

Analogical Mapping by Constraint Satisfaction

KEITH J. HOLYOAK

University of California, Los Angeles

PAUL THAGARD

Princeton University

A theory of analogical mapping between source and target analogs based upon interacting structural, semantic, and pragmatic constraints is proposed here. The structural constraint of *isomorphism* encourages mappings that maximize the consistency of relational correspondences between the elements of the two analogs. The constraint of *semantic similarity* supports mapping hypotheses to the degree that mapped predicates have similar meanings. The constraint of *pragmatic centrality* favors mappings involving elements the analogist believes to be important in order to achieve the purpose for which the analogy is being used. The theory is implemented in a computer program called ACME (Analogical Constraint Mapping Engine), which represents constraints by means of a network of supporting and competing hypotheses regarding what elements to map. A cooperative algorithm for parallel constraint satisfaction identifies mapping hypotheses that collectively represent the overall mapping that best fits the interacting constraints. ACME has been applied to a wide range of examples that include problem analogies, analogical arguments, explanatory analogies, story analogies, formal analogies, and metaphors. ACME is sensitive to semantic and pragmatic information if it is available, and yet able to compute mappings between formally isomorphic analogs without any similar or identical elements. The theory is able to account for empirical findings regarding the impact of consistency and similarity on human processing of analogies.

INTRODUCTION

At the core of analogical thinking lies the process of *mapping*: the construction of orderly correspondences between the elements of a source analog and those of a target. Identifying an appropriate mapping is crucial in allow-

The research reported in this article was supported by Contract MDA903-86-K-0297 from the Army Research Institute. The Connection Machine implementation of the program was supported by NSF Biological Facilities Award BBS 87-1420, and by a grant from the Keck Foundation. An initial version of the work described here was presented at the meeting of the Society for Philosophy and Psychology held at the University of California, San Diego, on June 21, 1987.

We are grateful to Greg Nelson for graphics programming. David Gochfeld and Greg Nelson developed examples used to test the Connection Machine implementation, and Eric Melz reimplemented the program in *LISP. We thank John Bransford, Dedre Gentner, Jim Greeno, Doug Hofstadter, Greg Nelson, Laura Novick and David Rumelhart for helpful comments on earlier drafts, and Jay McClelland for useful advice on activation functions.

Correspondence and requests for reprints should be sent to K.J. Holyoak, Department of Psychology, UCLA, Los Angeles, CA 90024.

ing useful transfer of knowledge. In this article a theory of analogical mapping based upon a small set of constraints is provided and a cooperative algorithm that allows the graceful integration of these constraints is described. The algorithm is implemented in a computer program that computes mappings between symbolic representations. The results of a number of applications of the computational model to a variety of analogies will be presented. The theory and implementation here are similar in many respects to those of Gentner and colleagues (Falkenhainer, Forbus, & Gentner, 1986; Gentner, 1983), but also differ in significant ways. Gentner has emphasized the importance of structural constraints in determining the correspondences between two analogs, but it is maintained here that semantic and pragmatic constraints must also be taken into account.

In order to formulate a theory of mapping, it is necessary to consider the relationship between mapping and other aspects of analogical thinking. The centrality of mapping is a point of general agreement among all theorists who have discussed the use of analogy, whether in problem solving (Carbonell, 1983, 1986; Gick, & Holyoak, 1980), in explanation (Gentner, 1983), in case-based reasoning (Hammond, 1986; Kolodner, Simpson, & Sycara, 1985) in theory formation (Darden, 1983; Thagard, 1988a), in the analysis of formal systems (Hesse, 1966; Polya, 1973), or in metaphor and other literary uses (Black, 1962; Gentner, 1982; Holyoak, 1982; Miller, 1979). There has been less agreement, however, on the relationship between mapping and other subprocesses of analogy, and on the related issue of whether a common set of principles governs mapping across different uses of analogies.

The view here is that analogy, and inference in general, must be understood pragmatically, taking into account the goals and purposes of the cognitive system (Holland, Holyoak, Nisbett, & Thagard, 1986; Holyoak, 1985). As many theorists have noted, it is useful to decompose analogy into four major components: (1) the retrieval or selection of a plausibly useful source analog, (2) mapping, (3) analogical inference or transfer, and (4) subsequent learning. In this article, the important issue of learning in the aftermath of analogy use will be set aside in order to focus on mapping and the components that immediately surround it: selection and transfer. These three subprocesses must collectively serve three crucial functions: picking out a plausibly useful source analog, identifying elements of the source that should determine transfer to the target, and effecting such transfer.

