Toward an Integrated Account of Reflexive and Reflective Reasoning

John E. Hummel (jhummel@lifesci.ucla.edu)
Department of Psychology
University of California Los Angeles
405 Hilgard Ave.
Los Angeles, CA 90095-1563

Jesse M. Choplin (choplin@lifesci.ucla.edu)
Department of Psychology
University of California Los Angeles
405 Hilgard Ave.
Los Angeles, CA 90095-1563

Abstract

Some inferences are seemingly automatic (*reflexive*; Shastri & Ajjanagadde, 1993), whereas others require more effort (i.e., are *reflective*). We present the beginnings of an integrated account of reflexive and reflective reasoning, based on the LISA model of analogical reasoning (Hummel & Holyoak, 1997). The account holds that reflexive inferences are those that can be generated automatically based on existing knowledge in long-term memory, whereas reflective inferences require explicit structure-mapping and therefore demand greater attention and working memory. According to this account, reflexive inferences manifest themselves in the semantic encoding of objects and predicates, whereas reflective inferences manifest themselves as explicit propositions. In contrast to reflexive inferences, which are equally reflexive, reflective inferences may require more or less effort. We present preliminary simulation results demonstrating that both kinds of inference can be modeled in a single architecture for representing propositional knowledge.

Reflexive vs. Reflective Reasoning

Some inferences are so effortless that we are barely aware of making them. Told that *Bill sold Mary his car*, you will infer that Mary now owns the car so automatically that Shastri and Ajjanagadde (1993) describe the inference as *reflexive*. Even more reflexive is the inference that Bill is probably an adult human male, and Mary an adult human female. Other inferences require more effort. Told that *Bill loves Mary* and *Mary loves Tom*, it is natural to infer that Bill is likely to be jealous of Tom, but this inference arguably requires a bit more reflection (and is less certain) than the inference that Mary is a woman. More effortful still are many kinds of inferences made in the context of scientific and mathematical reasoning, planning, and so forth. What is the relationship between reflexive inferences, such as *Bill is male* or *Mary owns the car*, and more *reflective* inferences, such as *Bill may be jealous of Tom*, or *matter and energy must be special cases of a common physical principle*? And what is the process by which reflective inferences become more reflexive with experience? To a young child, it may not be immediately obvious that Bill's selling Mary his car implies that she now owns the car; but after a sufficient number of examples, the child will eventually induce a schema that makes the relationship between buying and owning reflexive (if evidenced only by the fact that the inference is reflexive for an adult).

In the literature on human cognition, the study of reflexive and reflective reasoning have been largely separate, with the former more common in the study of (for instance) story comprehension (e.g., Kintsch & van Dijk, 1978; Shastri & Ajjanagadde, 1993; St. John, 1992; St. John & McClelland, 1990), and the latter predominating in the study of problem solving (e.g., De Soto, London & Handel, 1965; Byrne & Johnson-Laird, 1989; Newell & Simon, 1976) and reasoning by analogy (e.g., Forbus et al., 1995; Gentner, 1983; Holyoak & Thagard, 1989; Hummel & Holyoak, 1997). Similarly, computational accounts of reflexive inference (e.g., Shastri & Ajjanagadde, 1993; St. John, 1992) have typically had little to say about more reflective forms of reasoning, and models of reflective (e.g., analogical) reasoning have had little to say about the nature of reflexive reasoning.

The most reflexive form of inference is encoding—inferring, for example, that "Mary" in "Bill loves Mary" is an adult human female. It is this most reflexive form of inference that has been most neglected in models of reflective reasoning. One consequence is that these models must be given, in full detail, the representations they are to use for reasoning. For example, the models of Forbus et al. (1995), Holyoak and Thagard (1989) and Hummel and Holyoak (1997) draw analogies between situations whose representations are fully specified for them. In contrast to human reasoners, who can read a sentence such as "Bill loves Mary, but Mary loves Tom" and infer the details for themselves (e.g., that Bill and Tom are adult human males, etc.), these models must be handed all this information for each analogy they are asked to solve.[1] One reason for this division between models of reflective and reflexive reasoning may be that the two kinds of reasoning obey different computational constraints, and therefore demand different kinds of algorithms. At the same time, however, both kinds of inference take place within the same cognitive architecture, and must operate on the same mental representations.

