis due to the active driver SP. "*h*" indicates cumulative the value of the mapping hypothesis.

Once LISA knows that Mary maps to Tom the discovery of the remaining mappings is straightforward, and does not require LISA to place more than one proposition into WM at a time. When LISA fires *loves* (Bill, Mary), the mapping connection from Mary to Tom causes *loves* (Bill, Mary) to activate *loves* (Sally, Tom) rather than *loves* (Tom, Cathy), establishing the mapping of

Bill to Sally (along with the SPs and P units of the corresponding propositions). Analogous operations map Sam to Cathy when *loves* (Mary, Sam) fires. This example illustrates how a mapping connection established at one time (here, the connection between Mary and Tom, established when the two driver propositions were in the phase set together) can influence the mappings LISA discovers subsequently.

Analogical Inference

The correspondences that develop as a result of analogical mapping serve as a useful basis for initiating analogical inference. LISA suspends analogical inference until a high proportion (in simulations reported here, at least 70%) of the importance-weighted structures (propositions, predicates and objects) in the target map to structures in the source. This constraint, which is admittedly simplified,[7] is motivated by the intuitive observation that the reasoner should be unwilling to make inferences from a source to a target until there is evidence that the two are meaningfully analogous.

If the basic criterion for mapping success is met (or if the user directs LISA to license analogical inference regardless of whether it is met), then any unmapped unit in the source (driver) becomes a candidate for inferring a new unit in the target (recipient). LISA uses the mapping-based global inhibition as a cue for detecting when a unit in the driver is unmapped. Recall that if unit i in the recipient maps to some unit j in the driver, then i will be globally inhibited by all other driver units $k \neq j$. Therefore, if some driver unit, k, maps to nothing in the recipient, then when it fires, it will inhibit all recipient units that are mapped to other driver units and it will not excite any recipient units. The resulting inhibitory input, unaccompanied by excitatory input, is a reliable cue that nothing in the recipient corresponds to driver unit k. It therefore triggers the recipient to *infer* a unit to correspond to k. (Naturally, we do not assume the brain grows a new neuron to correspond to k. Instead, we assume the element corresponding to k is generated by the same process that generates structure units at the time of encoding, as discussed previously.) Units inferred in this manner connect themselves to other units in the recipient by simple Hebbian learning: Newly-inferred P units learn connections to active SPs; newly inferred SPs learn connections to active predicate and object units; and newly inferred predicate and object units learn connections to active semantic units (the details of the learning algorithm are given in Appendix A, Eq. A15 and A16 and the accompanying text). Together, these operations—inference of new units in response to unmatched global inhibition, and Hebbian learning of connections to inferred units—allow LISA to infer new objects, new relations among objects, and new propositions about existing and new relations and objects.

To illustrate, let us return to the familiar jealousy example (Figure 2). Once LISA has mapped Bill to Sally, Mary to Tom and Sam to Cathy (along with the corresponding predicates, SPs and P units), every unit in the target (i.e., the analog about Sally) will be mapped to some unit in the source. The target therefore satisfies LISA's 70% criterion for analogical inference. When the proposition *jealous* (Bill, Sam) fires, Bill and Sam will activate Sally and Cathy, respectively, via their mapping connections. However, *jealous1* (the agent role of *jealous*) and *jealous2* (the patient role) map to nothing in the recipient; nor do the SPs *jealous1+Bill* or *jealous2+Sam* or the P unit for the proposition as a whole (*P3*). As a result, when *jealous1+Bill* fires, the predicate unit *jealous1* will inhibit all predicate units in the recipient; *jealous1+Bill* will inhibit all the recipient's SPs, and P3 will inhibit all its P units. In response, the recipient will infer a predicate to correspond to *jealous1*, an SP to correspond to *jealous1+Bill* and a P unit to correspond to *P3*. By convention,

_____

[7] As currently implemented, the constraint is deterministic, but it could easily be made stochastic. We view the stochastic algorithm as more realistic, but for simplicity we left it deterministic for the simulations reported here. A more important simplification is that the algorithm considers only the mapping status of the target analog as a whole. A more realistic algorithm would arguably generate inferences based on the mapping status of subsets of the target analog—for example, as defined by higher-order relations (Gentner, 1983).

we will refer to inferred units using an asterisk, followed by the name of the driver unit that signaled the recipient to infer it, or in the case of inferred SPs, an asterisk followed by the names of the its role and filler units. Thus, the inferred units in the recipient are *jealous1*, *jealous1+Sally*, and *P3*. Because *jealous1*, *jealous1+Bill* and *Bill* are all firing in synchrony in the driver—and because *Bill* excites *Sally* directly via their mapping connection—*jealous1*, *jealous1+Sally* and *Sally* will all be firing in synchrony with one another in the recipient. This coactivity causes LISA to learn excitatory connections between *jealous1* and *jealous1+Sally*, and between *Sally* and *jealous1+Sally*. *Jealous1* also learns connections to active semantic units, i.e., those representing *jealous1*. Similarly, *P3* will be active when *jealous1+Sally* is active, so LISA will learn an excitatory connection between them. Analogous operations cause the recipient to infer *jealous2* and *jealous2+Cathy*, and to connect them appropriately. That is, in response to *jealous* (Bill, Sam) in the driver, the recipient has inferred *jealous* (Sally, Cathy).

This kind of learning is neither fully supervised, like back-propagation learning (Rumelhart, Hinton & Williams, 1986), in which exquisitely-detailed error correction by an external teacher driver drives learning, nor fully unsupervised (e.g., Kohonen, 1982; Marshall, 1995; von der Malsburg, 1973), in which learning is driven exclusively by the statistics of the input patterns. Rather, it is best described as *self-supervised*, because the decision about whether to license an analogical inference, and if so what inference to make, is based on the system's knowledge about the mapping between the driver and recipient. Thus, because LISA knows that Bill is jealous of Sam, and because it knows that Bill maps to Sally and Sam to Cathy, it infers *relationally* that Sally will be jealous of Cathy. This kind of self-supervised learning permits LISA to make inferences that lie well beyond the capabilities of traditional supervised and unsupervised learning algorithms.

Schema Induction

The same principles, augmented with the capacity to discover which elements are shared by the source and target, allow LISA to induce general schemas from specific examples (Hummel & Holyoak, 1996). In the case of the current example, the schema to be induced is roughly, "If person *x* loves person *y*, and *y* loves *z*, then *x* will be jealous of *z*." LISA induces this schema as follows. The schema starts out as an analog containing no token units. Tokens are created and interconnected in the schema according to exactly the same rules in the target: If nothing in the schema maps to the currently-active driver units, then units are created to map to them. Since the schema is initially devoid of tokens, they are created every time a token fires for the first time in the driver.

The behavior of the semantic units plays a central role in LISA's schema induction algorithm. In contrast to token units, semantic units represent objects and roles in a distributed fashion, so it is neither necessary nor desirable for them to inhibit one another. Instead, to keep their activations bounded between 0 and 1, semantic units normalize their inputs divisively:

$$a_i = \frac{n_i}{\max(\max(|n|),1)}, \qquad (7)$$

where n_i is the net input to semantic unit *i*, and *n* in the net input to any semantic unit. That is, a semantic unit takes as its activation its own input divided by the maximum of the absolute value of all inputs or 1, whichever is greater. Albrecht and Geisler (1991), Bonds (1989) and Heeger (1992) present physiological evidence for broadband divisive normalization in the visual system of the cat, and Foley (1994) and Thomas and Olzak (1997) present evidence for divisive normalization in human vision.

Because of Eq. 7, a semantic unit's activation is a linear function of its input (for any fixed value in the denominator). During mapping, units in the recipient feed activation back to the semantic units, so semantic units shared by objects and predicates in both the driver and recipient receive about twice as much input—and therefore become about twice as active—as semantic units unique to one or the other. As a result, object and predicate units in the schema learn stronger connections to shared semantic units than unshared ones. That is, the schema performs intersection discovery on the semantics of the objects and relations in the examples from which it is induced. In

the case of the jealousy example, the induced schema is "If person x loves person y, and y loves z, then x will be jealous of z," where x, y and z are all connected strongly to semantic units for "person", but are connected more weakly (and approximately equally) to units for "male" and "female": They are "generic" people. (See Appendix B for detailed simulation results.) This algorithm also applies a kind of intersection discovery at the level of complete propositions. If LISA fails to fire a given proposition in the driver (e.g., because its pragmatic importance is low), then that proposition will never have the opportunity to map to the recipient. As a result, it will neither drive inference in the recipient nor drive the generation of a corresponding proposition in the emerging schema. In this way, LISA avoids inferring many incoherent or trivial details in the recipient and it avoids incorporating such details into the emerging schema.

LISA's schema induction algorithm can be applied iteratively, mapping new examples (or other schemas) onto existing schemas, thereby inducing progressively more abstract schemas. Because the algorithm learns weaker connections to unique semantics than to shared semantics but does not discard unique semantics altogether, objects and predicates in more abstract schemas will tend to learn connections to semantic units with weights proportional to the frequency with which those semantics are present in the examples. For example, if LISA encountered ten love triangles, of which eight had a female in the central role—i.e., y in $loves$ (x, y) and $loves$ (y, z)—with males in the x and z roles, then the object unit for y in the final schema induced would connect almost as strongly to $female$ as to $person$; and x and z would likewise be biased toward being male.

Simulations of Human Relational Inference and Generalization

We will now illustrate LISA's ability to simulate a range of empirical phenomena concerning human relational inference and generalization. Table 3 provides a summary of 15 phenomena that we take to be sufficiently well-established that any proposed theory will be required to account for them. We divide the phenomena into those related to the generation of specific inferences about a target, those related to the induction of more abstract relational schemas, and those that involve interactions between schemas and inference. In LISA, as in humans, these phenomena are highly interconnected. Inferences can be made based on prior knowledge (representations of either specific cases or more general schemas); schemas can be induced by bottom-up learning from comparison of examples; and existing schemas can modulate the encoding of further examples, thereby exerting top-down influence on subsequent schema induction.

We make no claim that these phenomena form an exhaustive set. Our focus is on analogical inference as used in comprehension and problem solving, rather than more general rule-based reasoning, or phenomena that may be specific to language (although the core principles motivating LISA may well have implications for these related topics). Following previous analyses of the subprocesses of analogical thinking (Holyoak et al., 1994), we treat aspects of inference that can be viewed as pattern completion, but exclude subsequent evaluation and adaptation of inferences (as the latter processes make open-ended use of knowledge external to the given analogs). We focus on schema-related phenomena that we believe require use of relational working memory (reflective reasoning), rather than phenomena that likely are more automatic (reflexive reasoning; see Holyoak & Spellman, 1993; Shastri & Ajjanagadde, 1993). Our central concern is with schemas that clearly involve relations between elements, rather than (possibly) simpler schemas or prototypes based solely on features of a single object (as have been discussed in connection with experimental studies of categorization; e.g., Posner & Keele, 1968). Possible extensions of LISA to a broader range of phenomena, including reflexive use of schemas during comprehension, will be sketched in the *General Discussion*.

Table 3

Empirical Phenomena Associated With Analogical Inference and Schema Induction

I. Inference

1. Inferences can be made from a single source example.
2. Inferences (a) preserve known mappings; (b) extend mappings by creating new target elements to match unmapped source elements; (c) are blocked if critical elements are unmapped; and (d) yield erroneous inferences if elements are mismapped.
3. Increasing confidence in the mapping from source to target (e.g., by increasing similarity of mapped elements) increases confidence in inferences from source to target.
4. An unmapped higher-order relation in the source, linking a mapped with an unmapped proposition, increases the propensity to make an inference based on the unmapped proposition.
5. Pragmatically important inferences (e.g., those based on goal-relevant causal relations) receive preference.
6. When one-to-many mappings are possible, multiple (possibly contradictory) inferences are sometimes made (by different individuals, or a single individual).
7. Nonetheless, even when people report multiple mappings of elements, each individual inference is based on mutually-consistent mappings.

II. Schema Induction

8. Schemas preserve the relationally-constrained intersection of specific analogs.
9. Schemas can be formed from as few as two examples.
10. Schemas can be formed as a side effect of analogical transfer from one source analog to a target analog.
11. Different schemas can be formed from the same examples, depending on which relations receive attention.
12. Schemas can be further generalized with increasing numbers of diverse examples.

III. Interaction of Schemas With Inference

13. Schemas evoked during comprehension can influence mapping and inference.
14. Schemas can enable universal generalization (inferences to elements that lack featural overlap with training examples).
15. Schemas can yield inferences about dependencies between role fillers.

Analogical Inference

     Basic phenomena. Phenomena 1 and 2 are the most basic characteristics of analogical inference (see Holyoak & Thagard, 1995, for a review of evidence). Inferences can be made using a single source analog (Phenomenon 1) by using and extending the mapping between source and target analogs (Phenomenon 2). The basic form of analogical inference has been called "copy with substitution and generation", or CWSG (Holyoak et al., 1994; also see Falkenhainer et al., 1989), and involves constructing target analogs of unmapped source propositions by substituting the corresponding target element, if known, for each source element (Phenomenon 2a); and if no corresponding target element exists, postulating one as needed (Phenomenon 2b). This procedure gives rise to two important corollaries concerning inference errors. First, if critical elements are

difficult to map (e.g., because of strong representational asymmetries, such as those that hinder mapping a discrete set of elements to a continuous variable; Bassok & Olseth, 1995; Bassok & Holyoak, 1989), then no inferences can be constructed (Phenomenon 2c). Second, if elements are mismapped, predictable inference errors will result (Phenomenon 2d; Reed, 1987; see Holyoak et al., 1994).

All major computational models of analogical inference use some variant of CWSG (e.g., Falkenhainer et al., 1989; Halford et al., 1994; Hofstadter & Mitchell, 1994; Holyoak et al., 1994; Keane & Brayshaw, 1988; Kokinov, 1994), and so does LISA. CWSG is critically dependent on variable binding and mapping; hence models that lack these key elements (e.g., those based on back-propagation learning) fail to capture even the most basic aspects of analogical inference. LISA's ability to account for Phenomena 1 and 2 is apparent in its performance on the "jealousy" problem illustrated previously.

As another example, consider a problem adapted from St. John (1992). The source analog states that Bill has a Jeep, Bill wanted to go to the beach, so he got into his Jeep and drove to the beach; the target states that John has a Civic, and John wanted to go to the airport. The goal is to infer that John will drive to the airport in his Civic. This problem is trivial for people to solve, but it poses a serious difficulty for models that cannot dynamically bind roles to their fillers. This difficulty is illustrated by the performance of a particularly sophisticated example of such a model, the Story Gestalt model of story comprehension developed by St. John (1992; St. John & McClelland, 1990). In one computational experiment (St. John, 1992, Simulation 1), the Story Gestalt model was first trained with 1,000,000 short texts consisting of propositions based on 136 different constituent concepts. Each story instantiated a script such as "<person> decided to go to <destination>; <person> drove <vehicle> to <destination>" (e.g., "George decided to go to a restaurant; George drove a jeep to the restaurant"; "Harry decided to go to the beach; Harry drove a Mercedes to the beach").

After the model had learned a network of associative connections based on the 1,000,000 examples, St. John tested its ability to generalize by presenting it with a text containing a new proposition, such as "John decided to go to the airport." Although the proposition as a whole was new, it referred to people, objects and places that had appeared in other propositions used for training. St. John reported that when given a new proposition about deciding to go to the airport, the model would typically activate the restaurant or the beach (i.e., the destinations in prior examples of the same script) as the destination, rather than making the contextually appropriate inference that the person would drive to the airport. This type of error, which would appear quite unnatural in human comprehension, results from the model's inability to learn generalized linking relationships (e.g., that if a person wants to go location x then x will be the person's destination—a problem that requires the system to represent the variable x as well as its value, independently of its binding to the role of *desired location* or *destination*). As St. John noted, "Developing a representation to handle role binding proved to be difficult for the model" (1992, p. 294).

LISA solves this problem using self-supervised learning (see Appendix B for the details of this and other simulations). As summarized in Table 4, the source analog states that Bill has a Jeep (P1); Bill wanted (P3) to go to the beach (P2); therefore (P5), Bill drove his Jeep to the beach (P4). The target states that John has a Civic (P1) and John wanted (P3) to go to the airport (P2). When P2, *goto* (Bill, beach), fires in the driver, it maps to P2, *goto* (John, airport), in the recipient, mapping Bill to John, beach to airport, and the roles of *goto* to their corresponding roles in the recipient. Likewise, *has* (Bill, Jeep) in the driver maps to *has* (John, Civic), and *want* (Bill, P2) maps to *want* (John, P2). However, when *drive-to* (Bill, beach) fires, only *Bill* and *beach* map to units in the recipient; the P unit (*P4*), its SPs, and the predicate units *drive-to1*, and *drive-to2* do not correspond to any existing units in the recipient. Because of the mappings established in the context of the other propositions (P1-P3), every unit in the recipient has a positive mapping connection to *some* unit in the driver, but none map to *P4*, its SPs, or to the predicate units *drive-to1* and *drive-to2*. As a result, these units globally inhibit all P, SP and predicate units in the recipient. In response, the recipient infers new P, SP and predicate units, and connects them together (into a proposition) via Hebbian learning. The resulting units represent the proposition *drive-to* (John, airport). That is, LISA has inferred that John will drive to the airport. Other inferences (e.g., the fact that John's

decision to go to the airport caused him to drive to the airport) are generated by exactly the same kinds of operations. By the end, the recipient is structurally isomorphic with the driver, and represents the idea "John wanted to go to the airport, so he drove his Civic to the airport" (see Appendix B for details).

Table 4

The Airport/Beach Analogy (adapted from St. John, 1992)

Source (Analog 1)	Target (Analog 2)
P1: *has* (Bill, Jeep)	P1: *has* (John, Civic)
P2: *go-to* (Bill, Beach)	P2: *go-to* (John, Airport)
P3: *want* (Bill, P2)	P3: *want* (John, P2)
P4: *drive-to* (Bill, Jeep, Beach)	
P5: *cause* (P3, P4)	

We have run this simulation many times, and the result is always the same: LISA invariably infers that John's desire to go to the airport will cause him to drive his Civic to the airport. As illustrated here, LISA makes these inferences by analogy to a specific episode (i.e., Bill driving to the beach). By contrast, most adult human reasoners would likely make the same inference based on more common knowledge, namely, a schema specifying that when person *x* wishes to go to destination *y*, then *x* will drive to *y* (this is perhaps especially true for residents of Los Angeles). When the "Bill goes to the beach" example is replaced with this schema, LISA again invariably infers that John's desire to go to the airport will cause him to drive to the airport.

Constraints on selection of source propositions for inference. Although all analogy models use some form of CWSG, it has been widely recognized that human analogical inference involves additional constraints (e.g., Clement & Gentner, 1991; Holyoak et al., 1994; Markman, 1997). If CWSG were unconstrained, then *any* unmapped source proposition would generate an inference about the target. Such a loose criterion for inference generation would lead to rampant errors whenever the source was not isomorphic to the target (or a subset of the target); and such isomorphism will virtually never hold for problems of realistic complexity.

Phenomena 3-5 highlight human constraints on the selection of source propositions for generation of inferences about the target. All these phenomena (emphasized in Holyoak & Thagard's, 1989, 1995, multiconstraint theory of analogy) were demonstrated in a study by Lassaline (1996; also see Clement & Gentner, 1991; Spellman & Holyoak, 1996). Lassaline had college students read analogs (mainly describing properties of hypothetical animals) and then rate various possible target inferences for "the probability that the conclusion is true, given the information in the premise" (1996, p. 758). Participants rated potential inferences as more probable when the source and target analogs shared more attributes, and hence mapped more strongly (Phenomenon 3). In addition, their ratings were sensitive to structural and pragmatic constraints. The presence of a higher-order linking relation in the source made an inference more credible (Phenomenon 4). For example, if the source and target animals were both described as having an acute sense of smell, and the source animal was said to have a weak immune system that "develops before" its acute sense of smell, then the inference that the target animal also has a weak immune system would be bolstered relative to stating only that the source animal had an acute sense of smell "and" a weak immune system. The benefit conveyed by the higher-order relation was increased (Phenomenon 5) if the relation was explicitly causal (e.g., in the source animal, a weak immune system "causes" its acute sense of smell), rather than less clearly causal ("develops before").

Two properties of the LISA algorithm permit it to account for these constraints on the selection of source propositions for generating inferences about the target. One is the role of importance and support relations that govern the firing order of propositions. More important propositions in the driver are more likely to be chosen to fire, and are therefore more likely to drive analogical inference. Similarly, propositions with a great deal of support from other propositions—

for example, by virtue of serving as arguments of a higher-order causal relation—are more likely to fire (and therefore to drive inference) than those with little support. The second relevant property of the LISA algorithm is that inference is only attempted if at least 70% of theimportance-weighted structures in the target map to structures in the source (Eq. A15).