Is there, in fact, a general purpose mapping component that operates in fundamentally the same way for different varieties of analogy, and if so what role does it play in this overall task? This question can be addressed, indirectly, by examining the functions performed by the subprocesses of selection and transfer, and then considering what remains. Clearly, the selection component is crucial to the success of analogy. Spontaneous retrieval of a relevant source analog depends upon the presence of similar elements in the source and target (Gentner & Landers, 1985), including (in the

case of problem analogs) similar constraints and goals (Brown, Kane, & Echols, 1986; Holyoak & Koh, 1987). In the absence of clear similarities, useful analogies are often missed (Gick & Holyoak, 1980); if misleading surface similarities are present, false analogies may be accessed and lead to negative transfer (Novick, 1988).

Once a possible source analog is retrieved spontaneously or provided by a teacher, further selection must be made of the aspects of the source relevant to the analogy. Analogies are virtually always used to serve some known purpose, and the purpose will guide selection. If, for example, one is simply asked to compare what is known about Nicaragua with what is known about Cuba, all elements of the two representations are relevant. But if one is asked to assess likely political trends in Nicaragua by analogy to Cuba, then only a subset of what is known about Cuba—roughly, facts, which bear upon the development of its political system—need be mapped. For example, it is relevant to consider the degree to which Nicaragua's Daniel Ortega resembles Cuba's Fidel Castro. In contrast, suppose one is asked to predict the suitability of Nicaragua for sugar cane production, again by analogy to Cuba. The subset of knowledge about the source that is likely to be mapped will be very different—the similarity of Nicaragua to Cuba in terms of temperature and rainfall will loom much larger when the question concerns agriculture rather than politics. In examples such as these, the selection process can use pragmatic knowledge about the purpose of the analogy to identify not only a relevant source analog, but also *which aspects* of the source are important in the context. Much of the work of identifying aspects of the source that will determine transfer to the target can be done prior to mapping, based upon knowledge of the purpose of the analogy coupled with causal knowledge concerning the source.

Similarly, knowledge can be brought to bear on the transfer process *after* mapping has established correspondences between elements of the source and target. The mapping implicitly defines a set of inferences that could be made about the target, based upon correspondences with predicates and objects in the source domain. Thus if predicate P and object O in the source map onto P' and O' in the target, and the proposition $P(O)$ holds in the source, then the proposition $P'(O')$ can be constructed as a candidate inference about the target. Whether a candidate inference will in fact be seriously considered as a plausible hypothesis about the target will depend upon such pragmatic factors as whether the inference is relevant to the analogist's goals in using the analogy and whether the inference is consistent with what is already known about the target domain.

Given what functions can reasonably be ascribed to the selection and transfer components of analogy, it appears that the central task of the mapping component is to take as inputs representations of a target analog and a plausibly relevant source, augmented if possible with information about the apparent pragmatic importance of elements of each analog, and to compute

a set of correspondences between elements of the source and target that is likely to yield useful candidate inferences. Given that the analogist will often have imperfect knowledge of either or both analogs, a robust mapping process should be capable of operating even in the face of some uncertainty about what aspects of the two analogs are in fact most central for effective transfer.

A CONSTRAINT-SATISFACTION THEORY

The fundamental problem of analogical mapping is how to find appropriate correspondences between two analogs. If the analogs each have m predicates and n constants, and it is assumed that predicates map only to predicates and constants to constants, and that the mapping is one-to-one, then there are $m!n!$ possible mappings from which to select. Thus a typical analogy between analogs with 10 predicates and 5 constants each generates over 400 million possible mappings. Efficient selection of the best mapping requires that some constraints be placed upon what it might be. This problem is similar to that of stereoscopic vision (Marr & Poggio, 1976). Stereopsis requires that points in two visual images, one from each eye, be appropriately paired; however, there is no a priori basis for uniquely deciding which point should go with which. Similarly, given representations of two complex analogs, there is no a priori basis for establishing a determinate set of correspondences between elements in the two analogs. In order to account for stereopsis, Marr and Poggio proposed several qualitative constraints on the visual system. These constraints lead to the emergence of a unique set of point-to-point pairings, with each pairing consisting of points in each image arising from the same spatial position in the environment.

Numerous models of analogical mapping have been proposed by researchers in cognitive psychology and artificial intelligence, and a thorough review will not be attempted here (see Hall, 1989; Thagard, 1988b). Three classes of constraints tend to recur in theoretical treatments of analogy: structural,¹ semantic, and pragmatic. After discussing these constraints, a set of principles governing analogical mapping will be proposed.