This paper presents the beginnings of an algorithmic account of the relationship between reflexive and reflective reasoning. In broad strokes the account holds that both kinds of reasoning require the capacity to dynamically bind

[1] One notable exception to this generalization is Hofstadter & Mitchel's (1994) CopyCat model, which solves analogy problems of the form X:Y::Z:?, and uses routines to change its representation of X, Y and Z in order to find the best possible analogy. In contrast to other models of analogy, CopyCat is not "stuck" with fixed representations of the elements of its analogies. At the same time, however, this model cannot simulate the kind of encoding discussed here, or the type of reflexive inferences discussed by Shastri & Ajjanagadde (1993).

variables to values (or equivalently, roles to fillers) in order to permit flexible (rule-like) generalization (cf. Shastri & Ajjanagadde, 1993, on the role of variable binding in reflexive reasoning; Holyoak & Hummel, 2000, and Hummel & Holyoak, 1997, on the role of variable binding in reflective reasoning). That is, both reflexive and reflective inferences are operations on symbolic representations. In addition, we propose that what makes reflective reasoning more effortful than reflexive reasoning is, at least in part, that the most reflexive inferences result from a kind of structured memory retrieval (i.e., retrieval that exploits and maintains variable-value bindings), whereas more reflective inferences require explicit structure mapping. That is, as illustrated in the simulations below, we propose that an inference will be fully reflexive when the to-be-inferred information is already available in long-term memory (LTM), and that it becomes progressively more reflective as the to-be-inferred information must be constructed on the basis of mapping large, multi-proposition structures.

The starting point for this effort is Hummel & Holyoak's (1997) LISA model of analogical reasoning, so we will briefly sketch that model's approach to knowledge representation and reflective inference (including memory retrieval, structure mapping, and schema induction). Mapping and retrieval are described in detail in Hummel and Holyoak (1997), and inference an schema induction are described in detail in Holyoak and Hummel (2000).

The LISA Model

The core of LISA's architecture is a system for representing propositions in working memory (WM) by dynamically binding roles to their fillers, and encoding those bindings in LTM. LISA uses synchrony of firing for dynamic binding in WM (Hummel & Holyoak, 1992; Shastri & Ajjanagadde, 1993). Case roles and objects are represented in WM as distributed patterns of activation on a collection of *semantic units* (small circles in Figure 1); case roles and objects fire in synchrony when they are bound together and out of synchrony when they are not. For example, to represent the proposition *sell-to (Bill, Mary, car)* in WM, semantic units representing the *seller* role of the *sell-to* relation (e.g., *transaction*, *exchange*, etc.) fire in synchrony with units representing *Bill* while units representing the *buyer* role fire in synchrony with units representing *Mary*, and units for the *object* role fire in synchrony with units representing *car*. The three sets of units (*Bill+seller*, *Mary+buyer* and *car+object*) must be mutually de-synchronized with one another.

A proposition is encoded in LTM by a hierarchy of *structure units* (Figures 1 and 2). At the bottom of the hierarchy are *predicate* and *object* units (triangles and large circles, respectively, in Figure 1). Each predicate unit locally codes one case role of one predicate. For example, *seller* represents the first (seller) role of the predicate *sell-to*, and has bi-directional excitatory connections to all the semantic units representing that role; *buyer* and *sell-object* represents the buyer and object roles, respectively, and are connected to the corresponding semantics. Semantically-related predicates share units in corresponding roles (e.g., *seller* and *giver* share many units), making the semantic similarity of different predicates explicit. Object units are like predicate units except that they are connected to semantic units describing things rather than roles. For example, *Mary* might be connected to units for *human*, *adult*, *female*, etc., whereas *car* might be connected to *object*, *vehicle*, etc.

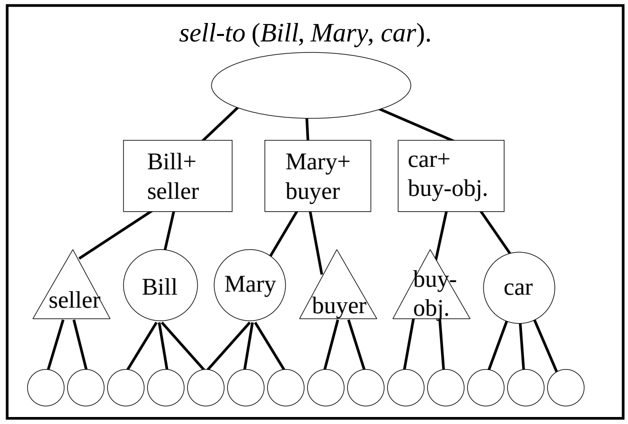

Figure 1. The LISA LTM representation of the proposition *sell-to* (*Bill*, *Mary, car*).