We simulated the findings of Lassaline (1996) by generating three versions of her materials, corresponding to her strong, medium and weak constraint conditions. In all three simulations, the source analog stated that there was an animal with an acute sense of smell, and that this animal had a weak immune system; the target stated that there was an animal with an acute sense of smell. Both the source and target also contained three propositions stating additional properties of this animal, plus three propositions stating the properties of an additional distractor animal. The propositions about the distractor animal served to reduce the probability of firing the various propositions about the critical animal.

In the strong constraint simulation, the source stated that the crucial animal's acute sense of smell caused it to have a weak immune system. The propositions stating the acute sense of smell (P1), the weak immune system (P2), and the causal relation between them (P3) were all given an importance value of 5; the other six propositions had importance values of 1 (the default value; henceforth, when importance values are not stated, they are 1). In addition, propositions P1 - P3 mutually supported one another with strength 5. No other propositions supported one another (the default value for support is zero). To simulate Lassaline's (1996) manipulation of the similarity relations among the animals, the critical animals in the source and target shared three properties in addition to their acute sense of smell (properties A, B and C, stated in propositions P4 – P6 in the source, and in propositions P2 – P4 in the target). In the target, all the propositions describing the critical animal (P1 – P4) had importance 5. The remaining propositions (P5 - P7) had importance 1 and stated the properties of the distractor animal.

The medium and weak constraint simulations were identical to the strong constraint simulation with the following changes. In the medium constraint simulation, the causal relation between acute smell and the weak immune system (P3) was replaced by the less constraining relation that the acute sense of smell preceded the weak immune system. The importance of the three statements (P1 - P3) was accordingly reduced from 5 to 2, as was the support between them. In addition, the similarity between the critical animals in the source and target was reduced from three additional features (A, B and C) to two (A and B) by giving feature C to the distractor animal. In the target, the propositions stating properties of the critical animal had importance 2. In the weak constraint simulation, the causal relation between the acute sense of smell and the weak immune system (P3) was removed from the source altogether, and the importance of P1 (stating the acute sense of smell) and P2 (stating the weak immune system) were accordingly reduced to the default value of 1. In the target, features A, B and C (propositions P2 - P4) were assigned to the distractor animal rather than the critical animal, and their importance was dropped to 1. In all other respects (except a small difference in firing order, noted below), both the weak and medium constraint simulations were identical to the strong constraint simulation.

We ran each simulation 50 times. On each run, the target was initially the driver, the source was the recipient, and LISA fired five propositions, randomly chosen according to Eq. 1. Then the source was designated the driver, the target the recipient. In the strong and medium constraint conditions, LISA first fired P1 (the acute sense of smell) in the source, and then fired five randomly-chosen propositions. In the weak constraint condition, the source fired only the five randomly-chosen propositions. This difference reflects our assumption that the acute sense of smell is more salient in the strong and medium conditions than in the weak condition. This kind of "volleying" in which the target is initially the driver, followed by the source, is the canonical flow of control in LISA (Hummel & Holyoak, 1997). As described above, LISA "decided" for itself whether to license analogical inference based on the state of the mapping.

LISA's inferences in the 50 runs of each simulation are summarized in Table 5. In the strong constraint simulation, LISA inferred both that the critical target animal would have a weak immune system and that this condition would be caused by its acute sense of smell on 35 of the 50 runs; it inferred the weak immune system without inferring it would be caused by the acute sense of

smell on two runs; on nine runs it inferred that the acute sense of smell would cause something, but failed to infer that it would cause the weak immune system. On four runs it failed to infer anything.

Table 5

Results of Simulations of Study by Lassaline (1996)

Inference(s)

Simulation	Both[a]	Weak[b]	Cause[c]	Neither[d]
Strong	35	2	9	4
Medium	15	11	13	11
Weak	NA	14	NA	36

    a   Number of simulations (out of 50) on which LISA inferred both the weak immune system and the higher-order *cause* (strong constraint simulation) or *preceed* (medium constraint simulation) relation

    b   Number of simulations on which LISA inferred the weak immune system only

    c   Number of simulations on which LISA inferred the higher-order relation only

    d   Number of simulations on which LISA inferred neither the weak immune system nor the higher-order relation

       LISA made fewer inferences, and notably fewer causal inferences, in the medium constraint simulation. Here, on 15 runs it made both inferences; on 11 it inferred the weak immune system only; on 13 it inferred that the acute sense of smell would precede something, but failed to infer what it would precede, and on 11 runs it made no inferences at all. LISA made the fewest inferences in the low constraint simulation. On 14 runs it inferred the weak immune system (recall that the source in this simulation contains no higher-order relation, so none can be inferred in the target), and on the remaining 36 runs it made no inferences. These simulations demonstrate that LISA, like Lassaline's (1996) subjects, is sensitive to pragmatic (causal relations), structural (higher-order relations) and semantic (shared features) constraints on the selection of source propositions for generation of analogical inferences about the target.

       Inferences with one-to-many mappings. Phenomena 6-8 involve further constraints on analogical inference that become apparent when the analogs suggest potential one-to-many mappings between elements. LISA, like all previous computational models of analogical mapping and inference, favors unique (one-to-one) mappings of elements. The one-to-one constraint is especially important in analogical inference, since inferences derived using a mix of incompatible element mappings are likely to be nonsensical. For example, suppose the source analog "Iraq invaded Kuwait but was driven out" is mapped onto the target "Russia invaded Afghanistan; Argentina invaded the Falkland Islands." If the one-to-one constraint on mapping does not prevail over the positive evidence that Iraq can map to both Russia and Argentina, and Kuwait to both Afghanistan and the Falklands, then the CWSG algorithm might form the target inference, "Russia was driven out of the Falkland Islands." This "inconsistent" inference is not only false, but bizarre, since the target analog gives no reason to suspect that Russia had anything to do with the Falklands.

       Although all analogy models have incorporated the one-to-one constraint into their mapping algorithms, they differ in whether or not the constraint is inviolable. Models such as SME (Falkenhainer et al., 1989) and IAM (Keane & Brayshaw, 1988) compute strict one-to-one mappings; indeed, without this hard constraint, their mapping algorithms would catastrophically fail. In contrast, models based on soft constraint satisfaction, such as ACME (Holyoak & Thagard, 1989) and LISA (Hummel & Holyoak, 1997), treat the one-to-one constraint as a strong preference rather than an inviolable rule. Empirical evidence indicates that when people are asked to map homomorphic analogs, and are encouraged to give as many mappings as they consider appropriate,

they will sometimes report multiple mappings for an element (Markman, 1997; Spellman & Holyoak, 1992, 1996). However, when several mappings form a cooperative set, which is incompatible with a "rival" set of mappings (much like the perceptual Necker cube), an individual reasoner will generally give only one consistent set of mappings; if one set is more relevant to the reasoner's goal, that set will be strongly preferred (Spellman & Holyoak, 1996).

Models with a "soft" one-to-one constraint can readily account for the tendency for people to settle on a single set of consistent mappings (see Spellman & Holyoak, 1996, for simulations using ACME; and Hummel & Holyoak, 1997, for comparable simulations using LISA). Even if the analogy provides equal evidence in support of alternative sets of mappings, randomness in the mapping algorithm (coupled with inhibitory competition between inconsistent mappings) will tend to yield a consistent set of winning mappings. When people are asked to make inferences based on such homomorphic analogs, different people will find different sets of mappings, and hence may generate different (and incompatible) inferences (Phenomenon 6); but any single individual will generate consistent inferences (Phenomenon 7; Spellman & Holyoak, 1996).

LISA provides a straightforward account of these phenomena. LISA can discover multiple mappings for a single object, predicate or proposition, in the sense that it can discover different mappings on different runs with an ambiguous analogy (such as those of Spellman & Holyoak, 1996). However, on any given run, it will always generate inferences that are consistent with the mappings it discovers: Because inference is driven by mapping, LISA is incapable of making an inference that is inconsistent with the mappings it has discovered.

Table 6

LISA's Schematic Propositional Representations of the Relations Between Characters in Experiment 3 of Spellman and Holyoak (1996) (see also Appendix B).

Plot 1	Plot 2
P1 *bosses* (Peter, Mary)	P1 *bosses* (Nancy, John) P3 *bosses* (David, Lisa)
P2 *loves* (Peter, Mary)	P2 *loves* (John, Nancy) P4 *loves* (Lisa, David)
P3 *cheats* (Peter, Bill)	P5 *cheats* (Nancy, David)
	P6 *cheats* (Lisa, John)
P4 *fires* (Peter, Mary)	
P5 *courts* (Peter, Mary)	

To explore LISA's behavior when multiple mappings are possible, we have extended the simulations of the results of Spellman and Holyoak (1996, reported by Hummel & Holyoak, 1997) to include inference as well as mapping. In Spellman and Holyoak's Experiment 3, participants read plot summaries for two elaborate soap operas. Each plot involved various characters interconnected by professional relations (one person was the boss of another), romantic relations (one person loved another), and cheating relations (one person cheated another out of an inheritance) (see Table 6). Each of these relations appeared once in Plot 1 and twice in Plot 2. Plot 1 also contained two additional relations not stated in Plot 2: *courts*, which follows from the *loves* relation, and *fires*, which follows from the *bosses* relation. Based solely on the *bosses* and *loves* relations, Peter and Mary in Plot 1 are four-ways ambiguous in their mapping to characters in Plot 2: Peter could map to any of Nancy, John, David, or Lisa, as could Mary. If the *bosses* relation is emphasized over the *loves* relation (i.e., if a processing goal provides a strong reason to map *bosses* to *bosses* and hence the characters in the *bosses* propositions to one another), then Peter should map to either Nancy or David and Mary would correspondingly map to either John or Lisa. There would be no way to choose between these two alternative mappings (ignoring gender, which in the experiment was controlled by counterbalancing). But if the *cheats* relations are considered, then they provide a basis for selecting unique mappings for Peter and Mary: Peter maps to Nancy so Mary must map to John. Similarly, if the *loves* propositions are emphasized (absent *cheats*), then Peter should map to

John or Lisa and Mary would map to Nancy or David; and if the *cheats* propositions are considered in addition to *loves*, then Peter should map to Lisa and hence Mary to David.

Spellman and Holyoak had participants make plot extensions of Plot 2 based on Plot 1. That is, participants were encouraged to indicate which specific characters in Plot 2 were likely to court or fire one another, based on their mappings to the characters in Plot 1. Spellman and Holyoak manipulated participants' processing goals by encouraging them to emphasize either the *bosses* or the *loves* relations in extending Plot 2 based on Plot 1; the cheating relations were always irrelevant. Participants' preferred mappings were revealed on this plot-extension (i.e., analogical inference) task by their choice of Plot 2 characters to play roles analogous to those played by Peter and Mary in Plot 1. Afterwards, participants were asked directly to provide mappings of Plot 2 characters onto those in Plot 1.

Spellman and Holyoak's (1996) results are presented in the left column of Figure 8, which depicts the percentage of mappings for the pair Peter/Mary that were consistent with respect to the processing goal (collapsing across focus on professional versus romantic relations), inconsistent with the processing goal, or some other response (e.g., a mixed mapping). The goal-consistent and goal-inconsistent mappings are further divided into those consistent versus inconsistent with the irrelevant cheating relation. The plot-extension and mapping tasks were both sensitive to the processing goal, in that participant's preferred mappings in both tasks tended to be consistent with the goal-relevant relation. This effect was more pronounced in the plot-extension than the mapping task. In contrast, consistency with the cheating relation was more pronounced in the mapping than the plot-extension task.

To simulate these findings, we gave LISA schematic representations of the two plots based on the propositions in Table 6 (see Appendix B for details). We generated four levels of pragmatic focus: High emphasis on the professional relation (in which LISA was strongly encouraged to focus on the *bosses* relation over the *loves* relation; *P-High*); low emphasis on the professional relation (in which LISA was weakly encouraged to focus on *bosses* over *loves*; *P-Low*); high emphasis on the romantic relation (in which LISA was strongly encouraged to focus on *loves* over *bosses*; *R-High*); and low emphasis on the romantic relation (in which LISA was weakly encouraged to focus on *loves* over *bosses*; *R-Low*). The high-emphasis simulations (P-High and R-High) were intended to simulate Spellman and Holyoak's plot extension task, and the low-emphasis simulations (P-Low and R-Low) were intended to simulate their mapping task.

In all four simulation conditions, propositions in the source (Plot 1) and target (Plot 2) were assigned support relations reflecting their causal connectedness and shared predicates and arguments. In the source, P1 and P4 supported one another with strength 100 (reflecting their strong causal relation), as did P2 and P5. These support relations were held constant across all simulations, reflecting our assumption that the relations among statements remain relatively invariant under the reasoner's processing goals. The only factor we varied across conditions was the importance values of individual propositions (see Appendix B). The Spellman and Holyoak materials are substantially more complex than the Lassaline (1996) materials, in the sense that the Spellman and Holyoak materials form multiple, mutually-inconsistent sets of mappings. Accordingly, we set the importance and support relations to more extreme values in these simulations in order to strongly bias LISA in favor of discovering coherent sets of mappings. The corresponding theoretical claim is that human reasoners are sensitive to the pragmatic and causal interrelations between elements of analogs, and can use those interrelations to divide complex analogs (such as the Spellman and Holyoak materials) into meaningful subsets. (Actually, the differences in the parameter values are less extreme than they may appear because LISA's behavior is a function of the propositions relative values of support and importance, rather than the absolute values of these parameters; recall Eq. 1 – 4.) In the P-High simulations, *bosses* (Peter, Mary) in Plot 1 was given an importance value of 20, as were *bosses* (Nancy, John) and *bosses* (David, Lisa) in Plot 2; all other propositions in both plots were given the default value of 1. In the P-Low simulation, all the *bosses* propositions (i.e., bosses (Peter, Mary) in the source and the corresponding propositions in the target) had an importance of 2 (rather than 20, as in P-High). In the R-High and R-Low simulations, the importance values were precisely analogous to those in P-

High and P-Low, respectively, except that the emphasis was on the romantic relations rather than the professional ones.
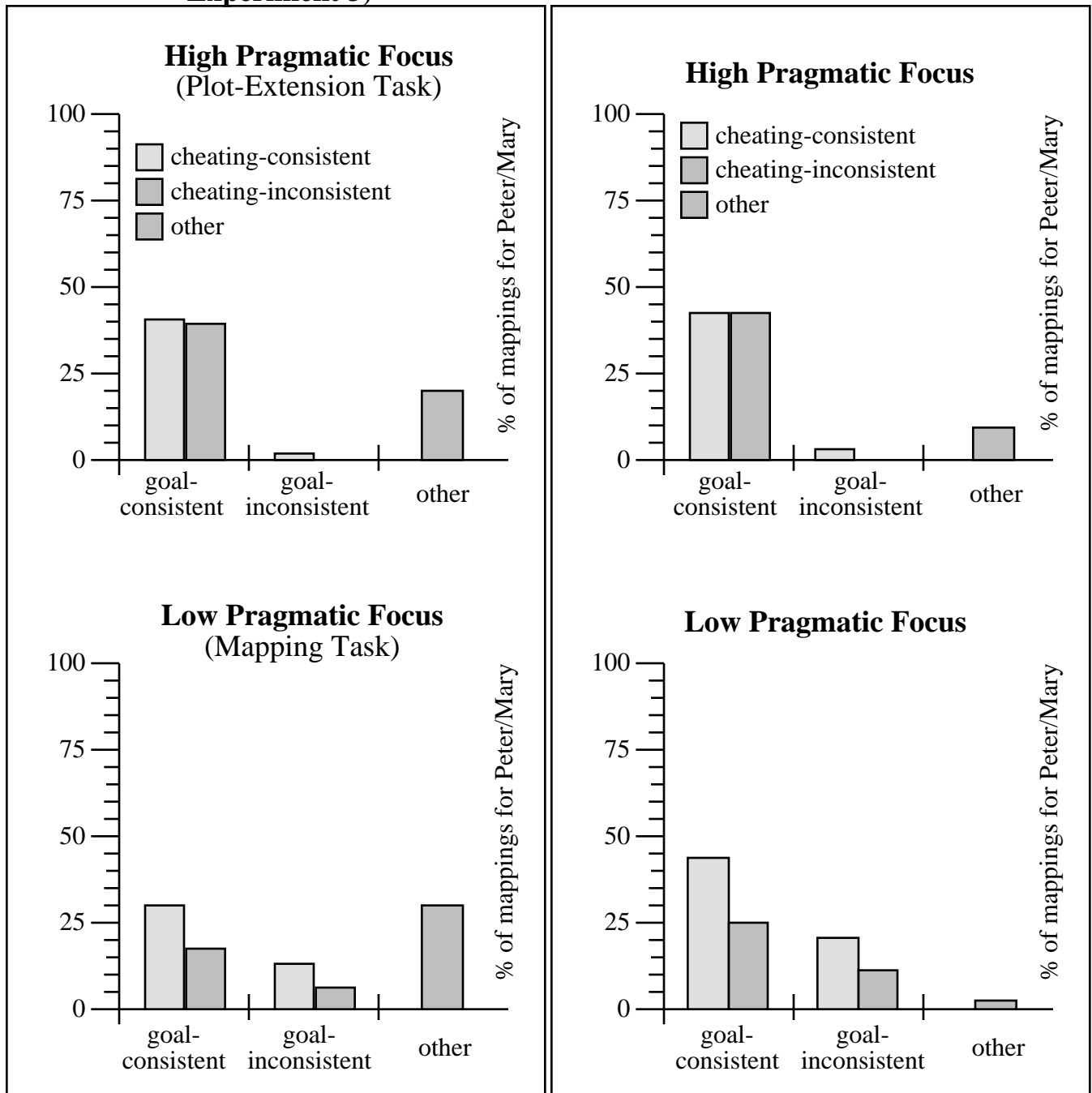

Figure 8. Results of Spellman and Holyoak (1996, Experiment 3) (left column) and simulation results of LISA (right column) in the high-pragmatic-focus (top row) and low-pragmatic focus conditions (bottom row). See text for details.

We ran each simulation 48 times. LISA's mapping performance in the high- and low-focus conditions (summed over romantic and professional relations) is shown in the second column of Figure 8. We did not attempt to produce a detailed quantitative fit between the simulation results and human data, but a close qualitative correspondence is apparent. In the high pragmatic-focus simulations, LISA's Peter/Mary mappings were almost always goal-consistent, but were independent of the irrelevant cheating relation. In the low-focus simulations, LISA was again more likely to produce mappings that were goal-consistent, rather than goal-inconsistent, but this preference was weaker than in the high-focus simulations. On the other hand, in the low-pragmatic focus simulations LISA more often selected mappings that were consistent with the cheating relation. Thus LISA, like people, produced mappings that were dominated by the goal-relevant relation alone when pragmatic focus was high, but that were also influenced by goal-irrelevant relations when pragmatic focus was low. LISA's most notable departure from the human data is that it was less likely to report "other" mappings—i.e., mappings that are inconsistent with the structure of the analogy. We take this departure from the human data to reflect primarily random factors in human performance (e.g., lapses of attention, disinterest in the task) to which LISA is immune.

Importantly, LISA's inferences (i.e., plot extensions) in the high-focus conditions were always consistent with its mappings. In the P-High simulations, if LISA mapped Peter to David and Mary to Lisa, then it always inferred that David would fire Lisa; if it mapped Peter to Nancy and Mary to John, then it inferred that Nancy would fire John. In the R-High simulations, if it mapped Peter to John and Mary to Nancy, then it inferred that John would court Nancy, and if it mapped Peter to Lisa and Mary to David, then it inferred that Lisa would court David.

These results highlight three important properties of LISA's mapping and inference algorithm. First, as illustrated by the results of the high pragmatic-focus simulations, LISA will stochastically produce different mappings, consistent with the goal-relevant relation but less dependent on the goal-irrelevant relations, when the latter are only given a limited opportunity to influence mapping. Second, as illustrated by the low pragmatic-focus simulations, differences in selection priority are sufficient (when no single relation is highly dominant) to bias LISA in one direction or another in resolving ambiguous mappings. And third, even though LISA is capable of finding multiple different mappings in ambiguous situations, its analogical inferences are always consistent with only a single mapping.

Schema Induction

We now consider how LISA accounts for several phenomena related to the induction of relational schemas. There is a great deal of empirical evidence that comparison of multiple analogs can result in the induction of relationally-constrained generalizations, which in turn facilitate subsequent transfer to additional analogs (Phenomenon 8). The induction of such schemas has been demonstrated in both adults (Catrambone & Holyoak, 1989; Gick & Holyoak, 1983; Ross & Kennedy, 1989) and young children (Brown, Kane & Echols, 1986; Holyoak, Junn & Billman, 1984). We have previously shown by simulation that LISA correctly predicts that an abstract schema will be retrieved more readily than a remote analog (Hummel & Holyoak, 1997), using the convergence analogs discussed by Gick and Holyoak (1983). Here we extend those simulations to show how LISA induces schemas from specific analogs.