¹ Throughout this article the term "structural" is used to refer to "structural consistency" between two analogs, following the terminology of Falkenhainer et al. (1986). This sense of "structural" is to be distinguished from that used by Holyoak (1985) and Holyoak and Koh (1987), who defined "structural" properties as the goal-relevant aspects *within* a single analog. Structural properties in the latter sense will be termed "pragmatically central" or simply "important" properties in the present paper. A pragmatically useful analogy is one in which structural consistency holds between the important properties of the source and target. Use of different senses of the term "structural" in the analogy literature has contributed to some theoretical misunderstandings (e.g., Gentner, 1989; Holyoak, 1985).

Structural Consistency

Many theorists, particularly Gentner (1983), have stressed the importance of consistent structural correspondences as a criterion for an intuitively satisfying analogical mapping (Burstein, 1986; Falkenhainer et al., 1986; Gick & Holyoak, 1980; Hofstadter, 1984; Winston, 1980). Loosely speaking, a source analog can serve as a model for the target if objects in those two analogs can be placed into correspondence so that relations also correspond. A formal definition of structural consistency can be developed in terms of the concept of a *morphism*. Essentially the same characterization of structural consistency has been adopted in many different contexts, including formal model theory (Tarski, 1954), mathematical category theory (MacLane, 1971), and the theory of simulation (Ziegler, 1976). Within psychology, the concept of a morphism underlies the theory of measurement (Coombs, Dawes, & Tversky, 1970; Suppes & Zinnes, 1963), as well as theoretical treatments of mental representation (Halford, 1987; Halford & Wilson, 1980; Palmer, 1978), mental models (Holland et al., 1986), and analogy (Holyoak, 1984, 1985; Indurkha, 1987; Palmer, 1989). In the case of analogy, let T be an ordered n -tuple representing a target analog consisting of a set of objects, O , and some number, n , of relations on O ,

$$T = \langle O, R_1, R_2, \dots, R_n \rangle.$$

The representation of the source can similarly be defined as

$$S = \langle O', R'_1, R'_2, \dots, R'_n \rangle.$$

Let m be a mapping function that takes objects and relations in the target into objects and relations in the source,

$$m: o_i \rightarrow o'_i; R_i \rightarrow R'_i.$$

The mapping function m defines a valid isomorphism if, and only if, the mapping is one-to-one and for any objects and relations in T and S

$$o_i R_k o_j \text{ implies } m(o_i) m(R_k) m(o_j). \quad (1)$$

A valid analogy, A , is thus an ordered triple consisting of the relational systems T and S and a mapping function m with the above properties,

$$A = \langle T, S, m \rangle.$$

Figure 1 depicts the essence of a structurally valid analogy. Relations in the two analogs and the mapping function m are represented by directed lines. The consistency requirement expressed in (1) corresponds to the property of *commutativity of the diagram*: Following the arc representing the relation R_k between o_i and o_j in the target and then applying m produces the same result as first applying m to o_i to arrive at o'_i and then following the arc representing R'_i between o'_i and o'_j in the source.