Sub-proposition units (*SPs*; rectangles in Figure 1) bind roles to objects in LTM. *Sell-to (Bill, Mary, car)* would be represented by three SPs, one binding Bill to *seller*, one binding Mary to *buyer*, and one binding car to *sell-object*. SPs have bi-directional excitatory connections with the object and predicate units they bind together. *Proposition* (*P*) units (oval in Figure 1) reside at the top of the hierarchy and have bi-directional excitatory connections with the corresponding SPs. Complete, multi-proposition analogs (i.e., situations, events or schemas) are represented by collections of structure units (see Figure 2).

The final component of LISA's architecture is a set of *mapping connections* between structure units of the same type in different analogs. Every P unit in one analog may share a mapping connection with every P unit in every other analog; likewise, SPs share connections across analogs, as do objects and predicates. For the purposes of mapping and retrieval, analogs are divided into two mutually exclusive sets: a *driver* and one or more *recipients*. Retrieval and mapping are controlled by the driver. LISA performs retrieval and mapping as a form of guided pattern matching. As P units in the driver become active, they generate (via their SP, predicate and object units) synchronized patterns of activation on the semantic units (one pattern for each role-filler binding). The semantic units are shared by all propositions, so the patterns generated by one proposition tend to activate one or more similar propositions in LTM (retrieval) or in working memory (analogical mapping). Mapping differs from retrieval solely by the addition of the modifiable mapping connections. During mapping, the weights on the mapping connections grow larger when the units they link are active simultaneously, permitting LISA to learn the correspondences generated during retrieval. These connection weights also serve to constrain subsequent memory access. By the end of a simulation run, corresponding structure units will have large positive weights on their mapping connections, and non-corresponding units will have strongly negative weights. Hummel & Holyoak (1997) showed that these operations account for a large body of findings in the literature on human analogical reasoning.

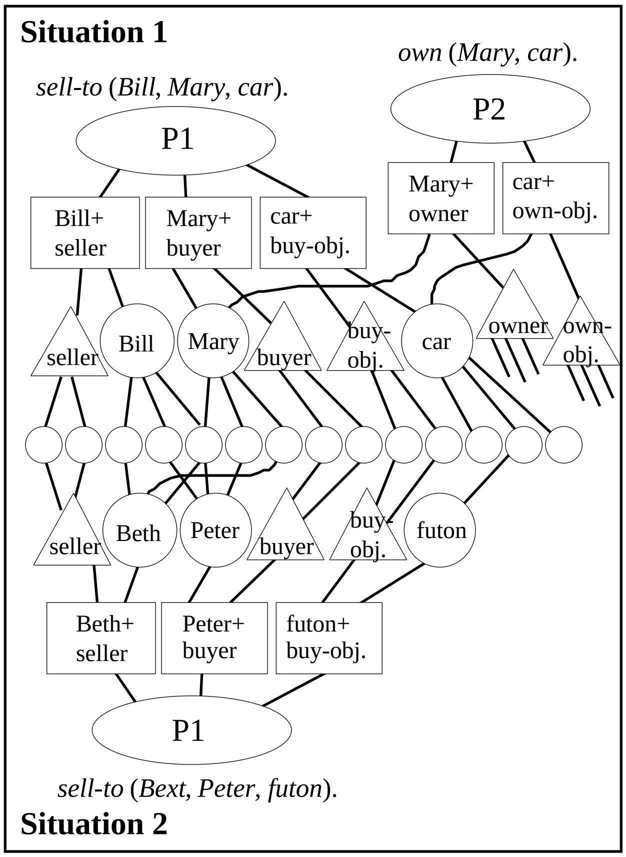

Figure 2. LISA LTM representation of *sell-to* (*Bill*, *Mary*, *car*) and *own* (*Mary*, *car*) (Situation 1; top) and *sell-to* (*Beth*, *Peter*, *futon*) (Situation 2; bottom).