People are able to induce schemas by comparing just two analogs to one another (Phenomenon 9; Gick & Holyoak, 1983). Indeed, people will form schemas simply as a side effect of applying one solved source problem to one unsolved target problem (Phenomenon 10; Novick & Holyoak, 1991; Ross & Kennedy, 1989). We illustrated LISA's basic capacity for schema induction previously, with the example of the "jealousy" schema. We have also simulated the induction of a schema using the "beach-airport" analogy taken from St. John and McClelland (1990), discussed earlier. For this example, the schema states (roughly): "Person-X ownsvehicle-Y. Person-X wants to go to destination-Z. This causes person-X to drive vehicle-Y to destination-Z." Since both people in the examples from which this schema was induced were male, person-X is also male. But with a few examples of females driving places, the schema would quickly become gender-neutral.

The previous simulations are based on simple "toy" analogies. As a more challenging test of LISA's ability to induce schemas from examples, we gave it the task of using a source "convergence" problem—the "fortress" story of Gick and Holyoak (1980, 1983)—to solve Duncker's (1945) "radiation" problem, and then examined the schema formed in the aftermath of analogical transfer. The simulations started with representations of the radiation and fortress problems that have been used in other simulations of the radiation/fortress analogy (e.g., Holyoak & Thagard, 1989; Kubose et al., in press). The details of the representations are given in Appendix B. The fortress analog states that there is a general who wishes to capture a fortress that is surrounded by friendly villages. The general's armies are powerful enough to capture the fortress, but if he sends all his troops down any single road leading to the fortress, they will trigger mines and destroy the villages. Smaller groups of his troops would not trigger the mines, but no small group by itself would be powerful enough to capture the fortress. The general's dilemma is to capture the fortress without triggering the mines. The solution, provided in the statement of the problem, is for the general to break his troops into many small groups and send them down several roads to converge on the fortress in parallel, thereby capturing the fortress without triggering the mines. The radiation problem states that a doctor has a patient with an inoperable tumor. The doctor has a device that can direct a beam of radiation at the tumor, but any single beam powerful enough to destroy the tumor would also destroy the healthy tissue surrounding it; any beam weak enough to spare the healthy tissue would also spare the tumor. The doctor's dilemma is to destroy the tumor without also destroying the healthy tissue. The solution, which is not provided in the statement of the problem, is to break the beam into several smaller beams which, converging on the tumor from multiple directions, will spare the healthy tissue but destroy the tumor.

Given this problem, LISA induced a more general schema based on the intersection of the fortress and radiation problems. The detailed representation of the resulting schema is given in Appendix B. In brief, the schema states that there is an object (the fortress/tumor) surrounded by another object (the villages/healthy tissue), and a protagonist (general/doctor) who wants to destroy the inner object without harming the surrounding object. The protagonist can form small forces (units/beams), and if he uses small forces, he will destroy the inner object. Given the simplified representation of the problem provided to LISA, this is the correct schema to have induced from the examples.

Although people can form schemas in the aftermath of analogical transfer, the schemas formed will differ depending on which aspects of the analogs are the focus of attention (Phenomenon 11). In the case of problem schemas, more effective schemas are formed when the goal-relevant relations are the focus rather than incidental details (Brown et al., 1986; Brown, Kane & Long, 1989; Gick & Holyoak, 1983). Again borrowing from the materials used by Kubose et al. (2001), we gave LISA the fortress/radiation analogy, but this time biased the simulation against LISA's finding an adequate mapping between them. Specifically, we used the unsolved radiation problem (rather than the solved fortress problem) as the driver, and we set the propositions' importance and support relations so that LISA's random firing algorithm would be unlikely to fire useful propositions in a useful order (see Appendix B and Kubose et al., in press). On this simulation, LISA did not infer the correct solution to the general problem, and it did not induce a useful schema (see Appendix B). Interestingly, it inferred that there is *some* solution to the convergence problem, but it was unable to figure out what that solution is: P unit P15 in the schema states *if-then* (P16, P4), where P4 was a statement of the goal, *destroyed* (*tumor), and P16 presumably should specify an act that will satisfy the goal. However, P16 has no semantic content: It is a P unit without roles or fillers.

Two examples can suffice to establish a useful schema, but people are able to incrementally develop increasingly abstract schemas as additional examples are provided (Phenomenon 12; Brown et al., 1986; Brown et al., 1989; Catrambone & Holyoak, 1989). We will describe a simulation of the incremental acquisition of a schema in the next section.

Interactions Between Schemas and Inference

As schemas are acquired from examples, they in turn guide future mappings and inferences. Indeed, one of the basic properties ascribed to schemas is their capacity to fill in missing

information (Rumelhart, 1980). Here we consider a number of phenomena involving the use of a relationally-constrained abstraction to guide comprehension of new examples that fit the schema.

Schema-guided mapping and inference. Bassok, Wu and Olseth (1995) have shown that people's mappings and inferences are guided by preexisting schemas (Phenomenon 13). These investigators used algebra word problems (permutation problems) similar to ones studied previously by Ross (1987, 1989). Participants were shown how to compute permutations using an example (the source problem) in which items from a set of *n* members were assigned to items from a set of *m* members (e.g., how many different ways can you assign computers to secretaries if there are *n* computers and *m* secretaries?). They were then tested for their ability to transfer this solution to new target problems. The critical basis for transfer hinged on the assignment relation in each analog—i.e., what kinds of items (people or inanimate objects) served in the roles of *n* and *m*. In some problems (type OP) objects (*n*) were assigned to people (*m*; e.g., "computers were assigned to secretaries"); in others (PO) people (*n*) were assigned to objects (*m*; "secretaries were assigned to computers"); and in others (PP) people (*n*) were assigned to other people (*m*) of the same type ("doctors were assigned to doctors"). Solving a target problem required the participant to map the elements of the problem appropriately to the variables *n* and *m* in the equation for calculating the number of permutations.

Although all the problems were formally isomorphic, Bassok et al. (1995) argued that people will typically interpret an "assign" relation between an object and a person as one in which the person *gets* the object (i.e., *assign* (object, person) $\rightarrow$ *get* (person, object)). Importantly, the "get" schema favors interpreting the person as the recipient of the object no matter which entity occupies which role in the stated "assign" relation. As a result, *assign* (person, object), while less natural than the reversed statement, will also tend to be interpreted as implying *get* (person, object). In contrast, because people do not typically own or "get" other people, an "assign" relation involving two people is likely to be interpreted in terms of a symmetrical relation such as "pair" (i.e., *assign* (person1, person2) $\rightarrow$ *pair* (person1, person2)).

These distinct interpretations of the stated "assign" statement yielded systematic consequences for analogical mapping. Given an OP source analog, Bassok et al. (1995) found that people tended to link the "assigned" object to the "received" role (rather than the "recipient" role) of the "get" schema, which in turn was then mapped to the mathematical variable *n*, the number of "assigned" objects. As a result, when the target analog also had an OP structure, transfer was accurate (89%); but when the target was in the reversed PO structure, the set inferred to be *n* was systematically wrong (0% correct)—i.e., the object set continued to be linked to the "received" role of "get", and hence erroneously mapped to *n*. When the source analog was in the less natural PO form, transfer was more accurate to a PO target (53%) than to an OP target (33%). The overall reduction in accuracy when the source analog had the PO rather than the OP form was consistent with people's general tendency to assume that objects are assigned to people, rather than vice versa.

A very different pattern of results obtained when the source analog was in the PP form, so as to evoke the symmetrical "pair" schema. These symmetrical problems did not clearly indicate which set mapped to *n*; hence accuracy was at chance (50%) when the target problem was also of type PP. When the target problem involved objects and people, *n* was usually determined by following the general preference that the assigned set be objects, not people. This preference led to higher accuracy for OP targets (67%) than for PO targets (31%).

We simulated two of Bassok et al.'s (1995) conditions—namely, target OP and target PO, both with source OP—as these conditions yielded the clearest and most striking difference in performance (89% vs. 0% accuracy, respectively). Following the theoretical interpretation of Bassok et al., the simulations each proceeded in two phases. In the first phase, the target problem (OP or PO) was mapped onto the "get" schema, and the schema was allowed to augment the representation of the target (by building additional propositions into the target). In the second phase, the augmented target was mapped onto the source problem in order to map objects and people into the roles of *n* and *m*. The goal of these simulations was to observe how LISA maps the objects and people in the stated problem onto *n* and *m*. As LISA cannot actually solve permutations problems, we assumed that mapping objects and people to *n* and *m* in the mathematically correct

fashion corresponds to finding the correct solution to the problem, and mapping them incorrectly corresponds to finding the incorrect solution.

In all the simulations, the "get" schema contained two propositions. P1 stated that an assigner assigned an object to a recipient—i.e., *assign-to* (*assigner*, *object*, *recipient*)—where the second (object) and third (recipient) roles of the assignment relation are semantically very similar (sharing four of five semantic features; see Appendix B), but the arguments *object* and *recipient* are semantically very different. The semantic features of *object* strongly bias it to map to inanimate objects rather than people, and the features of *recipient* bias it to map to people rather than inanimate objects. As a result, an assignment proposition, *assign-to* (A, B, C), in a target analog will tend to map B to *object* if it is inanimate and to *recipient* if it is a person; likewise, C will tend to map to *object* if it is inanimate and to *recipient* if it is a person. Because the second and third roles of *assign-to* are semantically similar, LISA is only weakly biased to preserve the role bindings stated in the target proposition (i.e., mapping B in the second role to *object* in the second role, and C in the third to *recipient* in the third). It is thus the semantics of the *arguments* of the *assign-to* relation that implement Bassok et al.'s assumption that people are biased to assign objects to people rather than vice versa. The second proposition in the "get" schema, P2, states that the recipient will get the object—i.e., *get* (*recipient*, *object*). Whichever argument of the target's *assign-to* relation maps to *recipient* in the schema's *assign-to* will also map to *recipient* in the context of *gets*.[8] Likewise, whichever argument maps to *object* in the context of *assign-to* will also map to *object* in *get*. For example, the target proposition *assign-to* (boss, computers, secretaries) will map computers to *object*, and secretaries to *recipient*, leading to the (correct) inference *get* (secretaries, computers). However, the target proposition *assign-to* (boss, secretaries, computers)—in which computers "get" secretaries—will also map computers to *object*, and secretaries to *recipient*, leading to the incorrect inference *get* (secretaries, computers).

In both the OP and PO simulations, the source problem (an OP problem) contained four propositions. A teacher assigned prizes to students (P1 is *assign-to* (teacher, students, prizes)), and students got prizes (P2 is *get* (students, prizes)). Thus, prizes instantiate the variable *n* (P3 is *instantiates* (prizes, *n*)), and students instantiate the variable *m* (P4 is *instantiates* (students, *m*); see Appendix B for details). In the OP simulation, the target started with the single proposition *assign-to* (boss, secretaries, computers). In the first phase of the simulation, this proposition was mapped onto the "get" schema (with the target serving as the driver), mapping secretaries to *recipient* and computers to *object*. The schema was then mapped back onto the target, generating (by self-supervised learning) the proposition *get* (secretaries, computers) in the target. In the second phase of the simulation, the augmented target was mapped onto the source. As a result of their shared roles in the *get* relation, secretaries mapped to students and computers to prizes. The source was then mapped back onto the target, generating the propositions *instantiates* (computers, *n*) and *instantiates* (secretaries, *m*). That is, in this case, LISA correctly assigned items to variables in the target problem by analogy to their instantiation in the source problem.

The PO simulation was directly analogous to the OP simulation except that the target problem stated that the boss assigned secretaries to computers—i.e., *assign-to* (boss, computers, secretaries)—reversing the assignment of people and objects to *n* and *m*. As a result, in the first phase of the simulation, the *get* schema augmented the target with the incorrect inference *get* (secretaries, computers). When this augmented target was mapped to the source, LISA again mapped secretaries to students and computers to prizes (based on their roles in *get*). When the

_____

[8] Recall that, within an analog, the same object unit represents an object in all propositions in which that object is mentioned. Thus, in the schema, the object unit that represents *recipient* in *assign-to* (*assigner*, *recipient*, *object*) also represents *recipient* in *gets* (*recipient*, *object*). Therefore, by mapping to *recipient* in the context of *assign-to*, an object in the target automatically maps to *recipient* in *gets* as well.

source was mapped back onto the target, it generated the incorrect inferences *instantiates* (computers, n) and *instantiates* (secretaries, m). Both the OP and PO simulations were repeated over 20 times, and in each case LISA behaved exactly as described above: Like Bassok et al.'s subjects, LISA's ability to correctly map people and objects to n and m in the equation for permutations was dependent on the consistency between the problem as stated (i.e., whether objects were assigned to people or vice versa) and the model's expectations as embodied in its *get* schema. These simple simulations demonstrate that LISA provides an algorithmic account of how people's mappings and inferences are guided by preexisting schemas (Phenomenon 13).

Universal generalization. As we have stressed, the defining property of a relational representation (i.e., symbol system) is that it specifies how roles and fillers are bound together without sacrificing their independence. This property of symbol systems is important because it is directly responsible for the capacity for universal generalization (Phenomenon 14): If and only if a system can (a) represent the roles that objects play explicitly and independently of those objects, (b) represent the role-filler bindings and (c) respond on the basis of the roles objects play, rather than the semantic features of the objects themselves, then that system can generalize responses learned in the context of one set of objects universally to any other objects. We illustrate this property by using LISA to simulate the *identity function* (see also Holyoak & Hummel, 2000a), which Marcus (1998, 2001) used to demonstrate the limitations of traditional (i.e., non-symbolic) connectionist approaches to modeling cognition.

The identity function is among the simplest possible universally quantified functions, and states that for all x, $f(x) = x$. That is, the function takes any arbitrary input, and as output simply returns the input. Adults and children readily learn this function with very few training examples (often as few as one) and readily apply it correctly to new examples bearing no resemblance to the training examples.[9] Marcus (1998) showed that models that cannot bind variables to values (e.g., models based on back-propagation learning) cannot learn this function and generalize it with human-like flexibility. By contrast, LISA can generalize this function universally based on a single training example (much like its performance with the airport and beach analogy). As a source analog, we gave LISA the two propositions $P1^s$, *input* (one), and $P2^s$, *output* (one). This analog states that the number one is the input (to the function), and the number one is the output (of the function). The object unit *one* was connected to semantic units for *number*, *unit*, *single* and *one*. The predicate unit *input* was connected to semantics for *input*, *antecedent* and *question*; *output* was connected to *output*, *consequent* and *answer*. The target analog contained the single proposition $P1^t$, *input* (two), where *input^t* was connected to the same semantics as *input^s*, and *two* was connected to *number*, *unit*, *double* and *two*. The source served as the driver. We fired $P1^s$ followed by $P2^s$ (each fired only once).

When $P1^s$ fired, *input^s* mapped to *input^t*, *one* mapped to *two*, and $P1^s$ mapped to $P1^t$ (along with their SPs). In addition, $P1^t$, its SP, *input^t* and *two* learned global inhibitory connections to and from all unmapped units in the source (i.e., $P2^s$, its SP, and *output*). When $P2^s$ fired, the global inhibitory signals to *input^t* (from *output^s*), to $P1^t$ (from $P2^s$) and to the SP (from the SP under $P1^s$) caused the target to infer units to correspond to the inhibiting source units, which we will refer to as $P2^t$, $SP2^t$ and *output^t*. The object unit *two* in the target was not inhibited, but rather was excited via

---

[9] The first author routinely uses the identity function to illustrate the concept of universal generalization in his introductory cognitive science class. He tells the class he is thinking of a function (but does not tell them the nature of the function), gives them a value for the input to the function, and asks them to guess the output (e.g., "The input is 1. What is the output?"). Students volunteer to guess; if the student guesses incorrectly, the instructor states the correct answer and gives another example input. Most students catch on after as few as two or three examples, and many guess the correct function after the first example. Once the instructor reveals the nature of the function, the students apply it flawlessly, even when the input bears no resemblance whatsoever to the previously used examples (e.g., given "The input is *War and Peace*. What is the output?," all students respond "*War and Peace*," even if all the previous examples had only used numbers as inputs and outputs).

its mapping connection to *one* in the source. As a result, all these units ($P2^t$, $SP2^t$, $output^t$, and *two*) were active at the same time, so LISA connected them to form proposition $P2^t$, *output* (two): Based on the analogy between the target and the source, LISA inferred (correctly) that the output of the function was two.

We ran precisely analogous simulations with two additional target analogs, one stating *input* (Mary) and the other stating *input* (flower), and in each case LISA made the appropriate inference (i.e., *output* (Mary) and *output* (flower), respectively; see Appendix B). Importantly, the target objects Mary and flower had no semantic overlap whatsoever with objects in the source. (Mary was connected to *object*, *living*, *animal*, *human*, *pretty* and *Mary*, and flower to *object, living*, *plant*, *pretty* and *flower*.) In contrast to the simulation with "two" as the novel input, it was not possible *in principle* for LISA to solve these problems based on the semantic similarity between the training example ("one") and the test examples ("Mary" and "flower"). Instead, LISA's ability to solve these more distant transfer problems hinges strictly on the semantic overlap between $input^t$ and $input^s$; in turn, this semantic overlap is preserved only because LISA can bind roles to their arguments without affecting how they are represented. By contrast, traditional connectionist models generalize input-output mappings based strictly on the similarity of the training and test examples, so they cannot in principle generalize a mapping learned on the basis of the example "one" to a non-overlapping test case such as "Mary" or "flower" (cf. Marcus, 1998; St. John, 1992).

Incremental induction of abstract schemas. The examples we used to illustrate universal generalization of the identity function are highly simplified, and serve more as an in-principle demonstration of LISA's capacity for universal generalization than as a psychologically realistic simulation of human performance learning the identity function. In particular, we simplified the simulations by giving LISA stripped-down representations containing only what is necessary to solve the identity function. In more realistic cases, the power of relational inference depends heavily on what is—and what is *not*—represented as part of the source analog, rule or schema (recall the simulations of the radiation/fortress analogy). Irrelevant or mismatching information in the source can make it more difficult to retrieve from memory given a novel target as a cue (Hummel & Holyoak, 1997), and can make it more difficult to map to the target (e.g., Kubose et al., in press). Both factors can impede the use of a source as a basis for making inferences about future targets. In a practical sense, a function is universally quantified only to the extent that the reasoner knows when and how to apply it to future problems; and for this purpose, an abstract schema or rule is generally more useful than a single example.

LISA acquires schemas that are as context-sensitive as the training examples indicate is appropriate. At the same time, LISA's capacity for dynamic variable binding allows it to induce schemas as context-*invariant* as the training examples indicate is appropriate: Contextual shadings are learned from examples, not imposed by the system's architecture. In the limit, if very different fillers are shown to occupy parallel roles in multiple analogs, the resulting schema will be sufficiently abstract as to permit universal generalization (Phenomenon 14).

Another series of simulations using the identity function analogs illustrates this property of the LISA algorithm, as well as the incremental nature of schema induction (Phenomenon 12). In the first simulation, we allowed LISA to map the source *input* (one), *output* (one) onto the target, *input* (two), as described previously, except that we allowed LISA to induce a more general schema. The resulting schema contained two propositions, *input* (*one) and *output* (*one), where the predicate *input* was semantically identical to $input^t$ and $input^s$, and *output* was semantically identical to $output^t$ and $output^s$. (Recall that LISA's intersection discovery algorithm causes the schema to connect its object and predicate units to the semantic units that are common to the corresponding units in the source and target. Since $input^s$ and $input^t$ are semantically identical, their intersection is identical to them as well. The same relation holds for $output^s$ and $*output^t$.) By contrast, the object unit *one was connected very strongly (with connection weights of .98) to the semantic units *number* and *unit*, which are common to "one" and "two", and connected more weakly (weights between .45 and .53) to the unique semantics of "one" and "two" (*single* and *one*, and *double* and *two*, respectively). That is, the object unit *one represents the concept *number*, but with a preference

for the numbers "one" and "two". Based on the examples *input* (one), *output* (one) and *input* (two), LISA induced the schema *input* (*number*), *output* (*number*).

We next mapped *input* (flower) onto this newly induced schema (i.e., the schema *input* (X), *output* (X), where X was semantically identical to **one* induced in the previous simulation: strongly connected to number and unit, and weakly connected to single, one, double and two).   In response, LISA induced the schema *input* (*X), *output* (*X), where *input* and *output* had the same semantics as the corresponding units in the source and target, and **X* learned a much weaker preference for numbers, generally (with weights to number and unit of .44), and an even weaker preference for one and two (with weights in the range of .20 - .24). With three examples, LISA induced progressively more abstract (i.e., universally quantified) schemas for the rule *input* (*x*), *output* (*x*), or *f(x) = x*.