Augmented with unsupervised learning and intersection discovery, LISA's approach to mapping supports inference and schema induction as a natural consequence (Holyoak & Hummel, 2000). Consider an analogy between two situations (Figure 2): In situation 1, Bill sells his car to Mary (proposition P1), so Mary now owns the car (P2); in situation 2, Beth sells her futon to Peter (P1), but there is no explicit statement that Peter now owns the futon. During mapping, corresponding elements in the two analogs will become active simultaneously. For instance, *sell-to (Bill, Mary, car)* in the driver, will activate *sell-to (Beth, Peter, futon)* in the recipient, so corresponding elements (such as *Bill* and *Beth*) will fire in synchrony with one another, and non-corresponding elements (e.g., *Bill* and *futon*) will fire out of synchrony. As a result, LISA learns mapping connections from *Bill* to *Beth*, *Mary* to *Peter*, and *car* to *futon*. Likewise, the roles of *sell-to* in situation 1 map to the corresponding roles of *sell-to* in situation 2. However, nothing in situation 2 maps to the roles of *owns* in situation 1. Therefore, when *owns* (*Mary car*) fires in situation 1, LISA will build units in situation 2 to correspond to the structures in situation 1 representing that proposition: It will build units corresponding to *owner* and *owned*, and connect them to the semantic units representing those roles; it will build SPs corresponding to *owner+Mary* and *owned+car*, and connect them to *owner* and *Peter* and *owned* and *futon*, respectively; finally, it will also build a P unit corresponding to the whole proposition, connecting it to the newly created SPs. (LISA "knows" what to connect to what simply by virtue of which units are firing in synchrony with one another; see Holyoak & Hummel, 2000.) That is, it will infer that Peter now owns the futon.

The same operations permit LISA to perform schema induction in a third "schema" analog. Although we have described the activation of semantic units only from the perspective of the driver, recipient analogs also feed activation to the semantic units. The activation of a semantic unit is a linear function of its input, so any semantic unit that is common to both the driver and recipient will receive input from both and become roughly twice as active as any semantic unit receiving input from only one analog. Shared semantic elements are thus tagged as such by their activations. These shared elements are encoded into the schema by the same unsupervised learning algorithm that performs analogical inference: Units in the schema connect themselves to semantic units and to one another based on their co-activity. Because the learning algorithm is sensitive to the activations of the semantic units, object and predicate units in the schema preferentially learn connections to the semantic that are common to—i.e., the intersection of—the corresponding units in the known situations. In the case of the current example, the induced schema would be roughly *sell-to (person1, person2, object)*, and *own (person2, object)*.

Extension to Reflexive Reasoning

As described above, LISA is a model of reflective reasoning that makes inferences about novel situations based on explicit analogies (i.e., structure-mappings) to familiar situations. However, the operations it uses for analogy, inference and schema induction—most notably, the feedback from the recipient analog to the semantic units (henceforth *recipient feedback*)—suggest themselves as the beginnings of an account of reflexive inference. The basic idea is to use the recipient feedback from structures in LTM (including both general schemas and specific situations) to encode the semantic representation of predicates and objects in the driver. That is, encoding is seen as a collection of reflexive inferences about the properties of the predicates and objects.

Using the recipient feedback in this way solves only one of several problems that must be solved in order to provide a general integrated account of reflexive and reflective reasoning; however, the simulations reported here suggest that it is a useful first step.

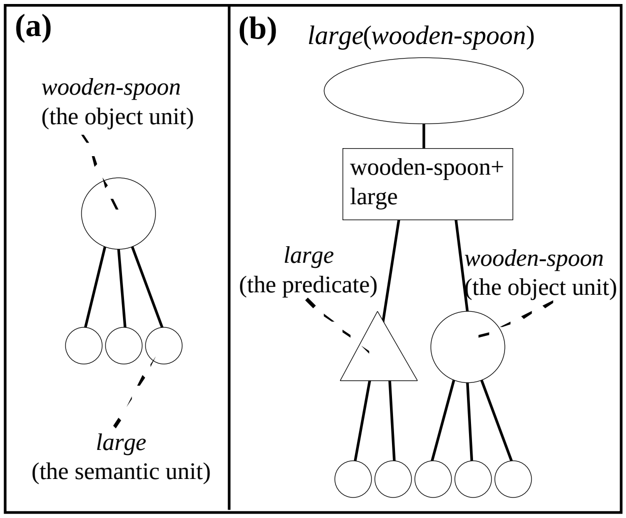

Figure 3. Ways to represent the fact that wooden spoons are large in LISA: (a) *large* as a semantic feature connected to the object unit *wooden-spoon*; (b) *large* as a predicate in the proposition *large* (*wooden-spoon*).