We assume that schema induction is likely to take place whenever one situation or schema is mapped onto another (see Gick & Holyoak, 1983). What is less clear is the fate of the various schemas that get induced after incremental mapping of multiple analogs. Are they kept or thrown away? If they are kept, what prevents them from proliferating? Are they simply gradually forgotten, or are they somehow absorbed into other schemas? It is possible to speculate on the answers to these important questions, but unfortunately, they lie beyond the scope of the current version of LISA.

Learning dependencies between roles and fillers. As illustrated by the findings of Bassok et al. (1995), some schemas "expect" certain kinds of objects to play specific roles. For example, the *get* schema, rather than being universally quantified (i.e., "for all *x* and all *y...* "), appears to be quantified roughly as "for any *person*, *x*, and any *inanimate object*, *y*...." Similarly, as Medin (1989) has noted, people appear to induce regularities involving subtypes of categories (e.g., small birds tend to sing more sweetly than large birds; wooden spoons tend to be larger than metal spoons). During ordinary comprehension, people tend to interpret the meanings of general concepts in ways that fit specific instantiations. Thus the term "container" is likely to be interpreted differently in "The cola is in the container" (bottle) as opposed to "The apples are in the container" (basket) (Anderson & Ortony, 1975). And as noted earlier, a relation such as "loves" seems to allow variations in meaning depending on its role fillers.

Hummel and Choplin (2000) describe an extension of LISA that accounts for these and other context effects on the interpretation and representation of predicates (and objects). The model does so using the same operations that allow intersection discovery for schema induction—namely, feedback from recipient analogs to semantic units. The basic idea is that instances (i.e., analogs) in LTM store the semantic content of the specific objects and relations instantiated in them (i.e., the objects and predicates in LTM are connected to specific semantic units), and that the population of such instances serves as an implicit record of the covariation statistics of the semantic properties of roles and their typical fillers. In the majority of episodes in which a person loves a food, the "loving" involved is of a culinary variety rather then a romantic variety, so the *loves* predicate units are attached to semantic features specifying "culinary love"; in the majority of episodes in which a person loves another person, the love is of a romantic or familial variety so the predicates are attached to semantic features specifying romantic or familial love. Told that Bill loves Mary, the reasoner is reminded of more situations in which people are bound to both roles of the loves relation than situations in which a person is the lover and a food the beloved (although, as noted previously, if Tom is known to be a cannibal, things might be different), so instances of romantic love in LTM become more active than instances of culinary love. Told that Sally loves chocolate, the reasoner is reminded of more situations in which a food is bound to the beloved role, so instances of culinary love become more active than instances of romantic love (although if Sally is known to have a peculiar fetish, things might again be different).

The Hummel and Choplin (2000) extension of LISA instantiates these ideas using exactly the same kind of recipient-to-semantic feedback that drives schema induction. The only difference is that, in the Hummel and Choplin extension, the analogs in question are "dormant" in LTM (i.e., not yet in active memory and therefore not candidates for analogical mapping), whereas in the case of schema induction, the analog(s) in question have been retrieved into active memory. This feedback from dormant analogs causes semantic units to become active to the extent that they correspond to instances in LTM. When "Sally loves chocolate" is the driver, many instances of

"culinary love" are activated in LTM, so the semantic features of culinary love become more active than the features of romantic love; when "Sally loves Tom" is the driver, more instances of romantic love are activated (see Hummel & Choplin, 2000, for details). Predicate and object units in the driver are permitted to learn connections to active semantic units, thereby "filling in" their semantic interpretation. (We assume that this process is part of the rapid encoding of propositions discussed in the *Introduction*.) When the driver states that Sally loves Tom, the active semantic features specify romantic love, so loves is connected to—i.e., interpreted as—an instance of romantic love. When the driver states that Sally loves chocolate, the active semantic features specify culinary love, so loves is interpreted as culinary love.

The simulations reported by Hummel and Choplin (2000) reveal that LISA is sensitive to intercorrelations among the features of roles and their fillers, as revealed in bottom-up fashion by the training examples. But as we stressed earlier, and as illustrated by the simulations of the identity function, LISA can pick up such significant dependencies without sacrificing its fundamental capacity to preserve the identities of roles and fillers across contexts.

General Discussion

Summary

We have argued that the key to modeling human relational inference and generalization is to provide an architecture that preserves the identities of elements across different relational structures while at the same time providing distributed representations of the meanings of individual concepts, and making the binding of roles to their fillers explicit. Neither localist/symbolic representations, nor simple feature vectors, nor complex conjunctive coding schemes such as tensor products, are adequate to meet these joint criteria (Holyoak & Hummel, 2001; Hummel, 2002). The LISA system provides an existence proof that these requirements can be met using a symbolic-connectionist architecture in which neural synchrony dynamically codes variable bindings in WM. We have shown that the model provides a systematic account of a wide range of phenomena concerning human analogical inference and schema induction. The same model has already been applied to diverse phenomena concerning analogical retrieval and mapping (Hummel & Holyoak, 1997; Kubose et al., in press; Waltz, Lau, Grewal & Holyoak, 2000), as well as aspects of visuospatial reasoning (Hummel & Holyoak, 2001), and relational same/different judgments (Kroger, Holyoak & Hummel, 2001). LISA thus provides an integrated account of the major processes involved in human analogical thinking, and lays the basis for a broader account of the human cognitive architecture.

The Power of Self-Supervised Learning

The problem of induction, long recognized by both philosophers and psychologists, is to explain how intelligent systems acquire general knowledge from experience (e.g., Holland, Holyoak, Nisbett & Thagard, 1986). In the late nineteenth century the philosopher Peirce persuasively argued that human induction depends on "special aptitudes for guessing right" (Peirce, 1931-1958, vol. 2, p. 476). That is, humans appear to make "intelligent conjectures" that go well beyond the data they have to work from; and though certainly fallible, these conjectures prove useful often enough to justify the cognitive algorithms that give rise to them.

One basis for "guessing right", which humans share to some degree with other species, is similarity: If a new case is similar to a known example in some ways, assume the new case will be similar in other ways as well. But the exceptional inductive power of humans depends on a more sophisticated version of this criterion: If a new case includes relational roles that correspond to those of a known example, assume corresponding role fillers will consistently fill corresponding roles. This basic constraint on induction is the basis for LISA's self-supervised learning algorithm, which generates plausible though fallible inferences on the basis of relational parallels between cases. This algorithm enables inductive leaps that in the the limit are unfettered by any need for

direct similarities between the elements that fill corresponding roles, enabling universal generalization.

LISA's self-supervised learning algorithm is both more powerful and more psychologically realistic than previous connectionist learning algorithms. Three differences are especially salient. First, algorithms such as back-propagation, which are based on representations that do not encode variable bindings, simply fail to generate human-like relational inferences, and fair poorly in tests of generalization to novel examples (see Hinton, 1986; Holyoak, Novick & Melz, 1994; Marcus, 1998). Second, back-propagation and similar algorithms are supervised forms of learning, requiring an external "teacher" to tell the model how it has erred so it can correct itself. The correction is extremely detailed, informing each output unit exactly how active it should have become in response to the input pattern. But no teacher is required in everyday analogical inference and schema induction; people can self-supervise their use of a source analog to create a model of a related situation or class of situations. And when people do receive feedback (e.g., in instructional settings), it does not take the form of detailed instructions regarding the desired activations of particular neurons in the brain. Third, previous connectionist learning algorithms (including not only supervised algorithms, but also algorithms based on reinforcement learning and unsupervised learning) typically require massive numbers of examples to reach asymptotic performance. In contrast, LISA makes inferences by analogy to a single example. Like humans, but unlike connectionist architectures lacking the capacity for variable binding, LISA can make sensible relational inferences about a target situation based on a single source analog (Gick & Holyoak, 1980), and can form a relational schema from a single pair of examples (Gick & Holyoak, 1983). Thus in such examples as the "beach-airport" analogy, LISA makes successful inferences from a single example whereas back-propagation fails despite repeated presentations of a million (St. John, 1992).

Using Causal Knowledge to Guide Structure Mapping

In LISA, as in other models of analogical inference (e.g., Falkenhainer et al., 1989; Holyoak et al., 1994), analogical inference is *reflective*, in the sense that it is based on an explicit structure mapping between the novel situation and a familiar schema. In humans and in LISA, structure-based analogical mapping is effortful, requiring attention and working memory resources.

One direction for future theoretical development is to integrate analogy-based inference and learning mechanisms with related forms of reflective reasoning, such as *explanation-based learning* (EBL; DeJong & Mooney, 1986; Mitchell, Keller & Kedar-Cabelli, 1986). EBL involves schema learning in which a single instance of a complex concept is related to general background knowledge about the domain. A standard hypothetical example is the possibility of acquiring an abstract schema for "kidnapping" from a single example of a rich father who pays $250,000 to ransom his daughter from an abductor. The claim is that a learner would likely be able to abstract a schema in which various incidental details in the instance (e.g., that the daughter was wearing blue jeans) would be deleted, and certain specific values (e.g., $250,000) would be replaced with variables (e.g., "something valuable"). Such EBL would appear to operate on a single example, rather than the minimum of two required to perform analogical schema induction based on a source and target.

The kidnapping example is artificial in that most educated adults presumably already possess a "kidnapping" schema, so in reality they would have nothing new to learn from a single example. There is evidence, however, that people use prior knowledge to learn new categories (e.g., Pazzani, 1991), and in particular, that inductively-acquired causal knowledge guides subsequent categorization (Lien & Cheng, 2000). The use of causal knowledge to guide learning is entirely consistent with the LISA architecture. A study by Ahn, Brewer and Mooney (1992) illustrates the close connection between EBL and LISA's approach to schema acquisition. Ahn et al. investigated the extent to which college students could abstract schematic knowledge of a complex novel concept from a single instance under various presentation conditions. The presented instance illustrated an unfamiliar social ritual, such as the potlatch ceremony of natives of the Pacific Northwest. Without any additional information, people acquired little or no schematic knowledge from the instance.

However, some schematic knowledge was acquired when the learners first received a description of relevant background information (e.g., the principles of tribal organization), but only when they were explicitly instructed to use this background information to infer the purpose of the ceremony described in the instance. Moreover, even under these conditions, schema abstraction from a single instance was inferior to direct instruction in the schema. As Ahn et al. summarized, "…EBL is a relatively effortful task and … is not carried out automatically every time the appropriate information is provided" (1992, p. 397).

It is clear from Ahn et al.'s (1992) study that EBL requires reflective reasoning, and is dependent on strategic processing in WM. Indeed, their study strongly suggests that EBL, rather than being a distinct form of learning from structure mapping, in fact depends on it. Although it is often claimed that EBL enables schema abstraction from a single instance, Ahn et al.'s instructions in the most successful EBL conditions involved encouraging learners to actively relate the presented instance to a text that provided background knowledge. In LISA, structure mapping can be performed not only between two specific instances, but between a single instance and a more abstract schema. The "background knowledge" used in EBL can be viewed as an amalgam of loosely related schematic knowledge, which is collectively mapped onto the presented instance to guide abstraction of a more integrated schema. The connection between use of prior knowledge and structure mapping is even more evident in a further finding reported by Ahn et al.: once background knowledge was provided, learning of abstract variables was greatly enhanced by presentation of a *second* example. Thus schema abstraction is both guided by prior causal knowledge (a generalization of Phenomenon 11 in Table 2), and is incremental across multiple examples (Phenomenon 12). The LISA architecture appears to be well-suited to provide an integrated account of schema induction based on structure mapping constrained by prior causal knowledge.

Integrating Reflective with Reflexive Reasoning

Although reflective reasoning clearly underlies many important types of inference, people also make myriad inferences that appear much more automatic and effortless. For example, told that Dick sold his car to Jane, one will with little apparent effort infer that Jane now owns the car. Shastri and Ajjanagadde (1993) refer to such inferences as *reflexive*. Even more reflexive is the inference that Dick is an adult human male and Jane an adult human female (Hummel & Choplin, 2000). Reflexive reasoning of this sort figures centrally in human event and story comprehension (see Shastri & Ajjanagadde, 1993).

As illustrated in the discussion of the Hummel & Choplin (2000) extension, LISA is perhaps uniquely well-suited to serve as the starting point for an integrated account of both reflexive and reflective reasoning, and of the relationship between them. The organizing principle is that reflexive reasoning is to analog retrieval as reflective reasoning is to analogical mapping. In LISA, retrieval is the same as mapping except that mapping, but not retrieval, results in the learning of new mapping connections. Applied to reflective and reflexive reasoning, this principle says that reflective reasoning (based on mapping) is the same as reflexive reasoning (based on retrieval) except the former, but not the latter, can exploit newly-created mapping connections. Reflexive reasoning, like retrieval, must rely strictly on existing connections (i.e., examples in LTM). Based on the same type of recipient-based semantic feedback that drives intersection discovery for schema induction, LISA provides the beginnings of an account of reflexive inferences.

Integrating Reasoning With Perceptual Processes

Because LISA has close theoretical links to models of high-level perception (Hummel & Biederman, 1992; Hummel & Stankiewicz, 1996, 1998), it has great promise as an account of the interface between perception and cognition. This is of course a large undertaking, but we have made a beginning (Hummel & Holyoak, 2001). One fundamental aspect of perception is the central role of *metric* information—values and relationships along psychologically continuous dimensions, such as size, distance, and velocity. Metric information underlies not only many

aspects of perceptual processing, but also conceptual processes such as comparative judgments of relative magnitude, both for perceptual dimensions such as size and for more abstract dimensions such as number, social dominance and intelligence. As our starting point to link cognition with perception, we have added to LISA a *Metric Array Module* (MAM; Hummel & Holyoak, 2001), which provides specialized processing of metric information at a level of abstraction applicable to both perception and quasi-spatial concepts.

Our initial focus has been on transitive inference (3-term series problems, such as "A is better than B; C is worse than B; who is best?;" e.g., Clark, 1969; DeSoto, London & Handel, 1965; Huttenlocher, 1968; Maybery, Bain & Halford, 1986; Sternberg, 1980). This reasoning task can be modeled by a 1-dimensional array, the simplest form of metric representation, in which objects are mapped to locations in the array based on their categorical relations (in the previous example, A would be mapped to a location near the top of the array, B to a location near the middle, and C to a location near the bottom; see Huttenlocher, 1968; Hummel & Holyoak, 2001). A classic theoretical perspective has emphasized the quasi-spatial nature of the representations underlying transitive inference (De Soto et al., 1965; Huttenlocher, 1968), and parallels have been identified between cognitive and perceptual variants of these tasks (Holyoak & Patterson, 1981). A long-standing but unresolved theoretical debate has pitted advocates of quasi-spatial representations against advocates of more "linguistic" representations for the same tasks (e.g., Clark, 1969). LISA offers an integration of these opposing views, accommodating "linguistic" phenomena within a propositional system that has access to a metric module.

The operation of MAM builds upon a key advance in mapping flexibility afforded by LISA: the ability to violate the *n*-ary restriction, which forces most other computational models of analogy (for an exception see Kokinov & Petrov, 2001) to map *n*-place predicates only onto other *n*-place predicates (see Hummel & Holyoak, 1997). LISA's ability to violate the *n-ary* restriction—that is, to map an *n*-place predicate to an *m*-place predicate, where *n* *m*—makes it possible for the model to map back and forth between metric *relations* (e.g., more, taller, larger, above, better), which take multiple arguments, and metric *values* (quantities, heights, weights, locations), which take only one. MAM communicates with the other parts of LISA in the same way that "ordinary" analogs communicate: both directly, through mapping connections, and indirectly, through the common set of semantic units. However, unlike "ordinary" analogs, MAM contains additional networks for mapping numerical quantities onto relations and vice versa (Hummel & Biederman, 1992). For example, given the input propositions *greater* (A, B) and *greater* (B, C), MAM will map A, B and C onto numerical quantities, and then map those quantities onto relations such as *greatest* (A), *least* (C) and *greater* (A, C). Augmented with MAM, LISA provides a precise, algorithmic model of human transitive inference in the spirit of the array-based account proposed qualitatively by Huttenlocher (1968). By providing a common representation of object magnitudes and relational roles, LISA can account for "linguistic" effects on transitive inference (e.g., markedness and semantic congruity; see Clark, 1969) as a natural consequence of mapping object values onto an internal array.

Limitations and Open Problems

Finally, we consider some limitations of the LISA model and some open problems in the study of relational reasoning. Perhaps the most serious limitation of the LISA model is that, like virtually all other computational models of which we are aware,[10] its representations must be hand-coded by the modeler: We must tell LISA that Tom is an adult human male, Sally an adult human female, and *loves* is a two-place relation involving strong positive emotions; similarly, we must supply every proposition in its representation of an analog, and tell it where one analog ends and the

_____

[10] The one notable exception to this generalization involves models of very basic psychophysical processes, the representational conventions of which are often constrained by a detailed understanding of the underlying neural representations being modeled.

next begins. This limitation is important because LISA is exquisitely sensitive to the manner in which objects, predicates and analogs are represented. More generally, the problem of hand-coded representations is among the most serious problems facing computational modeling as a scientific enterprise: All models are sensitive to their representations, so the choice of representation is among the most powerful wild cards at the modeler's disposal. Algorithms such as back-propagation represent progress toward solving this problem by providing a basis for discovering new representations in their hidden layers; and models based on unsupervised learning represent progress in their ability to discover new encodings of their inputs. However, both these approaches rely on hand-coded representations of their inputs (and in the case of back-propagation, their outputs), and it is well known that these algorithms are extremely sensitive to the details of these representations (e.g., Marcus, 1998; Marshall, 1995).

Although a general solution to this problem seems far off (and hand-coding of representations is likely to remain a practical necessity even if a solution is discovered), the LISA approach suggests a few directions for understanding the origins of complex mental representations. One is the potential perceptual/cognitive interface suggested by the LISA/MAM model (Hummel & Holyoak, 2001) and by the close ties between LISA and the JIM models of shape perception (Hummel, 2001; Hummel & Biederman, 1992; Hummel & Stankiewicz, 1996, 1998). Perception provides an important starting point for grounding at least some "higher" cognitive representations (cf. Barsalou, 1998). LISA and JIM speak the same language in that both are based on dynamic binding of distributed relational representations in WM, integrated with localist conjunctive coding for tokenization and storage of structures in LTM. Together, they provide the beginnings of an account of how perception communicates with the rest of cognition. Another suggestive property of the LISA algorithm is the integration of reflexive and reflective reasoning as captured by the Hummel and Choplin (2000) model. Although this model still relies on hand-coded representations in LTM, it uses these representations to generate the representations of new instances automatically. Augmented with a basis for grounding the earliest representations in perceptual (and other) outputs, this approach suggests a possible starting point for automatizing LISA's knowledge representations.

A related limitation of the LISA model in its current state is that its representations are strictly formal, lacking meaningful content. For example, *cause* is simply a predicate like any other, connected to semantics that nominally specify the semantic content of a causal relation, but that otherwise have no different status than the semantics of other relations, such as *loves* or *larger-than*. An important open question, both for LISA and for cognitive science more broadly, is how the semantic "features" of predicates and objects come to have meaning via their association with the mind's specialized computing modules—for instance, the machinery of causal induction (e.g., Cheng, 1997) in the case of *cause*, or the machinery for processing emotions in the case of *loves*, or the machinery of visual perception in the case of *larger-than*. This limitation is an incompleteness of LISA's current instantiation rather than a fundamental limitation of the approach as a whole, but is has important practical implications. One is that LISA cannot currently take advantage of the specialized computing machinery embodied in different processing modules. For example, how might LISA process causal relations differently if the semantics of *cause* were integrated with the machinery of causal induction? The one exception to date—which also serves as an illustration of the potential power of integrating LISA with specialized computing modules—is the LISA/MAM model, which integrates LISA with specialized machinery for processing metric relations (Hummel & Holyoak, 2001).

Another important limitation of LISA's knowledge representations (shared by most if not all other models in the literature) is that it cannot distinguish propositions according to their ontological status: It does not know the difference between a proposition *qua* fact, versus conjecture, belief, wish, inference, or falsehood. In its current state, LISA could capture these distinctions using higher-level propositions that specify the ontological status of their arguments. For example, it could specify that the proposition *loves* (Bill, Mary) is false by making it the argument of the predicate *not* (*x*), as in *not* (*loves* (Bill, Mary)); or it could indicate that Mary wished Bill loved her with a hierarchical proposition of the form *wishes* (Mary, *loves* (Bill, Mary)). But this approach seems to miss something fundamental—for example, about the representation of

loves (Bill, Mary) in the mind of Mary, who knows it to be false but nonetheless wishes it were true. It is conceivable that this problem could be solved by connecting the P unit for a proposition directly to a semantic unit specifying its ontological status; but it is unclear whether this approach is adequate. More work needs to be done before LISA (and computational models more generally) can address this important and difficult problem in a meaningful way.

A further limitation is that LISA currently lacks explicit processing goals or knowledge of its own performance. As a result, it cannot decide for itself whether a mapping is good or bad (although, as discussed in the context of the simulations of the results of Lassaline, 1996, we have made some progress in this direction), or whether it has "thought" long enough about a problem and should therefore quit, satisfied with the solution it has reached, or quit and announce defeat. Similarly, in its current state, LISA cannot engage in goal-directed activities such as problem solving. It seems likely that these and related problems will require that LISA be equipped with much more sophisticated metacognitive routines for assessing the state of its own knowledge, its own mapping performance, and its current state relative to its goal state. The system as described in the present paper represents only some first steps on what remains a long scientific journey.

References

Ahn, W., Brewer, W. F., & Mooney, R. J. (1992). Schema acquisition from a single example. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *18*, 391-412.

Albrecht, D. G., & Geisler, W. S. (1991). Motion selectivity and the contrast-response function of simple cells in the visual cortex. *Journal of Neurophysiology*, 7, 531-546.

Alias, D., Blake, R., & Lee, S. H. (1998). Visual features that vary together over time group together over space. *Nature Neuroscience*, *1*, 160-164.

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Erlbaum.

Anderson, R. C., & Ortony, A. (1975). On putting apples into bottles—a problem pf polysemy. *Cognitive Psychology*, *7*, 167-180.

Asaad, W. F., Rainer, G., & Miller, E. K. (1998). Neural activity in the primate prefrontal cortex during associative learning. *Neuron, 21*, 1399-1407.

Baddeley, A. D. (1986). *Working memory*. New York: Oxford University Press.

Baddeley, A. D., & Hitch, G. J. (1974). Working memory. In G. H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 8, pp. 47-89). New York: Academic Press.

Bassok, M., & Holyoak, K. J. (1989). Interdomain transfer between isomorphic topics in algebra and physics. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *15*, 153-166.

Bassok, M., & Olseth, K. L. (1995). Object-based representations: Transfer between cases of continuous and discrete models of change. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *21*, 1522-1538.

Bassok, M., Wu, L.L., & Olseth, K. L. (1995). Judging a book by its cover: Interpretive effects of content on problem-solving transfer. *Memory & Cognition*, *23*, 354-367.

Bonds, A. B. (1989). Role of inhibition in the specification of orientation selectivity of cells in the cat striate cortex. *Visual Neuroscience*, *2*, 41-55.

Broadbent, D. E. (1975). The magical number seven after fifteen years. In A. Kennedy & A. Wilkes (Eds.), *Studies in long-term memory* (pp. 3-18). New York: Wiley.

Brown, A. L., Kane, M. J., & Echols, C. H. (1986). Young children's mental models determine analogical transfer across problems with a common goal structure. *Cognitive Development*, *1*, 103-121.

Brown, A. L., Kane, M. J., & Long, C. (1989). Analogical transfer in young children: Analogies as tools for communication and exposition. *Applied Cognitive Psychology*, *3*, 275-293.

Bundesen, C. (1998). Visual selective attention: Outlines of a choice model, a race model and a computational theory. *Visual Cognition*, *5*, 287-309.

Catrambone, R., & Holyoak, K. J. (1989). Overcoming contextual limitations on problem-solving transfer. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *15*, 1147-1156.

Cheng, P. W. (1997). From covariation to causation: A causal power theory. *Psychological Review*, *104*, 367-405.

Christoff, K., Prabhakaran, V., Dorfman, J., Zhao, Z., Kroger, J. K., Holyoak, K. J., & Gabrieli, J. D. E. (2001). Rostrolateral prefrontal cortex involvement in relational integration during reasoning. *NeuroImage*, *14*, 1136-1149.

Clark, H. H. (1969). Linguistic processes in deductive reasoning. *Psychological Review*, *76*, 387-404.

Clement, C. A., & Gentner, D. (1991). Systematicity as a selection constraint in analogical mapping. *Cognitive Science*, *15*, 89-132.

Cowan, N. (1988). Evolving conceptions of memory storage, selective attention, and their mutual constraints within the human information processing system. *Psychological Bulletin*, *104*, 163-191.

Cowan, N. (1995). *Attention and memory: An integrated framework.* New York: Oxford University Press.

Cowan, N. (in press). The magical number 4 in STM. *Behavioral and Brain Sciences*.

DeJong, G. F., & Mooney, R. J. (1986). Explanation-based learning: An alternative view. *Machine Learning*, *1*, 145-176.

Desmedt, J., & Tomberg, C. (1994). Transient phase-locking of 40 Hz electrical oscillations in prefrontal and parietal human cortex reflects the process of conscious somatic perception. *Neuroscience Letters*, *168*, 126-129.

DeSoto, C., London, M., & Handel, S. (1965). Social reasoning and spatial paralogic. *Journal of Personality and Social Psychology*, *2*, 513-521.

Duncan, J., Seitz, R. J., Kolodny, J., Bor, D., Herzog, H., Ahmed, A., Newell, F. N., & Emslie, H. (2000). A neural basis for general intelligence. *Science*, *289*, 457-460.

Duncker, K. (1945). On problem solving. *Psychological Monographs*, *58* (whole No. 270).

Eckhorn, R., Bauer, R., Jordan, W., Brish, M., Kruse, W. Munk, M. & Reitboeck, H.J. (1988). Coherent oscillations: A mechanism of feature linking in the visual cortex? Multiple electrode and correlation analysis in the cat. *Biological Cybernetics*, *60*, 121-130.

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

Ericsson, K. A., & Kintsch, W. (1995). Long-term working memory. *Psychological Review*, *102*, 211-245.

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

Foley, J. M. (1994). Human luminance pattern-vision mechanisms: Masking experiments require a new model. *Journal of the Optical Society of America*, *11*, 1710-1719.

Frazier, L., & Fodor, J. D. (1978). The sausage machine: A new two-stage parsing model. *Cognition*, *6*, 291-325.

Fuster, J. M. (1997). *The prefrontal cortex: Anatomy, physiology, and neuropsychology of the frontal lobe* (3rd ed.). Philadelphia, PA: Lippincott-Raven.

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

Gick, M. L., & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology*, *12*, 36-355.

Gick, M. L., & Holyoak, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology*, *15*, 1-38.

Goldstone, R. L., Medin, D. L., & Gentner, D. (1991). Relational similarity and the nonindependence of features in similarity judgments. *Cognitive Psychology*, *23*, 222-262.

Gray, C. M. (1994). Synchronous oscillations in neuronal systems: Mechanisms and functions. *Journal of Computational Neuroscience*, *1*, 11-38.

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.

Heeger, D. J. (1992). Normalization of cell responses in cat visual cortex. *Visual Neuroscience*, *9*, 181-197.

Hinton, G. E. (1986). Learning distributed representations of concepts. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Hofstadter, D. R. (2001). Epilogue: Analogy as the core of cognition. In D. Gentner, K. J. Holyoak, & B. N. Kokinov (Eds.), *The analogical mind: Perspectives from cognitive science* (pp. 499-538.. Cambridge, MA: MIT Press.

Hofstadter, D. R., & Mitchell, M. (1994). The Copycat project: A model of mental fluidity and analogy-making. In K. J. Holyoak & J. A. Barnden (Eds.), *Analogical connections. Advances in connectionist and neural computation theory, Vol. 2* (pp. 31-112). Norwood, NJ: Ablex.

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. (1991). Symbolic connectionism: Toward third-generation theories of expertise. In K. A. Ericsson & J. Smith (Eds.), *Toward a general theory of expertise: Prospects and limits* (pp. 301-335). Cambridge: Cambridge University Press.

Holyoak, K. J., & Hummel, J. E. (2000). The proper treatment of symbols in a connectionist architecture. In E. Deitrich & A. Markman (Eds.), *Cognitive dynamics: Conceptual change in humans and machines* (pp. 229-263). Mahwah, NJ: Erlbaum.

Holyoak, K. J., & Hummel, J. E. (2001). Toward an understanding of analogy within a biological symbol system. In D. Gentner, K. J. Holyoak, & B. N. Kokinov (Eds.), *The analogical mind: Perspectives from cognitive science* (pp. 161-195). Cambridge, MA: MIT Press.

Holyoak, K. J., Junn, E. N., & Billman, D. O. (1984). Development of analogical problem-solving skill. *Child Development, 55*, 2042-2055.

Holyoak, K. J., Novick, L. R., & Melz, E. R. (1994). Component processes in analogical transfer: Mapping, pattern completion, and adaptation (pp. 113-180). 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., & Patterson, K. K. (1981). A positional discriminability model of linear order judgments. *Journal of Experimental Psychology: Human Perception and Performance*, *7,* 1283-1302.

Holyoak, K. J., & Spellman, B . A. (1993). Thinking. *Annual Review of Psychology*, *44*, 265-315.

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.

Horn, D., & Usher, M. (1990). Dynamics of excitatory-inhibitory networks. *International Journal of Neural Systems*, *1*, 249-257.

Horn, D., Sagi, D., & Usher, M. (1992). Segmentation, binding and illusory conjunctions. *Neural Computation*, *3*, 509-524.

Hummel, J. E. (2000). Where view-based theories break down: The role of structure in human shape perception. In E. Deitrich & A. Markman (Eds.), *Cognitive dynamics: Conceptual change in humans and machines* (pp. 157-185). Mahwah, NJ: Erlbaum.

Hummel, J. E. (2001). Complementary solutions to the binding problem in vision: Implications for shape perception and object recognition. *Visual Cognition*, *8*, 489-517.

Hummel, J. E., & Biederman, I. (1992). Dynamic binding in a neural network for shape recognition. *Psychological Review*, *99*, 480-517.

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., & Choplin, J. M. (2000). Toward an integrated account of reflexive and reflective reasoning. In L. R. Gleitman & A. K. Joshi (Eds.) *Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society* (pp. 232-237). Mahwah, NJ: Erlbaum.

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

Hummel, J. E., & Holyoak, K. J. (1993). Distributing structure over time. *Behavioral and Brain Sciences*, *16*, 464.

Hummel, J. E., & Holyoak, K. J. (1996). LISA: A computational model of analogical inference and schema induction. In G. W. Cottrell (Ed.), *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, *104*, 427-466.

Hummel, J. E., & Holyoak, K. J. (2001). A process model of human transitive inference. In M. L. Gattis (Ed.), *Spatial schemas in abstract thought* (pp. 279-305). Cambridge, MA : MIT Press.

Hummel, J. E., & Saiki, J. (1993). Rapid unsupervised learning of object structural descriptions. *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society* (pp. 569-574). Hillsdale, NJ: Erlbaum.

Hummel, J. E., & Stankiewicz, B. J. (1996). An architecture for rapid, hierarchical structural description. In T. Inui & J. McClelland (Eds.), *Attention and performance XVI: Information integration in perception and communication* (pp. 93-121). Cambridge, MA: MIT Press.

Hummel, J. E., & Stankiewicz, B. J. (1998). Two roles for attention in shape perception: A structural description model of visual scrutiny. *Visual Cognition*, *5*, 49-79.

Hummel, J. E. (2002). The proper treatment of symbols in a neural architecture. Manuscript in preparation, Department of Psychology, UCLA.

Huttenlocher, J. (1968). Constructing spatial images: A strategy in reasoning. *Psychological Review*, *75*, 550-560.

Irwin, D. E. (1992). Perceiving an integrated visual world. In D. E. Meyer & S. Kornblum (Eds.), *Attention and performance XIV*: *Synergies in experimental psychology, artificial intelligence, and cognitive neuroscience - A Silver Jubilee* (pp. 121-142). Cambridge, MA: MIT Press.

Kahneman, D., & Treisman, A., & Gibbs, B. J (1992). The reviewing of object files: Object-specific integration of information. *Cognitive Psychology*, *24*, 175-219.

Kanwisher, N. G. (1991). Repetition blindness and illusory conjunctions: Errors in bindiong visual types with visual tokens. *Journal of Experimental Psychology: Human Perception and Performance*, *17*, 404-421.

Keane, M. T., & Brayshaw, M. (1988). The Incremental Analogical Machine: A computational model of analogy. In D. Sleeman (Ed.), *European working session on learning* (pp. 53-62). London: Pitman.

Kim, J. J., Pinker, S., Prince, A., & Prasada, S. (1991). Why no mere mortal has ever flown out to center field. *Cognitive Science*, *15*, 173-218.

Kintsch, W., & van Dijk, T. A. (1978). Toward a model of text comprehension and production. *Psychological Review*, *85*, 363-394.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*, 59-69.

Kokinov, B. N. (1994). A hybrid model of reasoning by analogy. In K. J. Holyoak & J. A. Barnden (Eds.), *Advances in connectionist and neural computation theory, Vol. 2: Analogical connections* (pp. 247-318). Norwood, NJ: Ablex.

Kokinov, B. N., & Petrov, A. A. (2001). Integration of memory and reasoning in analogy-making: The AMBR model. In D. Gentner, K. J. Holyoak, & B. N. Kokinov (Eds.), *The analogical mind: Perspectives from cognitive science* (pp. 59-124). Cambridge, MA: MIT Press.

König, P., & Engel, A. K. (1995). Correlated firing in sensory-motor systems. *Current Opinion in Neurobiology*, *5*, 511-519.

Kroger, J. K., Holyoak, K. J., & Hummel, J. E. (2001). Varieties of sameness: The impact of relational complexity on perceptual comparisons. Manuscript in preparation, Department of Psychology, Princeton University.

Kroger, J. K., Saab, F. W., Fales, C. L., Bookheimer, S. Y., Cohen, M. S., & Holyoak, K. J. (in press). Recruitment of anterior dorsolateral prefrontal cortex in human reasoning: A parametric study of relational complexity. *Cerebral Cortex*.

Kubose, T. T., Holyoak, K. J., & Hummel, J. E. (in press). The role of textual coherence in incremental analogical mapping. *Memory and Language*.

Lassaline, M. E. (1996). Structural alignment in induction and similarity. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*, 754-770.

Lee, S. H., & Blake, R. (2001). Neural synchrony in visual grouping: When good continuation meets common fate. *Vision Research*, *41*, 2057-2064.

Lien, Y., & Cheng, P. W. (2000). Distinguishing genuine from spurious causes: A coherence hypothesis. *Cognitive Psychology*, *40*, 87-137.

Lisman, J. E., & Idart, M. A. P. (1995). Storage of 7$\pm$2 short-term memories in oscillatory subcycles. *Science*, *267*, 1512-1515.

Luck, S. J., & Vogel, E. K. (1997). The capacity of visual working memory for features and conjunctions. *Nature*, *390*, 279-281.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, *88*, 375-407.

Marcus, G. F. (1998). Rethinking eliminative connectionism. *Cognitive Psychology*, *37*, 243-282.

Marcus, G. F. (2001). *The algebraic mind: Integrating connectionism and cognitive science*. Cambridge, MA: MIT Press.

Markman, A. B. (1997). Constraints on analogical inference. *Cognitive Science*, *21*, 373-418.

Marshall, J. A. (1995). Adaptive pattern recognition by self-organizing neural networks: Context, uncertainty, multiplicity, and scale. *Neural Networks*, *8*, 335-362.

Maybery, M. T., Bain, J. D., & Halford, G. S. (1986). Information processing demands of transitive inference. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *12*, 600-613.

Medin, D. L. (1989). Concepts and conceptual structure. *American Psychologist*, *44*, 1469-1481.

Medin, D. L., Goldstone, R. L., & Gentner, D. (1994). Respects for similarity. *Psychological Review*, *100*, 254-278.

Mitchell, T. M., Keller, R. M., & Kedar-Cabelli, S. T. (1986). Explanation-based generalization. *Machine Learning*, *1*, 47-80.

Newell, A. (1973). Production systems: Models of control structures. In W. C. Chase (Ed.), *Visual information processing* (pp. 463-526). New York: Academic Press.

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

Page, M. (2001). Connectionist modeling in psychology: A localist manifesto. *Behavioral and Brain Sciences*, *23*, 443-512.

Palmer, S. E. (1978). Structural aspects of similarity. *Memory and Cognition*, *6*, 91-97.

Pazzani, M. J. (1991). Influence of prior knowledge on concept acquisition: Experimental and computational results. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *17*, 416-432.

Peirce, C. S. (1931-1958). *Collected papers*, 8 vols. Edited by C. Hartshorne, P. Weiss & A. Burks. Cambridge, MA: Harvard University Press.

Posner, M. I., & Keele, S. W. (1968). The genesis of abstract ideas. *Jounral of Experimental Psychology*, *77*, 353-363.

Potter, M. C. (1976). Short-term conceptual memory for pictures. *Journal of Experimental Psychology: Human Learning and Memory*, *2*, 509-522.

Reed, S. K. (1987). A structure-mapping model for word problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *13*, 124-139.

Robin, N., & Holyoak, K. J. (1995). Relational complexity and the functions of prefrontal cortex. In M. S. Gazzaniga (Ed.), *The cognitive neurosciences* (pp. 987-997). Cambridge, MA: MIT Press.

Ross, B. (1987). This is like that: The use of earlier problems and the separation of similarity effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *13*, 629-639.

Ross, B. (1989). Distinguishing types of superficial similarities: Different effects on the access and use of earlier problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *15*, 456-468.

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.

Rumelhart, D. E., & Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group, *Parallel distributed processing: Explorations in the microstructure of cognition*, Vol. 1 (pp. 318-362). Cambridge, MA: MIT Press.

Rumelhart, D. E., McClelland, J. L., & the PDP Research Group, *Parallel distributed processing: Explorations in the microstructure of cognition*, Vol. 1. Cambridge, MA: MIT Press.

Shastri, L. (1997). A model of rapid memory formation in the hippocampal system. In M. G. Shafto & P. Langley (Eds.), *Proceedings of the Nineteenth Conference of the Cognitive Science Society* (pp. 680-685). 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.

Singer, W. (2000). Response synchronization, a universal coding strategy for the definition of relations. In M. S. Gazzaniga (Ed.), *The new cognitive neurosciences* (2nd edition). Cambridge, MA: MIT Press.

Singer, W. & Gray, C. M. (1995). Visual feature integration and the temporal correlation hypothesis. *Annual Review of Neuroscience*, *18*, 555-586.

Spellman, B. A., & Holyoak, K. J. (1992). If Saddam is Hitler then who is George Bush?: Analogical mapping between systems of social roles. *Journal of Personality and Social Psychology*, *62*, 913-933.

Spellman, B. A., & Holyoak, K. J. (1996). Pragmatics in analogical mapping. *Cognitive Psychology*, *31*, 307-346.

Smith, E. E., Langston, C., & Nisbett, R. E. (1992). The case for rules in reasoning. *Cognitive Science*, *16*, 1-40.

Sperling, G. (1960). The information available in brief presentations. *Psychological Monographs*, *74* (whole no. 498).

Sternberg, R. J. (1980). Representation and process in linear syllogistic reasoning. *Journal of Experimental Psychology: General*, *109*, 119-159.

Strong, G. W., & Whitehead, B. A. (1989). A solution to the tag-assignment problem for neural networks. *Behavioral and Brain Sciences*, *12*, 381-433.

St. John, M. F. (1992). The Story Gestalt: A model of knowledge-intensive processes in text comprehension. *Cognitive Science*, *16*, 271-302.

St. John, M. F., & McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, *46*, 217-257.

Stuss, D., & Benson, D. (1986). *The frontal lobes*. New York: Raven Press.

Thomas, J. P., & Olzak, L. A. (1997). Contrast gain control and fine spatial discriminations. *Journal of the Optical Society of America*, *14*, 2392-2405.

Usher, M., & Donnelly, N. (1998). Visual synchrony affects binding and segmentation processes in perception. Nature, 394, 179-182.

Usher, M., & Niebur, E. (1996). Modeling the temporal dynamics of IT neurons in visual search: A mechanism for top-down selective attention. Journal of Cognitive Neuroscience, 8, 311-327.

Vaadia, E., Haalman, I., Abeles, M., Bergman, H., Prut, Y., Slovin, H., & Aertsen, A. (1995). Dynamics of neuronal interactions in monkey cortex in relation to behavioural events. *Nature*, *373*, 515-518.

von der Malsburg, C. (1973) Self-organization of orientation selective cells in the striate cortex. *Kybernetik*, *14*: 85-100.

von der Malsburg, C. (1981). The correlation theory of brain function. Internal Report 81-2, Department of Neurobiology, Max-Planck-Institute for Biophysical Chemistry.

von der Malsburg, C., & Buhmann, J. (1992). Sensory segmentation with coupled neural oscillators. *Biological Cybernetics*, *67*, 233-242.