Consider the concept of a wooden spoon. The statement "wooden spoon" makes exactly two properties of the object *wooden spoon* explicit: it is wooden and it is a spoon. But upon encountering these properties, it is irresistible to infer additional properties, such as that it is large (as spoons go), it is more likely to be used for cooking than for eating, etc. (cf. Medin & Shoben, 1988). These properties may be represented in two qualitatively different ways: as semantic "features" of wooden spoons, or as explicit propositions. In terms of the LISA architecture, these ways of representing the properties of wooden spoons are, respectively, as connections from the *wooden-spoon* object unit to semantic units for *large*, *cooking*, etc., (Figure 3a) and as full propositions, complete with predicates, SPs and P units (Figure 3b). We hypothesize that the former type of (semantic feature) representation is established reflexively, as an automatic part of encoding the representation of wooden spoons, whereas the latter (propositional) form is established more reflectively, by thinking explicitly about the properties of wooden spoons. It is important to note that inferring the properties of wooden spoons (either reflexively or reflectively) it is not a simple matter of replacing the default value of the *material* slot in the spoon schema (i.e., *metal*) with the value *wooden* (e.g., as suggested by Smith & Osherson, 1984; Smith et al., 1988), because the attributes of spoons (i.e, the values bound to the slots of the schema) are correlated: Other attributes such as *size=large* will have to be infered, and these attributes will have to replace the corresponding default values in the schema.

We simulated the reflexive form of this inference as follows. We generated five situations (analogs), one corresponding to a "new" situation (thinking about a wooden spoon; analog 1), and the other four corresponding to various schemas in LTM. Analog 1 consists of the single proposition *exist* (*wooden-spoon*), where the object *wooden-spoon* is connected to a single semantic unit, *wooden-spoon*, which serves to represent the type *wooden spoons*; the predicate *exist* is not connected to any semantic units. (*Exist* is a vehicle for instantiating *wooden-spoon* in a proposition so that it can be activated—it allows LISA to "think about" wooden spoons devoid of any particular context.) Analog 2 is a schema for wooden spoons consisting of two propositions: *wooden* (*wooden-spoon*) and *big* (*wooden-spoon*). The object unit *wooden-spoon* has positive connections to the type semantic *wooden-spoon* and to semantics for *utensil*, *spoon*, *wooden*, and *big*. It has negative (inhibitory) connections to semantics for *metal*. The predicate *wooden* has positive connections to semantics for *material* and *wood*, and an inhibitory connection to *metal*; *big* excites semantics for *size* and *big*, and inhibits *small*. Analog 3 is a schema for metal spoons, and consists of *metal* (*metal-spoon*) and *small* (*metal-spoon*). The semantic representations of *metal-spoon*, *metal* and *small* are analogous to those of *wooden-spoon*, *wooden* and *big*, respectively, except that the appropriate semantics are reversed (e.g., the predicate *small* is connected to the semantic *small* rather than *big*, etc.). Analog 4 is a schema for spoons in general, and consists of the propositions *utensil* (*spoon*) and *concave* (*spoon*). Analog 5 is a schema for horseback riding, consisting of the single proposition *ride* (*horse*). Analog 5 serves as a foil to ensure that the model will not simply activate all knowledge in LTM in the course of drawing reflexive inferences about the (semantically empty) wooden spoon in analog 1.

The goal of the simulation is to activate *exist* (*wooden-spoon*) in analog 1 and observe which schemas it activates in LTM, and whether the recipient feedback from the objects and predicates in those schemas allow *wooden-spoon* in analog 1 to learn an appropriate semantic encoding. We therefore set analog 1 to be the driver, and left analogs 2 - 4 "dormant" in LTM (see Hummel & Holyoak, 1997). The proposition *exist* (*wooden-spoon*) was then fired, and propositions in LTM were allowed to respond, feeding activation back to the semantic units. When *exist* (*wooden-spoon*) first fired, both *wooden* (*wooden-spoon*) and *big* (*wooden-spoon*) became active in analog 2 (the wooden spoon schema). *Exist* is semantically empty, so the only semantic feature of analog 1 that activated anything in analog 2 is the type semantic *wooden-spoon* (which is shared by the object *wooden-spoon* in analog 2). As a result, *wooden* (*wooden-spoon*) and *big* (*wooden-spoon*) became equally active in analog 2. The feedback from these propositions to the semantic units began to activate other schemas in LTM: the semantics *utensil* and *spoon* activated units in analogs 3 (the metal spoon schema) and 4 (the generic spoon schema). At the same time, the object *wooden-spoon* (in analog 2) *inhibited* the semantics for *metal.* This inhibition propagated into analog 3 (the metal spoon schema), preventing that schema from becoming active and in turn, preventing it from activating its own semantics. When the pattern of activation settled, analogs 2 and 4 were fully active (i.e., both propositions were active in both analogs), along with all the semantic units to which they are connected. As a result, the object *wooden-spoon* in analog 1 learned connections to the semantics for *utensil*, *spoon*, *wooden* and *big* (due to the feedback from the wooden spoon schema), and to *concave* and *utensil* (based on the generic spoon schema): The model reflexively inferred the semantic properties of the wooden spoon.