Waltz, J. A., Knowlton, B. J., Holyoak, K. J., Boone, K. B., Mishkin, F. S., de Menezes Santos, M., Thomas, C. R., & Miller, B. L. (1999). A system for relational reasoning in human prefrontal cortex. *Psychological Science*, *10*, 119-125.

Waltz, J. A., Lau, A., Grewal, S. K., & Holyoak, K. J. (2001). The role of working memory in analogical mapping. *Memory & Cognition*, *28*, 1205-1212.

Wang, D., Buhmann, J., & von der Malsburg, C. (1991). Pattern segmentation in associative memory. *Neural Computation*, *2*, 94-106.

Xu, F., & Carey, S. (1996). Infants' metaphysics: The case of numerical identity. *Cognitive Psychology*, *30*, 111-153.

Appendix A:
Details of LISA's Operation

The general sequence of events in LISA's operation is summarized below. The details of each of these steps are described in the subsections that follow.

General Sequence of Events

Repeat the following until directed (by the user) to stop:
1) As designated by the user, select one analog, D, to be the driver, a set of analogs, R, to serve as recipients for mapping, and a set of analogs, L, to remain dormant for retrieval from LTM. R and/or L may be empty sets. In the current paper, we are concerned only with analogs that are in active memory—i.e., all analogs are either D or R—so we shall not discuss the operation of dormant analogs, L, further. For the details of dormant analog operation, see Hummel and Holyoak (1997).
2) Initialize the activation state of the network: Set all inputs and activations to zero.
3) Select a set of P units, P_S, in D to enter working memory. Either choose P_S randomly (according to Eq. 1), or choose a specific proposition(s) as directed by the user. Set the input to any SP unit connected to any P unit in P_S to 1.0.
4) Run the phase set with the units in P_S: Repeatedly update the state of the network in discrete time steps, $t = 1$ to $MaxT$, where $MaxT$ is 220 times the number of SPs in P_S. On each time step, t, do:
   4.1) Update the modes of all P units in R. (The modes of P units in D are simply set: A selected P unit (i.e., any P unit in P_S) is placed into parent mode, and any P units serving as their arguments are placed into child mode.)
   4.2) Update the inputs to all units in P_S (i.e., all P units in P_S, all SPs connected to those P units, and all predicate, object and P units connected to those SPs).
   4.3) Update the global inhibitor.
   4.4) Update the inputs to all semantic units.
   4.5) Update the inputs to all units in all recipient and dormant analogs.
   4.6) Update activations of all units.
   4.7) Retrieve into WM any unit in R that (a) has not already been retrieved and (b) has an activation > 0.5. Create mapping connections (both hypotheses and weights) corresponding to the retrieved unit and all units of the same type (e.g., P, SP, object or predicate) in D.
   4.8) Run self-supervised learning (if it is licensed by the state of the mapping or by the user): For any unit in D that is known not to correspond to any unit in R, build a unit in R to correspond to it. Update the connections between newly created units and other units.
5) Update the mapping weights and initialize the buffers.

2.4) Analogical Mapping

4.1) Updating P Unit Modes

P units operate in three distinct modes: *parent*, *child*, and *neutral* (see Hummel & Holyoak, 1997). (Although the notion of a "mode" of operation seems decidedly non-neural, modes are in fact straightforward to implement with two auxiliary units and multiplicative synapses; Hummel et al., 1994.) In parent mode, a P unit acts as the parent of a larger structure, exchanging input only with SP units "below" itself, i.e., the SPs representing its case roles. A P unit in child mode acts as an argument to a P unit in parent mode and exchanges input only with SPs "above" itself, i.e., SPs relative to which it serves as an argument. A P unit in neutral mode exchanges input with SPs both above and below itself. When the state of the network is initialized (step 2.2 above), all P units

enter neutral mode except those in P_S, which enter parent mode. P units in R update their modes on the basis of their inputs from SP units above and below themselves and via the input over their mapping connections to from P units in D:

$$m_i = \begin{cases} \text{Parent (1) if } SP^{Below} + Prop^{Parent} - SP^{Above} - Prop^{Child} > \vartheta \\ \text{Child (-1) if } SP^{Below} + Prop^{Parent} - SP^{Above} - Prop^{Child} < -\vartheta \\ \text{Neutral (0) otherwise,} \end{cases} \quad \text{(A1.1)}$$

where m_i is the mode of i, and $SP^{Below}, SP^{Above}, Prop^{Parent}$, and $Prop^{Child}$ are weighted activation sums:

$$a_j w_{ij}, \quad \text{(A1.2)}$$

where a_j is the activation of unit j, and w_{ij} is the connection weight from j to i. For SP^{Below}, j are SPs below i; for SP^{Above}, j are SPs above i; for $Prop^{Parent}$, j are P units in D that are in parent mode; and for $Prop^{Child}$, j are proposition units in D that are in child mode. For SP^{Below} and SP^{Above}, w_{ij} are 1.0 for all i and j under the same P unit and 0 for all other i and j. For $Prop^{Parent}$ and $Prop^{Child}$, w_{ij} are mapping weights (0..1.0). φ is a threshold above which P units go into parent mode and below which P units go into child mode. φ is set to a very low value (0.01) so that P units are unlikely to remain in neutral model for more than a few iterations. All model parameters are summarized in a Table A1 at the end of this Appendix.

4.2) Driver Inputs

SP units: SPs in the driver drive the activity of the predicate, object, and P units below themselves (as well as the P units above themselves), and therefore of the semantic units: Synchrony starts in the driver SPs and is carried throughout the rest of the network. Each SP consists of a pair of units, an *excitor* and an *inhibitor*, whose inputs and activations are updated separately. Each SP inhibitor is yoked to the corresponding excitor, and causes the excitor's activation to oscillate. In combination with strong SP-to-SP inhibition, the inhibitors cause separate SPs to fire out of synchrony with one another (Hummel & Holyoak, 1992, 1997).

In the driver, SP excitors receive input from three sources: (1) excitatory input from P units above themselves and from P, object, and predicate units below themselves; (2) inhibitory input from other driver SPs; and (3) inhibitory input from their own inhibitors. On each iteration the net input, n_i, to SP excitor i is:

$$n_i = \alpha + p_i - \frac{\sum_j^{s_i} e_j}{1 + NSP} - \iota I_i, \quad \text{(A2)}$$

where p_i is the activation of the P unit above i, ($= 1$) is the external input to i due to its being in P_s, and therefore attended (see text), e_j is the activation of the excitor of any other SP j in the driver, I_i is the activation of the inhibitor on SP$_i$, NSP is the number of SPs in the driver with activation > [I] ($=0.2$), and s_i is i's sensitivity to inhibition from other SPs (as elaborated below). The divisive normalization (i.e., the $1+NSP$ in the denominator of A2) serves to keep units in a winner-take-all network from oscillating by keeping the absolute amount of inhibition bounded. Sensitivity to inhibition is set to a maximum value, MaxS immediately after SP$_i$ fires (i.e., on the iteration when I_i crosses the threshold from $I_i < 1.0$ to $I_i = 1.0$), and decays by s toward a minimum value, MinS, on each iteration that i does not fire. As detailed below, this variable sensitivity to inhibition (Hummel

& Stankiewicz, 1998) serves to help separate SPs "time-share" by making SPs that have recently fired more sensitive to inhibition than SPs that have not recently fired.

An SP inhibitor receives input only from the corresponding excitor. I_i, changes according to:

$$I_i = \begin{cases} \gamma^s, & I_i \leq {}^L \\ \gamma^f, & I_i > {}^L \\ -\delta^s, & I_i \leq {}^U \\ -\gamma^f, & I_i < {}^U \end{cases}, \quad e_i > {}^I \tag{A3}$$

where e_i is the activation of the corresponding SP excitor, ^{s} is a slow growth rate, ^{f} is a fast growth rate, ^{s} is a slow decay rate, ^{L} is a lower threshold, and ^{U} an upper threshold. Eq. A3 causes the activation of an inhibitor to grow and decay in four phases. Whenever the activation of SP excitor i is greater than ^{I}, I_i, the activation of inhibitor i grows. While I_i is below a lower threshold, ^{L}, it grows slowly (i.e., by ^{s} per iteration). As soon as it crosses the lower threshold, it jumps (by ^{f}) to a value near 1.1 and then truncated to 1.0. Highly active inhibitors drive their excitors to inactivity (Eq. A2). The slow growth parameter, ^{s} (=0.001), was chosen to allow SP excitors to remain active for approximately 100 iterations before their inhibitors inhibit them to inactivity (it takes 100 iterations for I_i to cross ^{L}). The slow decay rate, ^{s} (= 0.01), was chosen to cause I_i to take 10 iterations to decay to the upper threshold, ^{u} (= 0.9), at which point, it drops rapidly to zero, giving e_i an opportunity to become active again (provided it is not inhibited by another SP).

This inhibitor algorithm, which is admittedly inelegant, was adapted by Hummel & Holyoak (1992, 1997) from a much more standard coupled oscillator developed by Wang, Buhmann and von der Malsburg (1991). Although our algorithm appears suspiciously "non-neural" due to all the "if" statements and thresholds, its behavior qualitatively very similar to the more elegant and transparently neurally plausible algorithm of Wang and colleagues. (The Wang et al. Algorithm does not use "if" statements.) Naturally, we do not assume that neurons come equipped with "if" statements, per se, nor do we wish to claim that real neurons use exactly our algorithm with exactly our parameters. Our only claim is that coupled oscillators, or something akin to them, serve an important role in relational reasoning by establishing asynchrony of firing.

SP Time-sharing and the Capacity Limits of WM

When the phase set contains only two SPs, it is straightforward to get them to time-share (i.e., fire cleanly out of synchrony with one another). However, when the phase set contains three or more SPs, getting them to time-share appropriately is more difficult. For example, if there are four SPs, A, B, C and D, then it is important to have all four fire equally often (e.g., in the sequence A B C D A B C D…), but based strictly on the operations described above, there is nothing to prevent, say, A and B from firing repeatedly, never allowing C and D to fire at all (e.g., in the pattern A B A B A B…).

One solution to this problem is to force the SPs to time-share by increasing the duration of the inhibitors' activation. For example, imagine that A fires first, followed by B. If, by the time B's inhibitor becomes active, A's inhibitor is still active, then A will still be inhibited, giving either C or D the opportunity to fire. Let C be the one that fires. When C's inhibitor becomes active, if both A's and B's inhibitors are still active, then D will have the opportunity to fire. Thus, increasing the duration of the inhibitors' activation can force multiple SPs to time-share, in principle without bound. The limitation of this approach is that it is extremely wasteful when there are only a few SPs in WM. For example, if the inhibitors are set to allow seven SPs into the phase set at once (i.e., so that the length of time an SP is inhibited is six times as great as the length of time it is

allowed to fire), and if there are only two SPs in the phase set, then only two of the seven "slots" in the phase set will be occupied by active SPs, leaving the remaining five empty.

A more efficient algorithm for time-sharing—the algorithm LISA uses—is to allow the SP inhibitors to be active for only a very short period of time (i.e., just long enough to inhibit the corresponding SP to inactivity), but to set an SP's sensitivity, s_i, to inhibition *from other SPs* as a function of the amount of time that has passed since the SP last fired (Eq. A2; see also Hummel & Stankiewicz, 1998). Immediately after an SP fires, its sensitivity to inhibition from other SPs goes to a maximum value; on each iteration that the SP does not fire its sensitivity decays by a small amount, s, toward a minimum value. In this way, an SP is inhibited from firing to the extent that (a) it has fired recently, and (b) there are other SPs that have fired less recently (and are therefore less sensitive to inhibition). The advantage of this algorithm is that it causes SPs to time-share efficiently. If there is one SP in the phase set, then it will fire repeatedly, with very little down-time (because there are no other SPs to inhibit it); if there are two or three or four, then they will divide the phase set equally.

The disadvantage of this algorithm—from a computational perspective, although it is an advantage from a psychological perspective—it that it is inherently capacity-limited. Note that the value of s determines how many SPs can time-share. If s is so large that s decays back to its minimum value in the time it takes only one SP to fire, then LISA will not reliably time-share more than two SPs: If A fires, followed by B, then by the time B has finished firing, A's sensitivity to inhibition will already be at its minimum value, and A will compete equally with C and D. If we make s half as large, so that s returns to its minimum value in the time it takes two SPs to fire, then LISA will be able to reliably time-share three SPs: If A fires, followed by B, then by the time B has finished firing, A's sensitivity to inhibition will still be greater than that of C and D; but by the time C finishes firing, A will compete equally with D. In general, if s is set so that s decays to its minimum value in the time that it takes for N SPs to fire, then LISA will be able to reliably time-share $N+1$ SPs.

So where is the capacity limit? Why not simply set s to a very low value and enjoy a very large-capacity working memory? The reason is that as s gets smaller, any individual s_i decays more slowly, and as result, the difference between $s_i(t)$ and $s_j(t)$, for any units i and j at any instant, t, necessarily gets smaller. It is the *difference* between $s_i(t)$ and $s_j(t)$ that determines whether i or j gets to fire. Thus, as s approaches zero, the system's ability to time-share goes up in principle, but the difference between $s_i(t)$ and $s_j(t)$ approaches zero. If there is any noise in the system, then as $s_i(t)$ approaches $s_j(t)$, the system's ability to reliably discriminate them goes down, reducing the system's ability to time-share. Thus, under the realistic assumption that the system's internal noise will be non-zero, there will always be some optimal value of s, above which WM capacity goes down because s decays too rapidly, and below which capacity goes down because the system cannot reliably discriminate $s_i(t)$ from $s_j(t)$. To our knowledge, there is currently no basis for estimating the amount of noise in—and therefore the optimal s for—the human cognitive architecture. We chose a value that value gives the model a WM capacity of five to six SPs, which as noted in the text, is close to the capacity of human WM. As an aside, it is interesting to note that attempting to increase the model's WM capacity by reducing s meets with only limited success: When s is too small, SPs begin to fail to time-share reliably.

P, Object, and Predicate Units: If P unit, i, in the driver is in parent mode, then its net input, n_i,is:

$$n_i = \sum_j e_j, \tag{A4}$$

where e_j is the activation of the excitor on any SP unit below i. The net input to any P unit in child mode, or to predicate or object unit is:

$$n_i = \sum_j (e_j - I_j), \tag{A5}$$

where e_j is the activation of the excitor on any SP above i, and I_j is the activation of the inhibitor on any SP above i.

4.3) Global Inhibitor

All SP units have both excitors and inhibitors, but SP inhibitors are updated only in the driver. This convention corresponds to the assumption that attention is directed to the driver and serves to control dynamic binding (Hummel & Holyoak, 1997; Hummel & Stankiewicz, 1996, 1998). The activity of a *global inhibitor*, Γ, (Horn et al., 1992; Horn & Usher, 1990; Usher & Niebur, 1996; von der Malsburg & Buhman, 1992) helps to coordinate the activity of units in the driver and recipient/dormant analogs. The global inhibitor is inhibited to inactivity ($\Gamma = 0$) by any SP excitor in the driver whose activation is greater than or equal to ϕ^G (=0.7); it becomes active ($\Gamma = 10$) whenever no SP excitor in the driver has an activation greater than or equal to ϕ^G. During a transition between the firing of two driver SPs (i.e., when one SP's inhibitor grows rapidly, allowing the other SP excitor to become active), there is a brief period when no SP excitors in the driver have activations greater than ϕ^G. During these periods, $\Gamma = 10$, and inhibits all units in all non-driver analogs to inactivity. Effectively, Γ serves as a "refresh" signal, permitting changes in the patterns of activity of non-driver analogs to keep pace with changes in the driver.

4.4) Recipient Analog (R) Inputs

Structure units in R receive four sources of input: within-proposition excitatory input, P; within-class (e.g., SP-to-SP, object-to-object, etc.) inhibitory input, C; out-of-proposition inhibitory input, O (i.e., P units in one proposition inhibit SP units in others, and SPs in one inhibit predicate and object units in others); and both excitatory and inhibitory input, M, via the cross-analog mapping weights. The net input, n_i, to any structure unit i in R is the sum:

$$n_i = P_i + M_i - C_i - \pi_i O_i - \pi_i \Gamma, \tag{A6}$$

where $\pi_i = 0$ for P units in parent mode and 1 for all other units. The top-down components of P (i.e., all input from P units to their constituent SPs, from SPs to their predicate and argument units, and from predicates and objects to semantic units) and the entire O term are set to zero until every selected SP in the driver has had the opportunity to fire at least once. This suspension of top-down input allows the recipient to respond to input from every SP in PS before committing to any particular correspondence between its own SPs and specific driver SPs.

P units: The input terms for P units in recipient analog A are:

$$P_i = (1 - \pi_i)\sum_j e_j + \pi_i \sum_k e_k, \tag{A7}$$

where $\pi_i = 0$ if i is in parent mode and 1 otherwise, j are SP units below i, and k are SPs above i.

$$C_i = \frac{\sum_{j \neq i} a_j \mathrm{m}(i,j)}{1 + n}, \tag{A8}$$

where n is the number of P units, k, in A such that $a_k > \gamma^I$. For P units i and j, $\mathrm{m}(i,j) = 1$ if i is in neutral mode or if i and j are in the same mode, and 0 otherwise. Normalizing the inhibition by the

number of active units is simply a convenient way to prevent units from inhibiting one another so much that the cause one another to ocsillate. For object units, j, m(i,j) is 0 if P unit i is in parent mode and 1 otherwise. P units in parent mode exchange inhibition with one another and with P units in neutral mode; and P units in child mode exchange inhibition with inhibit one another, with P units in neutral mode, and with object units. P units in child mode receive out-of-proposition inhibition, O, from SP units in A relative to which they do not serve as arguments:

$$O_i = \sum_j a_j, \tag{A9}$$

where j are SP units that are neither above nor below $\underline{i}$.

$$M_i = \sum_j m(i,j)a_j\left(3w_{ij} - \max(w_i) - \max(w_j)\right), \tag{A10}$$

where a_j is the activation of P unit j in D, w_{ij} (0...1) is the weight on the mapping connection from j to i, and m$(\underline{i},\underline{j})$ is defined as above, $\max(w_i)$ is the maximum value of any mapping weight leading into unit i in and $\max(w_j)$ is the maximum weight leading out of unit j. The max() terms implement global mapping-based inhibition in the recipient. The multiplier 3 on w_{ij} serves to cancel the effect of the global inhibition from j to i due to w_{ij}. For example, let w_{ij} be the maximum weight leading into i and out of j, and let its value be 1.0 for simplicity. Then the net effect of the terns in the parentheses is—i.e., the net effect of w_{ij}—will be 3*1 - 1 - 1 = 1, which is precisely its value.

SP units: The input terms for SP excitors in R are:

$$P_i = p_i + r_i + o_i + c_i, \tag{A11}$$

where p_i is the activation of the P unit above i if that P unit is in parent mode and 0 otherwise, r_i is the activation of the predicate unit under i, o_i is the activation of the object unit under i (if there is one; if not, o_i is 0), and c_i the activation of any P unit serving as a child to i, if there is one, and if it is in child mode (otherwise, c_i is 0).

C_i (within-class inhibition) is given by Eq. A8 where $j \neq i$ are SP units in A.

O_i (out-of-proposition inhibition) is given by Eq. A9, where j are P units in A that are in parent mode and are not above SP i. (i.e., P units representing propositions of which SP i is not a part).

M_i (input over mapping connections) is given by Eq. A10, where j are SP units in D, and m$(i,j) = 1$.

Predicate and object units: The input terms for predicate object units in R are:

$$P_i = \frac{\sum_{j=1}^{n} a_j w_{ij}}{1 + n} + \frac{\sum_{k=1}^{m} a_k}{1 + m}, \tag{A12}$$

where j are semantic units to which i is connected, w_{ij} is the connection weight from j to i, n is the number of such units, k are SP units above i, and m is the number of such units. The denominators of the two terms on the RHS of Eq. A12 normalize a unit's bottom-up (from semantic units) and top-down (from SPs) inputs as a Weber function of the number of units contributing to those inputs (i.e., n and m, respectively). This Weber normalization allows units with different numbers of inputs (including cases in which the sources of one unit's inputs are a subset of the sources of

another's) to compete in such a way that the unit best fitting the overall pattern of inputs will win the inhibitory (winner-take all) competition to respond to those inputs (Marshall, 1995).

C_i is given by Eq. A8 where $j \neq i$ are predicate or object units (respectively) in A.

O_i is given by Eq. A9, where j are SP units not above unit i (i.e., SPs with which i does not share an excitatory connection).

M_i is given by Eq. A10, where j are predicate or object units (respectively) in D, and $m(i, j) = 1$.

4.5) Semantic Units

The net input to semantic unit i is:

$$n_i = \sum_{j \, \alpha \, \{D,R\}} a_j w_{ij}, \tag{A13}$$

where j is a predicate or object unit in any analog, $\alpha \in \{D, R\}$: Semantic inputs receive weighted inputs from predicate and object units in the driver and in all recipients. Units in recipient analogs, R, do not feed input back down to semantic units until all SPs in P_S have fired once. Semantic units for predicates are connected to predicate units but not object units, and semantic units for objects are likewise connected to object units but not predicate units. Semantic units do not compete with one another in the way that structure units do. (In contrast to the structure units, more than one semantic unit of a given type may be active at any given time.) But to keep the activations of the semantic units bounded between 0 and 1, semantic units normalize their activations divisively (see Albrecht & Geisler, 1991, Bonds, 1989, and Heeger, 1992,for physiological evidence for broadband divisive normalization in the visual system of the cat, and Foley, 1994, and Thomas & Olzak, 1997, for evidence for divisive normalization in human vision):

$$a_i = \frac{n_i}{\max(\max(|n|),1)}, \tag{A14}$$

That is, a semantic unit takes as its activation its own input divided by the max of the absolute value of all inputs or 1, which ever is greater. Eq. A14 allows semantic units to take negative activation values in response to negative inputs. This property of the semantic units allows LISA to represent logical negation as mathematical negation (see Hummel & Choplin, 2000).

4.6) Updating the Activation of Structure Units

Structure units update their activations using a standard "leaky integrator," as detailed in Eq. 5.1 in the text.

4.7) Retrieving Units into WM

Unit i in a recipient analog is retrieved into WM, and connections are created between it and units of the same type in P_S, when a_i first exceeds θ^R. $\theta^R = 0.5$ in the simulations reported here.

4.8) Self-supervised Learning

LISA licenses self-supervised learning in analog A when proportion π^m (= 0.7) of A maps to D. Specifically, LISA will license self-supervised learning in A if:

$$\frac{\sum_i q_i j_i}{\sum_i q_i} \geq \pi^m, \tag{A15}$$

where q_i is the (user-defined) importance of unit i, and j_i is 1 if i maps to some unit in D and 0 otherwise. If the user tells it to, LISA will also license self-supervised learning even when Eq. A15. If licensed to do so, LISA will infer a structure unit in A in response to any unmapped structure unit in D. Specifically, as detailed in the text, if unit j in D maps to nothing in A, then when j fires, it will send a global inhibitory signal to all units in A. This uniform inhibition, unaccompanied by any excitation, signals LISA to infer a unit of the same type (i.e., P, SP, object or predicate) in A. Newly inferred units are connected to other units in A according to their co-activity (see text for details). Newly inferred object and predicate units update their connections to semantic units according to:

$$w_{ij} = \mu(a_j - w_{ij}), \tag{A16}$$

where μ (= 0.2) is a learning rate.

5) Mapping Hypotheses and Mapping Weights

At the end of the phase set, each mapping weight, w_{ij}, is updated competitively based on the value of the corresponding hypothesis, h_{ij}, and based on the values of other hypotheses, h_{ik}, referring to the same recipient unit, i (where k are units other than unit j in the driver), and hypotheses, h_{lj}, referring to the same driver unit, j (where l are units other than unit i in the recipient). That is, each weight is updated based on the corresponding mapping hypothesis, as well as the other hypotheses in the same "row" and "column" of the mapping connection matrix. This kind of competitive weight updating serves to enforce the 1:1 mapping constraint (i.e., the constraint that each unit in the source analog maps to no more than one unit in the recipient; Gentner, 1983; Holyoak & Thagard, 1989) by penalizing the mapping from j to i in proportion to the value of other mappings leading away from j and other mappings leading into i.

Mapping weights are updated in three steps. First, the value of each mapping hypothesis is normalized (divided) by the largest hypothesis value in the same row or column. This normalization serves the function of getting all hypothesis values in the range 0…1. Next, each hypothesis is penalized by subtracting from its value the largest *other* value (i.e., not including itself) in the same row or column. This kind of competitive updating is not uncommon in artificial neural networks, and can be conceptualized, neurally, as synapses competing for space on the dendritic tree: To the extent that one synapse grows larger, it takes space away from other synapses (see, e.g., Marshall, 1995). (We are not committed to this particular neural analogy, but it serves to illustrate the neural plausibility of the competitive algorithm.) After this subtractive normalization, all mapping hypotheses will be in the range –1 to 1. Finally, each mapping weight is updated in proportion to the corresponding normalized hypothesis value according to:

$$w_{ij} = \eta(1.1 - w_{ij})h_{ij}\Big]_0^1, \tag{A17}$$

where η is a learning rate and the superscripted 1 and subscripted 0 indicate truncation above 1 and below 0. The asymptotic value 1.1 in Eq. (A17) serves the purpose of preventing mapping connection weights from getting "stuck" at 1.0: With an asymptote of 1.1 and truncation above 1.0, even a weight of 1.0 can, in principle, be reduced in the face of new evidence that the corresponding units do not map; by contrast, with an asymptote of 1.0, a weight of 1.0 would be permanently stuck at that value because the RHS of Eq. (A17) would go to zero regardless of the value of h_{ij}. Truncation below zero means that mapping weights are constrained to take positive values: In contrast to the algorithm described by Hummel and Holyoak (1997), there are no inhibitory mapping connections in the current version of LISA.

In lieu of inhibitory mapping connections, each unit in the driver transmits a global inhibitory input to all recipient units. The value of the inhibitory signal is proportional to the driver unit's activation, and to the value of the largest mapping weight leading out of that unit; the effect of the inhibitory signal on any given recipient unit is also proportional to the maximum mapping weight leading into that unit (see Eq. A10). (We use the maximum weights in this context as a heuristic measure of the strength with which a unit maps; we do not claim the brain does anything literally similar.) This global inhibitory signal (see also Wang, Buhmann & von der Malsburg, 1992) has exactly the same effect as inhibitory mapping connections (as in the 1997 version of LISA), but is more "economical" in the sense that it does not require every driver unit of a given type to have a mapping connection to every recipient unit of that type. The parameters of the global inhibition are set so that if unit j has a strong excitatory mapping connection to unit i, then even though i receives global inhibition from j, the excitation received over the mapping weight compensates for it in such a way that the net effect is exactly equal to the mapping weight times the activation of unit j (see Eq. A10); thus it is as though the global inhibition did not affect unit i at all.

Table A1.
Parameters Governing the Behavior of Individual Units

Parameter	Value	Explanation
	0.01	Proposition mode selection threshold. (Eq. A1.1)
r	0.1	P unit readiness growth rate. (Eq. 4)
	0.5	Support decay rate. The rate at which a P unit i's support from other P units decays on each phase set during which i does not fire. (Eq.3)
	0.3	Activation growth rate. (Eq. 5.1)
	0.1	Activation decay rate. (Eq. 5.1)
R	0.5	Threshold for retrieval into WM.

Parameters Governing SP Oscillations

Parameter	Value	Explanation
ι	3	Inhibitory connection weight from an SP inhibitor to an SP excitor. (Eq. A2)
	1	External (attention-based) input to an SP unit in the phase set, P_S. (Eq. A2)
	0.2	Inhibition threshold: A unit whose activation is greater than is counted as "active" for the purposes of divisively normalizing lateral inhibition by the number of active units. (See Eq. A2.)
MinS	3	Minimum sensitivity to inhibition from other SPs. (See Eq. A2.)
MaxS	6	Maximum sensitivity to inhibition from other SPs. (See Eq. A2.)
s	0.0015	Inhibition-sensitivity decay rate: Rate at which an SP's sensitivity to inhibition from other SPs decays from MaxS to MinS. (See Eq. A2)
S	0.001	Inhibitor activation slow growth rate. (Eq. A3)
F	1	Inhibitor activation fast growth and fast decay rate. (Eq. A3)
S	0.01	Inhibitor activation slow decay rate. (Eq. A3)
L	0.1	Inhibitor lower activation threshold. When an inhibitor's activation grows above ^{L} it jumps immediately to 1.0. (Eq. A3)
U	0.9	Inhibitor upper activation threshold. When an inhibitor's activation decays below ^{U} it falls immediately to 0.0. (Eq. A3)
G	0.7	Threshold above which activation in a driver SP will inhibit the global inhibitor to inactivity.
MaxT	220	Number of iterations a phase set fires for each SP in that set.

Mapping Hypotheses and Connections

Parameter	Value	Explanation
	0.9	Mapping connection learning rate. (Eq. 7)

Self-Supervised Learning

Parameter	Value	Explanation
m	0.7	The proportion of analog A that must be map to the driver before LISA will license analogical inference from the driver into A. (Eq. A15)
μ	0.2	Learning rate for connections between semantic units and newly-inferred object and predicate units.

Appendix B:
Details of Simulations

In the descriptions that follow, we will use the following notation: Each proposition will appear on its own line.   The proposition label (e.g., "P1") will appear on the left, and the proposition's content will appear entirely within parentheses, with the predicate listed first, followed by its arguments (in order). If the proposition has an importance value greater than the default value of 1, then it will be listed after close parenthesis. Each proposition definition is terminated with a semicolon. For example, the line

P1 (Has Bill Jeep) 10;

means that proposition P1 is *has* (Bill, Jeep) and has an importance value of 10.   In the specification of the semantic features of objects and predicates, each object or predicate unit will appear on its own line. The name of the object or predicate unit will appear on the left, followed by the names of the semantic units to which it is connected. Each object or predicate definition is terminated with a semicolon. For example, the line

Bill animal human person adult male waiter hairy bill;

specifies that the object unit *Bill* is connected to the semantic units *animal*, *human*, *adult*, *male*, etc. Unless otherwise specified, all the connection weights between objects the semantics to which they are connected are 1.0, as are the connection weights between predicates and the semantics to which they are connected. The semantic coding of predicates will typically be specified using a shorthand notation in which the name of the predicate unit is followed by a number specifying the number of predicate and semantic units representing that predicate. For example, the line

Has 2 state possess keep has;

specifies that the predicate *has* is represented by two predicate units, *has1* for the agent (possessor) role, and *has2* for the patient (possessed) role; *has1* is connected to the semantic units *state1*, *possess1*, *keep1* and *has1*; and *has2* is connected to *state2*, *possess2*, *keep2* and *has2*. Predicate units listed in this fashion have non-overlapping semantic features in their roles (i.e., none of the semantic units connected to the first role are connected to the second role, and vice versa). Any predicates that deviate from this convention will have their semantic units listed in brackets following the name of the predicate. For example

Has [state possess keep] [state possessed kept];

indicates that both *Has1* and *Has2* are connected to the semantic unit *state.* In addition, *Has1* is connected to *possess* and *keep* (which *Has2* is not), and *Has2* is connected to *possessed* and *kept* (which *Has1* is not).

The firing order of propositions in the driver will be specified by stating the name of the driver analog on the left, followed by a colon, followed by the names of the propositions in the order in which they were fired. Updates of the mapping connections (i.e., the boundaries between phase sets) are denoted by slashes.   User-defined licensing and prohibition of unsupervised learning are denoted by L+ and L- respectively. L* indicates that the decision about whether to license learning was left to LISA (as described in the text).   The default state is user-defined prohibition of learning. For example, the sequence:

Analog 1: P1 P2 / P3 /

Analog 2: L+, P1 /

indicates that first, Analog 1 was designated as the driver, with self-supervised learning left prohibited by default. P1 and P2 were fired in the same phase set, followed by P3 in its own phase set (i.e., mapping connections were updated after P2 and after P3). Next, Analog 2 was made the driver, with self-supervised learning explicitly licensed by the user. P1 was then fired by itself and the mapping connections were updated. The letter R followed by two numbers separated by a slash indicates that LISA was directed to run a certain number of phase sets chosen at random. For example, "R 5/1" means that LISA ran five phase sets, putting one randomly-selected proposition in each; "R 6/2" runs six phase sets, placing two randomly-selected propositions into each.

I. Airport/Beach Analogy

Beach Story (Source)   **Airport Story (Target)**

Propositions: *Propositions*:
P1 (Has Bill Jeep);   P1 (Has John Civic);
P2 (Goto Bill Beach);   P2 (Goto John Airport);
P3 (Want Bill P2);   P3 (Want John P2);
P4 (DrvTo Bill Jeep Beach);
P5 (Cause P3 P4);

Predicates: *Predicates*:
Has 2 state possess keep has;   Has 2 state possess keep has;
Want 2 state desire goal want;   Want 2 state desire goal want;
GoTo 2 trans loc to goto;   GoTo 2 trans loc to goto;
DrvTo 3 trans loc to from vehic drto;
Cause 2 reltn causl cause;

Objects: *Objects*:
Bill animal human person adult male waiter   John animal human person adult male cabby
   hairy bill;   bald john;
Jeep artifact vehicle car american unreliable   Civic artifact vehicle car honda small cheap
   4wd jeep;   civic;
Beach location commerce recreation ocean   Airport location commerce transport fly
   beach;   airport;

Firing Sequence:
Airport Story: P1 / P2 / P3 /
Airport Story: P1 / P2 / P3 /
Beach Story: L+, P4 / P5 /

II. Lassaline (1996)

With the exception of *Cause*, which appeared only in the Strong Constraint simulation, and *Preceed*, which appeared only in the Medium Constraint simulation, all the Lassaline (1996) simulations used the same semantics for predicates and objects in both the source and the target. With the exception of *Cause* and *Preceed*, the predicate and object semantics are listed with the Strong Constraint simulation only. All simulations also used the same firing sequence, which is detailed only with the Weak Constraint simulation.

A. Weak Constraint Version:

Source

Propositions:
P1 (GoodSmell Animal);
P2 (WeakImmune Animal);
P3 (PrA Distractor);
P4 (PrB Distractor);
P5 (PrC Distractor);
P6 (PrD Distractor);
P7 (PrE Distractor);
P8 (PrF Distractor);

Predicates:
GoodSmell 1 smell sense strong goodsmell;
WeakImmune 1 immune disease weak
    weakimmune;
PrA 1 a aa aaa;
PrB 1 b bb bbb;
PrC 1 c cc ccc;
PrD 1 d dd ddd;
PrE 1 e ee eee;
PrF 1 f ff fff;

Objects:
Animal animal thing living;
Distractor animal thing living;

Target

Propositions:
P1 (GoodSmell Animal);
P2 (PrA Distractor);
P3 (PrB Distractor);
P4 (PrC Distractor);
P5 (PrD Distractor);
P6 (PrE Distractor);
P7 (PrF Distractor);

Predicates:
GoodSmell 1 smell sense strong goodsmell;
PrA 1 a aa aaa;
PrB 1 b bb bbb;
PrC 1 c cc ccc;
PrD 1 d dd ddd;
PrE 1 e ee eee;
PrF 1 f ff fff;

Objects:
Animal animal thing living;
Distractor animal thing living;

Firing Sequence:
Target: R 5/1
Source: L* R 5/1

B. Medium Constraint Version:

Source	**Target**

Propositions:

Source	Target
P1 (GoodSmell Animal) 2;	P1 (GoodSmell Animal) 2;
P2 (WeakImmune Animal) 2;	P2 (PrA Animal) 2;
P3 (Preceed P1 P2) 2;	P3 (PrB Animal) 2;
P4 (PrA Animal);	P4 (PrC Distractor);
P5 (PrB Animal);	P5 (PrD Distractor);
P6 (PrC Distractor);	P6 (PrE Distractor);
P7 (PrD Distractor);	P7 (PrF Distractor);
P8 (PrE Distractor);	
P9 (PrF Distractor);	

Support:
  P1 <--> P2 1;
  P1 <--> P3 1;
  P2 <--> P3 1;

Additional Predicates:
Preceed 2 relation time before;

C. Strong Constraint Version:

Source	**Target**

Propositions:

Source	Target
P1 (GoodSmell Animal) 5;	P1 (GoodSmell Animal) 5;
P2 (WeakImmune Animal) 5;	P2 (PrA Animal) 5;
P3 (Cause P1 P2) 5;	P3 (PrB Animal) 5;
P4 (PrA Animal);	P4 (PrC Animal) 5;
P5 (PrB Animal);	P5 (PrD Distractor);
P6 (PrC Animal);	P6 (PrE Distractor);
P7 (PrD Distractor);	P7 (PrF Distractor);
P8 (PrE Distractor);	
P9 (PrF Distractor);	

Support:
  P1 <--> P2 5;
  P1 <--> P3 5;
  P2 <--> P3 5;

Additional Predicate:
Cause 2 relation cause effect entail 3;

III. Spellman and Holyoak (1996)

All versions of the Spellman and Holyoak (1996) simulations used the same semantics for predicates and objects in both the source and the target, so they are listed with the Strong-Constraint/Power-Relevant simulation only. In addition, the support relations between propositions were the same across all simulations, so they are listed with the Strong-Constraint/Power-Relevant simulation only.

A. Strong Constraint, Power Relation Relevant:

Source

Propositions:
P1 (Bosses Peter Mary) 100;
P2 (Loves Peter Mary);
P3 (Cheats Peter Bill);
P4 (Fires Peter Mary);
P5 (Courts Peter Mary);
Support
  P1 <--> P4 100
  P2 <--> P5 100

Target

Propositions:
P1 (Bosses Nancy John) 100;
P2 (Loves John Nancy);
P3 (Bosses David Lisa) 100;
P4 (Loves Lisa David);
P5 (Cheats Nancy David);
P6 (Cheats Lisa John);
Support
  P1 <--> P5 100
  P3 <--> P1 20
  P2 <--> P4 20
  P5 <--> P6 20
  P4 <--> P6 100

Predicates:
Bosses 2 activity job hierarchy higher;
Loves 2 state emotion positive strong;
Cheats 2 activity strong negative nonjob;
Fires 2 activity job negative lose firea fireb;
Courts 2 activity emotion positive date courta
    courtb;

Predicates:
Bosses 2 activity job hierarchy higher;
Loves 2 state emotion positive strong;
Cheats 2 activity strong negative nonjob;

Objects:
Peter person adult worker blond;
Mary person adult worker redhead;
Bill person adult worker bald;

Objects:
John person adult worker thin;
Nancy person adult worker wig;
Lisa person adult worker wily;
David person adult worker hat;

Firing Sequence:
Source: P1 /
Target: R1/1
Source: P1 / P1 / L+, P4 /

B. Strong Constraint, Romantic Relation Relevant:

Source	**Target**
Propositions:	*Propositions*:
P1 (Bosses Peter Mary);	P1 (Bosses Nancy John);
P2 (Loves Peter Mary) 100;	P2 (Loves John Nancy) 100;
P3 (Cheats Peter Bill);	P3 (Bosses David Lisa);
P4 (Fires Peter Mary);	P4 (Loves Lisa David) 100;
P5 (Courts Peter Mary);	P5 (Cheats Nancy David);
	P6 (Cheats Lisa John);

Firing Sequence:
Source: P2 /
Target: R 1/1
Source: P2 / P2 /, L+, P5 /

C. Weak Constraint, Power Relation Relevant:

Source	**Target**
Propositions:	*Propositions*:
P1 (Bosses Peter Mary) 50;	P1 (Bosses Nancy John) 50;
P2 (Loves Peter Mary) 10;	P2 (Loves John Nancy) 10;
P3 (Cheats Peter Bill) 10;	P3 (Bosses David Lisa) 50;
P4 (Fires Peter Mary);	P4 (Loves Lisa David) 10;
P5 (Courts Peter Mary);	P5 (Cheats Nancy David) 10;
	P6 (Cheats Lisa John) 10;

Firing Sequence:
Target: R 6/1

D. Weak Constraint, Romantic Relation Relevant:

Source	**Target**
Propositions:	*Propositions*:
P1 (Bosses Peter Mary) 10;	P1 (Bosses Nancy John) 10;
P2 (Loves Peter Mary) 50;	P2 (Loves John Nancy) 50;
P3 (Cheats Peter Bill) 10;	P3 (Bosses David Lisa) 10;
P4 (Fires Peter Mary);	P4 (Loves Lisa David) 50;
P5 (Courts Peter Mary);	P5 (Cheats Nancy David) 10;
	P6 (Cheats Lisa John) 10;

Firing Sequence:
Target: R 6/1

IV. Jealousy Schema

Source

Propositions:
P1 (Loves Abe Betty);
P2 (Loves Betty Charles);
P3 (Jealous Abe Charles);

Predicates:
Loves 2 emotion strong positive affection
    loves;
Jealous 2 emotion strong negative envy
    jealous;

Objects:
Abe animal human person adult male cabby
    bald abe;
Betty animal human person adult female
    lawyer brunette betty;
Charles animal human person adult male
    clerk glasses charles;

Target

Propositions:
P1 (Loves Alice Bill);
P2 (Loves Bill Cathy);

Predicates:
Loves 2 emotion strong positive affection
    loves;

Objects:
Alice animal human person adult female
    doctor rich alice;
Bill animal human person adult male
    professor tall bill;
Cathy animal human person adult female
    waitress blond cathy;

Inferred Structures:
P3 (*Jealous Abe Cathy);
*jealous1: emotion1 strong1 negative1 envy1
    jealous1;
*jealous2: emotion2 strong2 negative2 envy2
    jealous2;

Firing Sequence:
Source (target, but not schema in active
memory): P1 P2 / P1 / P2 / (bring schema
into active memory) P3 / P3 / P1 / P1 / P2 /
P2 /

The Induced Schema

Propositions:
P1 (*Loves *Abe *Betty);
P2 (*Loves *Betty *Charles);
P3 (*Jealous *Abe *Charles);

Predicates:
*loves1: emotion1 strong1 positive1
    affection1 loves1;
*loves2: emotion2 strong2 positive2
    affection2 loves2;
*lealous1: emotion1 strong1 negative1 envy1
    jealous1;
*lealous2: emotion2 strong2 negative2 envy2
    jealous2;

Objects:
* Abe animal (.95) human (.95) person (.95)
    adult (.95) female (.55) doctor (.55) rich
    (.55) alice (.55) male (.41) cabby (.41)
    bald (.41) abe (.41);
* Betty animal (.97) human (.97) person
    (.97) adult (.97) male (.57) professor
    (.57) tall (.57) bill (.57) female (.44)
    lawyer (.44) brunette (.44) betty (.44);
* Charles animal human person adult
    waitress (.58) blond (.58) cathy (.58)
    clerk (.43) glasses (.43) charles (.43);

Note: Connection weights are listed inside
parentheses following the name of the
semantic unit to which they correspond.
Weights not listed are 1.0. Semantic units
with weights less than 0.10 are not listed.