In a second simulation, analog 1 consisted of the proposition *exist* (*metal-spoon*)—this time having LISA "think about" metal spoons—and we ran the same operations described above. This time, analog 3 (the metal spoon schema) and analog 4 (the generic spoon schema) became active, and the model inferred the properties of the *metal-spoon* in analog 1: *metal-spoon* learned connections to the semantic units for *utensil*, *spoon*, *metal* and *small* (due to the feedback from the metal spoon schema), and to *concave* and *utensil* (based on the generic spoon schema). In both these simulations, it is interesting to note that LISA assigned each object (the metal spoon or the wooden spoon) to the most general category appropriate (by activating the generic *spoon* schema), but it did not categorize metal spoons as wooden spoons, or vice versa. As a result, it made appropriate inferences about the objects at multiple levels of abstraction (e.g., that the wooden spoon would be big [which is specific to wooden spoons] and that it would be concave [which is general to all spoons]).

In the previous simulations, the inferences were purely reflective, in the sense that we did not allow LISA to retrieve the schemas from memory and map them back onto analog 1. When we allowed the model to reflect on the properties of wooden spoons—by making the wooden spoon schema the driver, the wooden spoon version of analog 1 the recipient, and allowing it to explicitly map the schema onto analog 1—it inferred the explicit propositions *wooden* (*wooden-spoon*) and *big* (*wooden-spoon*) in analog 1. (It did do by exactly the same operations described previously in the discussion of LISA's operation.) Similarly, when we allowed it to reflect on the fact that wooden spoons are spoons—by mapping the spoon schema into analog 1—it inferred the propositions *utensil* (*wooden-spoon*) and *big* (*wooden-spoon*). But importantly, it did not infer any of these propositions until it explicitly brought the corresponding schema into WM and mapped it onto analog 1. This property is interesting in combination with the model's ability to reason reflexively to the most generic category applicable (e.g., to assign wooden spoons semantic features that are true of all spoons based on the generic spoon schema): Together, they predict that reflexive inferences—which manifest themselves in the (implicit) semantic encoding of an object or predicate—will automatically take place across multiple levels of category abstraction, whereas reflective inferences—which cause the construction of explicit propositional structures—will only take place when the reasoner explicitly reflects on the fact that the object belongs to the category (i.e., explicitly maps the category schema onto the object). To our knowledge, no one has yet tested this prediction of the model.

Discussion

Using simple operations already in place to simulate reflective analogy-based inference—namely, recipient feedback and unsupervised learning—LISA was able to reflexively infer the meaning of "wooden spoon" and "metal spoon" based on examples in LTM. These inferences were reflexive in the sense that they did not require the model to explicitly map the structures in the new example (analog 1) to the structures in LTM. Instead, they were drawn in the course of what is analog retrieval in LISA (i.e., the process of retrieving a source analog or schema from LTM given a novel target as a cue; see Hummel & Holyoak, 1997). By the end of the first two simulations, the objects *wooden-spoon* (in the first simulation) and metal spoon (second simulation) had semantic encodings that were richer than what was provided at the beginning of the simulation. In each case, the object unit started with a single semantic feature (*wooden-spoon* or *metal-spoon*) and ended with a semantic encoding specifying its size (*big* or *small*), material (*wooden* or *metal*), shape (*concave*) and use (*utensil*). However, in neither of the first two simulations did analog 1 end up with any new propositions. By contrast, in the third and fourth simulations, when the schemas were called into WM and allowed to map to analog 1, the model inferred propositions that explicitly stated the properties of the wooden spoon. According to the present account, inferring a new proposition (e.g., one stating explicitly that the wooden spoon is big) is a reflective process that requires retrieval of a schema (or specific situation), and an explicit mapping of the structures in that schema to the structures in the new example.

In this respect, our use of the term "reflexive" is somewhat more restrictive than Shastri & Ajjanagadde's (1993). On our account, an inference such as "Mary now owns the car" is not strictly reflexive unless it is represented strictly as features in the *semantic* representation of Mary (i.e., as connections from the unit *Mary* to units representing *ownership*). If instead (or in addition) the inference is represented as an explicit proposition (*own* (*Mary car*)), we would classify it as "reflective but easy" (as noted previously, some reflective inferences are easier than others). This distinction between our account and that of Shastri & Ajjanagadde stems primarily from the fact that LISA represents objects and predicates as distributed patterns of activation in WM, which precludes binding an object to more than one predicate role at a time (see Hummel & Holyoak, 1997). As a result, LISA makes a strong distinction between properties *qua semantic features* and properties *qua explicit propositions*. By contrast, Shastri & Ajjanagadde's model represents each object or predicate as a localist unit, making it possible to "stack" predicates on objects, effectively representing multiple predicate-object bindings (i.e., multiple propositions) in parallel (cf. Hummel & Holyoak, 1997). Whether the human mind makes a strong distinction between features and propositions (like LISA), or permits "stacking" of predicates (like Shastri & Ajjanagadde's model) is an empirical question.

The Origins of Object Features

To this point our discussion of reflexive inference—based on learning connections to features activated by feedback from structures in LTM—has begged a major question: If objects learn the features that describe them by "comparing themselves to" other objects in LTM, then how did the objects in LTM learn the features that describe themselves in the first place? Answering this question "by comparing themselves to objects in LTM when *they* were encoded" is unsatisfying because it brings to mind the infinite regress, "and where did *those* objects learn *their* features?" etc. Although we are far from being able to provide a complete answer to this very difficult question, one aspect of the model that we have not yet discussed may provide a partial answer. Specifically, we allow some semantic features that belong to predicate units to attach themselves to object units during the course of reflexive inference. (In the original LISA, predicate semantics were attached strictly to predicates and object semantics strictly to objects [see Hummel & Holyoak, 1997]; this approach is a departure from that convention.) As a result, roles to which an object is attached in many situations or schemas in LTM (e.g., the role *big*, in the case of *big* (*wooden-spoon*)) can become attached directly to new instances of those objects as semantic features. In this way, an inference like *own* (*Mary, car*) can become truly reflexive in the sense of the definition suggested here: If, in several examples in LTM, the buyer of a product is also represented as the owner of that product (i.e., in a separate *own* (*person, object*) proposition), then the semantics of the predicate *own* will tend to become attached as semantic features of subsequent objects bound to the *buyer* role of a *sell-to* relation. We have yet to work out fully the details of this proposal, but preliminary simulations have so far been very promising.

References

Byrne, R. M. J., & Johnson-Laird, P. N. (1989). Spatial reasoning. *Journal of Memory and Language, 28*, 564-575.

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

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

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

Hofstadter, D. R., & Mitchell, M. (1994). An overview of the Copycat project. In K. J. Holyoak & J. A. Barnden (Eds.), *Advances in connectionist and neural computation theory, Vol. 2: Analogical connections*. Norwood, NJ: Erlbaum.

Holyoak, K. J., & Hummel, J. E. (2000). The proper treatment of symbols in a connectionist architecture. In E. Dietrich and A. Markman (Eds.). *Cognitive Dynamics: Conceptual Change in Humans and Machines*. Hillsdale, NJ: Erlbaum.

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

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

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

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

Medin, D. L., & Shoben, E. J. (1988). Context and structure in conceptual combination. *Cognitive Psychology*, *20*, 158-190.

Newell, A., & Simon, H. A. (1976). Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, *19*, 113-126.

Shastri, L., & Ajjanagadde, V. (1993). From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings. *Behavioral and Brain Sciences*, *16*, 417-494.

Smith, E. E. & Osherson, D. N. (1984). Conceptual Combination with prototype concepts, *Cognitive Science, 8*, 337-361.

Smith, E. E., Osherson, D. N., Rips, L. J. & Keane, M. (1988). Combining prototypes: A selective modification model, *Cognitive Science*, *12*, 485-527.

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.

Acknowledgments

This research was supported by NSF Grant SBR-9729023, and by grants from the UCLA Academic Senate and HRL Laboratories.