V. Fortess/Radiation Analogy

A. Strong Constraint (Good firing Order):

Fortress Story (Source)

Propositions:
P1 (inside fortress country);
P2 (captured fortress);
P3 (want general P2);
P4 (canform army largeunit);
P5 (cancapture largeunit fortress);
P6 (gointo roads fortress);
P7 (gointo army fortress);
P8 (vulnerable fortress army);
P9 (vulnerable army mines);
P10 (surround roads fortress);
P11 (surround villages fortress);
P12 (use general largeunit);
P13 (ifthen P12 P2);
P14 (blowup mines largeunit);
P15 (blowup mines villages);
P16 (notwant general P14);
P17 (notwant general P15);
P18 (ifthen P12 P14);
P19 (ifthen P12 P15);
P20 (wantsafe general villages);
P21 (wantsafe general army);
P22 (canform army smallunit);
P23 (cnotcapture smallunit fortress);
P24 (use general smallunit);
P25 (ifthen P24 P2);
P26 (captured dictator);
P27 (want general P26);

Predicates:
gointo 2 approach go1 go2;
vulnerable 2 vul1 vul2 vul3;
surround 2 surround1 surround2 surround3;
innocent 1 state inn1 inn2;
wantsafe 2 safe1 safe2 safe3;
canform 2 trans potential change divide
    canform1;
inside 2 state location in1;
cancapture 2 trans potential change reduce
    cancapture1;
cnotcapture 2 trans potential neutral
    powerless cnotcapture1;
ifthen 2 conditional;
blowup 2 trans change reduce explosive
    blowup1;
use 2 trans utilize use1;
want 2 mentalstate goal want1;
notwant 2 mentalstate goal negative notwant1;
captured 1 state capture1 owned;

Objects:
roads object straight linear road1 road2
    extend;
army object force power army1 army2;
mines object force power mine1 mine2
    blowup;
fortress object castle1 castle2 castle3;
villages object human alive people houses
    villages1;
general person general1 general2 general3;
dictator person dictator1 dic2 dic3;
country location place political large
    country1;
largeunit humangrp military large lunit1;
smallunit humangrp military small sunit1;

Radiation Problem (Target)

Propositions:
P1 (inside tumor stomach);
P2 (destroyed tumor);
P3 (want doctor P2);
P4 (canmake raysource hirays);
P5 (candestroy hirays tumor);
P7 (gointo rays tumor);
P8 (vulnerable tumor rays);
P9 (vulnerable tissue rays);
P11 (surround tissue tumor);
P12 (use doctor hirays);
P13 (ifthen P12 P2);
P14 (candestroy hirays tissue);
P15 (destroyed tissue);
P16 (notwant doctor P15);
P17 (ifthen P12 P15);
P18 (wantsafe doctor tissue);
P20 (canmake raysource lorays);
P21 (cannotdest lorays tumor);
P22 (cannotdest lorays tissue);
P23 (use doctor lorays);
P24 (ifthen P23 P2);

Inferred Propositions:
P25 (use doctor *smallunit);

Firing Sequence:
Source (General story): P11 P20 / P11 P20 /
P11 / P20 / P7 P8 / P7 P8 / P7 / P8 / P3 /
P17 / P24 P25 / P22 P8 / P2 P3 / P15 P17 /
P3 / P17 / R10/1, L+, P11 / P20 / P11 / P20 /
P7 / P8 / P7 / P8 / P3 / P17 / P24 / P25 / P22
/ P8 / P2 / P3 / P15 / P17 / P3 / P17 /

Predicates:
gointo 2 approach go1 go2;
vulnerable 2 vul1 vul2 vul3;
surround 2 surround1 surround2 surround3;
wantsafe 2 safe1 safe2 safe3;
inside 2 state location in1;
destroyed 1 state reduce destroyed1;
want 2 mentalstate goal want1;
notwant 2 mentalstate goal negative notwant1;
canmake 2 trans potential change generate
    canmake1;
candestroy 2 trans potential change reduce
    candestroy1;
cannotdest 2 trans potential neutral powerless
    cannotdest1;
ifthen 2 conditional;
use 2 trans utilize use1;

Inferred Predicates:
*canform2: trans2 poential2 change2 divide2
    canform12 .

Objects:
rays object force power straight linear extend
    blowup;
tissue object biological tissue1 tissue2;
tumor object bad death tumor1;
stomach object biological tissue1 stomach1;
doctor person human alive doc1 doc2 doc3;
patient person human alive pat1 pat2 pat3;
raysource object artifact raysource1;
hirays energy radiation hirays1;
lorays energy radiation lorays1;

Inferred Objects:
*smallunit humangrp military small sunit1.

Induced Schema

Propositions:
P1 (*surround *villages *fortress);
P2 (*wantsafe *general villages);
P3 (*gointo *army *fortress);
P4 (*vulnerable *fortress *army);
P5 (*want *general P6);
P6 (*captured *fortress);
P7 (*notwant *general P8);
P8 (*blowup *mines *villages);
P9 (*use *general *smallunit);
P10 (*ifthen P9 P6);
P11 (*canform *army *smallunit);

Predicates:
*surround1: surround11 surround21
    surround31;
*wantsafe1: safe11 safe21 safe31;
*wantsafe2: safe12 safe22 safe32;
*surround2: surround12 surround22
    surround32;
*gointo1: approach1 go11 go21;
*gointo2: approach2 go12 go22;
*vulnerable1: vul11 vul21 vul31;
*vulnerable2: vul12 vul22 vul32;
*want1: mentalstate1 goal1 want11;
*want2: mentalstate2 goal2 want12;
*notwant1: mentalstate1 goal1 negative1
  notwant11;
*notwant2: mentalstate2 goal2 negative2
    notwant12;
*use1: trans1 utilize1 use11;
*use2: trans2 utilize2 use12;
*ifthen1: conditional1;
*ifthen2: conditional2;

*canform1: trans1 potential1 change1 divide1
    (.52) canform1 (.52) reduce (.49)
    candestroy1 (.49);
*canform2: trans2 potential2 change2;
*captured1: state1 reduce1 (.56) destroyed1
    (.56) capture1 (.44) 1 owned1 (.44);
*blowup1: trans1 change1 potential (.58)
    generate1 (.58) canmake1 (.58) reduce1
    (.43) explosive1 (.43) blowup1 (.43);
*blowup2: trans2 change2 generate2 (.56)
    canmake2 (.56) reduce2 (.45) explosive2
    (.45) blowup2 (.45);

Objects:
*villages: object biological (.57) tissue1 (.57)
    tissue2 (.57) human (.44) alive (.44)
    people (.44) houses (.44) villages1 (.44);
*general: person human (.57) alive (.57) doc1
    (.57) doc2 (.57) doc3 (.57) general1 (.44)
    general2 (.44) general3 (.44);
*fortress: object bad (.57) death (.57) tumor
    (.57) castle1 (.44) castle2 (.44) castle3
    (.44);
*army: object force power straight (.57)
    linear (.57) extend (.57) blowup (.57)
    army1 (.44) army2 (.44);
*smallunit: humangrp military small sunit1;
*mines: object artifact (.57) raysource1 (.57)
    force (.43) power (.43) mine1 (.43) mine2
    (.43) blowup (.43);

Note: Connection weights are listed inside
parentheses following the name of the
semantic unit to which they correspond.
Weights not listed were 1.0. Semantic units
with weights less than 0.10 are not listed.

B. Weak Constraint (Poor Firing Order):

The semantics of objects and predicates in this simulation were the same as those in the previous simulation. All that differed between the two simulations were: which analog served as the source/driver, which propositions were stated (specifically, the solution to the problem was omitted from the doctor story; see also Kubose et al., 2001), and the order in which propositions were fired (specifically, the firing order was more random in the current simulation than in the previous one).

Fortress Story (Target)

Propositions:
P1 (inside fortress country);
P2 (captured fortress);
P3 (want general P2);
P4 (canform army largeunit);
P5 (cancapture largeunit fortress);
P6 (gointo roads fortress);
P7 (gointo army fortress);
P8 (vulnerable fortress army);
P9 (vulnerable army mines);
P10 (surround roads fortress);
P11 (surround villages fortress);
P12 (use general largeunit);
P13 (ifthen P12 P2);
P14 (blowup mines largeunit);
P15 (blowup mines villages);
P16 (notwant general P14);
P17 (notwant general P15);
P18 (ifthen P12 P14);
P19 (ifthen P12 P15);
P20 (wantsafe general villages);
P21 (wantsafe general army);
P22 (canform army smallunit);
P23 (cnotcapture smallunit fortress);
P24 (use general smallunit);
P25 (ifthen P24 P2);
P26 (captured dictator);
P27 (want general P26);

Inferred Propositions:
P28 (notwant general P26);
P29 (ifthen P12 P26);

Radiation Problem (Source)

Propositions:
P1 (inside tumor stomach);
P2 (destroyed tumor);
P3 (want doctor P2);
P4 (canmake raysource hirays);
P5 (candestroy hirays tumor);
P7 (gointo rays tumor);
P8 (vulnerable tumor rays);
P9 (vulnerable tissue rays);
P11 (surround tissue tumor);
P12 (use doctor hirays);
P13 (ifthen P12 P2);
P14 (candestroy hirays tissue);
P15 (destroyed tissue);
P16 (notwant doctor P15);
P17 (ifthen P12 P15);
P18 (wantsafe doctor tissue);

Firing Sequence:
Source (Doctor story): P3 / P16 / R20/1 P3 /
P16 / R20/1, L+, P3 / P16 / R20/1 P3 / P16 /
R20/1

Induced Schema

Propositions:
P1 (*want *doctor P2);
P2 (*destroyed *tumor);
P3 (*notwant *doctor P4);
P4 (*destroyed *tissue);
P5 (*wantsafe *doctor *tissue);
P6 (*vulnerable *tumor *rays);
P7 (*ifthen P8 P2);
P8
P9
P10 (*vulnerable *tissue *rays);
P11 (*canmake *raysource *hirays);
P12 (*surround *tissue *tumor);
P13 (*gointo *rays *tumor)
P14 (*candestroy *hirays *tumor);
P15 (*ifthen P16 P4)
P16
P17
P18 (*candestroy *hirays *tissue);
P19 (*use *doctor *hirays);
P20 (*inside *tumor *stomach);

(Note: Propositions P8, P9, P16 and P17
have P units, but no SPs, predicates or
objects, i.e., no semantic content.)

Predicates:
*want1: mentalstate1 goal1 want11;
*notwant1: mentalstate1 goal1 negative1
    notwant11;
*notwant2: mentalstate2 goal2 negative2
    notwant12;
*surround1: surround11 surround21
    surround31;
*surround2: surround12 surround22
    surround32;
*vulnerable1: vul11 vul21 vul31;
*vulnerable2: vul12 vul22   vul32;
*wantsafe1: safe11 safe21 safe31;
*wantsafe2: safe12 safe22 safe32;
*ifthen1: conditional1;
*ifthen2: conditional2;
*gointo1: approach1 go11 go21;

*gointo2: approach2 go12 go22;
*candestroy1: reduce1 trans1 potential1
    change1 cancapture11 (.58) candestroy11
    (4.3);
*candestroy2: reduce2 trans2 potential2
    change2 cancapture12 (.58) candestroy12
    (4.3);
*inside1: state1 location1 in11;
*inside2: state2 location2 in12;
*destroyed1: state1 capture11 (.56) owned11
    (.56) reduce1 (.44) destroyed11 (.44);
*canmake1: trans1 potential1 change1
    divide1 (.58) canform1 (.58) generate1
    (.43) canmake1 (43);
*canmake2: trans2 potential2 change2
    divide2 (.58) canform2 (.58) generate2
    (.43) canmake2 (43);
*want2: mentalstate2 goal2 want12;
*use1: trans1 utilize1 use11;
*use2: trans2 utilize2 use12;

objects:
*doctor: person general1 (.56) general2 (.56)
    general3 (.56) human (.44) alive (.44)
    doc1 (.44) doc2 (.44) doc3 (.44);
*tissue: object biological tissue1 tissue2;
*tumor: object castle1 (.52) castle2 (.52)
    castle3 (.52) bad (.49) death (.49) tumor
    (.49);
*rays: object force power army1 (.54) army2
    (.54) straight (.47) linear (.47) extend
    (.47) blowup (.47);
*hirays: large humangrp military lunit1
    energy (.77) radiation (.77) hirays1 (.77);
*stomach: location place political large
    country1 object (0.25) biological (0.75)
    tissue1 (0.75) stomach1 (0.75);
*raysource: object artifact raysource1;

Note: Connection weights are listed inside
parentheses following the name of the
semantic unit to which they correspond.
Weights not listed were 1.0. Semantic units
with weights less than 0.10 are not listed.

VI. Bassok et al. (1995)

       The Get Schema and the source OP problem were the same in both the target OP and target PO simulations. The semantics of predicates and objects were consistent throughout (e.g., the predicate AssignTo is connected to the same semantics in every analog in which it appears), so their semantic features are listed only the first time they are mentioned.

Get **Schema**

Propositions:
P1 (AssignTo Assigner Object Recipient);
P2 (Get Recipient Object);

Predicates:
AssignTo
  [assign1 assigner agent]
  [assign2 assign3 assigned2 assigned3 gets2]
  [assign2 assign3 assigned2 assigned3 gets1];
Get 2 action receive gets getsa getsb;

Objects:
Assigner   animate human -object incharge assigner;
Recipient animate human -object -incharge recip;
Object -animate -human object;

Source Problem (Condition OP)

Propositions:
P1 (AssignTo Teacher Prizes Students);
P2 (Get Students Prizes);
P3 (Instantiates Prizes N);
P4 (Instantiates Students M);

Predicates:
Instantiates 2 relation role filler variable value binding;

Objects:
Teacher animate human -object boss incharge adult teacher;
Students animate human -object -boss -incharge child plural students;
Prizes-animate -human gift object desirable prizes;
N variable abstract -animate -object N -M;
M variable abstract -animate -object M -N;

(Note: A negation sign, "-", preceding a semantic unit indicates that the semantic unit an the corresponding object unit have a connection strength of -1. This is a way of explicitly representing that the semantic feature is *not* true of the object. See Appendix A for details.)

A. Target OP Simulation

Stated OP Target Problem
(prior to augmentation by the schema)

Propositions:
P1 (AssignTo Boss Computers Secretaries);

Objects:
Boss animate human -object boss incharge adult boss;
Secretaries animate human -object -boss -incharge adult plural employees;
Computers -animate -human tool object desirable computers;

Firing Sequence:
Target: P1 / P2 /
Source: P1 / L+, P1 / P2 /

Inferred Structures in Target:
P2 (*Get Secretaries Computers);
*Get 2 gets action receive getsa getsb;

Schema-Augmented OP Target Problem

Propositions:
P1 (AssignTo Boss Computers Secretaries);
P2 (Get Secretaries Computers);

Firing Sequence:
Target: P1 / P2 / P1 / P2 /
Source: P1 / P2 / L+, P 3 / P4 / P 3 / P4 /

Inferred Structures in Augmented Target:
P3 (*Instantiates Computers *N);
P4 (*Instantiates Secretaries *M);
*Instantiates 2 relation role filler variable value binding;
*N variable abstract N;
*M variable abstract M;

B. Target PO Simulation

Stated PO Target Problem
(prior to augmentation by the schema)

Propositions:
P1 (AssignTo Boss Secretaries Computers);

Firing Sequence:
Target: P1 / P2 /
Source: P1 / L+, P1 / P2 /

Inferred Structures in Target:
P2 (*Get Secretaries Computers);
*Get 2 gets action receive getsa getsb;

Schema-Augmented OP Target Problem

Propositions:
P1 (AssignTo Boss Secretaries Computers);
P2 (Get Secretaries Computers);

Firing Sequence:
Target: P1 / P2 / P1 / P2 /
Source: P1 / P2 / L+, P 3 / P4 / P 3 / P4 /

Inferred Structures in Augmented Target:
P3 (*Instantiates Computers *N);
P4 (*Instantiates Secretaries *M);
*Instantiates 2 relation role filler variable value binding;
*N variable abstract N;
*M variable abstract M;

VII. Identity Function: Analogical Inference

The semantics of predicates and objects were consistent throughout these simulations, so they are listed only the first time the predicate or object is mentioned.

Solved Example (Source):
Input (1), Output (1)

Propositions:
P1 (Input One);
P2 (Output One);

Predicates:
Input 1 input antecedent question;
Output 1 output consequent answer;

Objects:
One number unit single one1;

Firing Sequence (all simulations):
Source: L+ P1 / P2 /

Input (2) (Target)

Propositions:
P1 (Input Two);

Objects:
Two number unit double two1;

Inferred Structures:
P2 (*Output Two);
*Output 1 output consequent answer;

Input (Flower) (Target)

Propositions:
P1 (Input Flower);

Objects:
Flower object living plant pretty flower1;

Inferred Structures:
P2 (*Output Flower);
*Output 1 output consequent answer;

Input (Mary) (Target)

Propositions:
P1 (Input Mary);

Objects:
Mary object living animal human pretty mary1;

Inferred Structures:
P2 (*Output Mary);
*Output 1 output consequent answer;

VIII. Identity Function: Schema Induction

The semantics of predicates and objects were consistent throughout these simulations, so they are listed only the first time the predicate or object is mentioned.

A. Step 1: Mapping the target Input (2) onto the source Input (1), Output (1)

Source

Propositions:
P1 (Input One);
P2 (Output One);

Predicates:
Input 1 input antecedent question;
Output 1 output consequent answer;

Objects:
One number unit single one1;

Firing Sequence (all simulations):
Source: L+ P1 / P2 /

Target

Propositions:
P1 (Input Two);

Objects:
Two number unit double two1;

Inferred Structures:
P2 (*Output Two);
*Output 1 output consequent answer;

Induced Schema

Propositions:
P1 (*Input *One);
P2 (*Output *One);

Predicates:
*Input 1 input antecedent question;
*Output 1 output consequent answer;

Objects:
*One number (.98) unit (.98) double (.53) two (.53) single (.45) one (.45);

Note: Connection weights are listed inside parentheses following the name of the semantic unit to which they correspond. Weights not listed are 1.0. Semantic units with weights less than 0.10 are not listed.

B. Step 2: Mapping the target Input (flower) onto the schema induced in Step 1: Input (Number), Output (Number)

Source

Propositions:
P1 (Input X);
P2 (Output X);

Predicates:
Input 1 input antecedent question;
Output 1 output consequent answer;

Objects:
X: number (.98) unit (.98) double (.53) two (.53) single (.45) one (.45);

Target

Propositions:
P1 (Input flower);

Objects:
Flower object living plant pretty flower1;

Inferred Structures:
P2 (*Output flower);
*Output 1 output consequent answer;

Induced Schema

Propositions:
P1 (*Input *Number);
P2 (*Output *Number);

Predicates:
*Input 1 input antecedent question;
*Output 1 output consequent answer;

Objects:
*X: number (.44) unit (.44) double (.24) two (.24) single (.20) one (.20);

Note: Connection weights are listed inside parentheses following the name of the semantic unit to which they correspond. Weights not listed are 1.0. Semantic units with weights less than 0.10 are not listed.