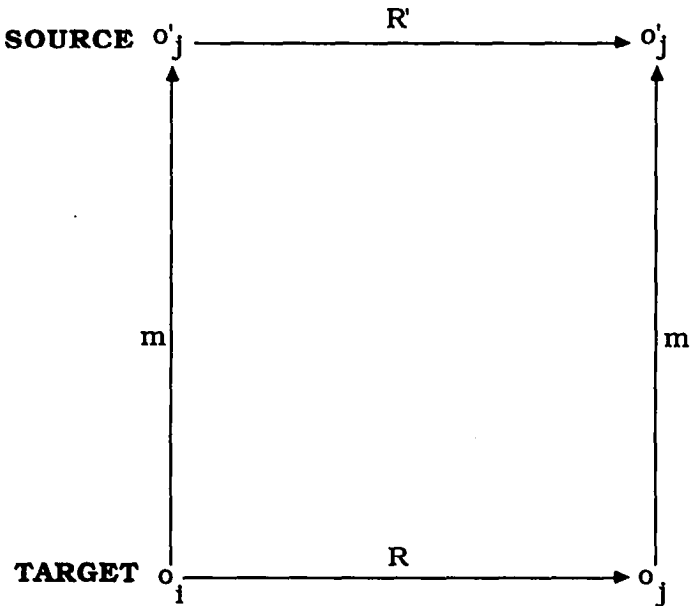


Figure 1. Representing a structurally valid analogy as an isomorphism

Although (1) is stated for relations with two arguments, the basic principle can readily be generalized to n -place predicates (including $n = 1$, as an example below will illustrate). In more general terms, a proposition P in the target is in correspondence to a proposition P' in the source if, and only if, the predicate and arguments of P are mapped into the predicate and arguments of P' by a function m that leads to a structurally consistent analogy A . Note that the consistency requirement in (1) implies that it is not generally possible to decide whether any pair P and P' are in correspondence without considering the entire set of correspondences between propositions in T and S . This interdependence inherent in the constraint of structural consistency poses one of the major problems that must be solved by a computational model of mapping.

The strict formal definition of an isomorphism is clearly inappropriate as a characterization of the kinds of analogies of psychological interest, which virtually never have the structure of a valid isomorphism. Rather, some elements of the target may have no apparent corresponding element in the source (or vice versa); some correspondences may be many-to-one (a homomorphism) or one-to-many (violating the formal definition of a function); and the consistency requirement in (1) may occasionally be violated. None of these types of violations of the formal requirements for an isomorphism necessarily preclude the analogy being potentially useful to a human reasoner. Nonetheless, useful naturalistic analogies intuitively can be viewed as

approximations to isomorphisms. In order to characterize mental representations that violate the strict definition of an isomorphism, Holland et al. (1986) extended the concept to homomorphisms with exceptions, or *quasi-homomorphisms*. Similarly, the theory to be proposed here treats the structural constraint of isomorphism as an ideal that can be satisfied to some imperfect degree, rather than as an absolute requirement for a successful mapping.

Semantic Similarity

The formal definition of an isomorphism makes no reference to the similarity of the objects and relations involved in the two analogs. Consider, for example, analogies between linear orderings (Halford, 1987). "John is taller than Bill, and Bill is taller than Sam" is analogous to "Mary is heavier than Susan, and Susan is heavier than Beth," with mappings between John and Mary, Bill and Susan, Sam and Beth, and the relations "taller than" and "heavier than." In this case both the objects and the relations being mapped are relatively similar. However, an equally valid analogy holds between the former analog and "communism is more radical than socialism, and socialism is more radical than capitalism," with mappings between John and communism, Bill and socialism, Sam and capitalism, and "taller than" and "more radical than." Even though similarity of relations and objects is sharply reduced in the latter analogy, the degree of structural consistency is the same in both cases.

Various theorists have suggested, and empirical evidence confirms, that object and predicate similarity influence the mapping process, with high semantic similarity leading to greater ease of mapping (Gentner & Toupin, 1986; Holyoak & Koh, 1987; Ross, 1987; Winston, 1980). The question therefore arises whether semantic similarity should be viewed as a distinct constraint on mapping, or whether it can somehow be assimilated to more basic constructs. Object similarity can potentially be reduced to predicate similarity: two objects are similar to the extent they serve as arguments of similar predicates. Predicate similarity may in turn be analyzed in terms of feature overlap (Tversky, 1977). One possibility, therefore, is to assimilate semantic similarity to structural consistency by imposing an added restriction on the latter constraint: corresponding relations must either be identical (Falkenhainer et al., 1986)² or share common features, such as a common superordinate (Burstein, 1986; Winston, 1980). A requirement of strict identity between corresponding relations does not seem satisfactory as a psychological model, since people can readily find mappings involving non-

² Note that Falkenhainer et al. (1986) allow mappings between nonidentical one-place predicates and objects; they only impose the identity restriction on multiplace relations. Since Falkenhainer et al. represent functions as one-place predicates, SME can map nonidentical functions.

identical relations (Burstein, 1986). (A number of relevant examples will be presented below.) Restrictions stated in terms of feature overlap (thus allowing mappings between similar rather than just identical relations) are much more plausible. However, the prerequisite requirement to provide a satisfactory analysis of natural language concepts in terms of semantic features remains a formidable challenge.

In addition to the sheer difficulty of reducing semantic similarity to structural consistency, there is empirical evidence that the two types of constraints have distinct consequences. Semantic similarity has a more pronounced effect on the retrieval of a source analog than on the mapping process (Gentner & Landers, 1985; Holyoak & Koh, 1987). In addition, although judgments of the aptness or soundness of analogies and metaphors are positively correlated with structural consistency, they are negatively correlated with similarity (Tourangeau & Sternberg, 1982). People's judgments thus reflect the intuition that although analogies based upon similar objects and relations are easy to map, they are less interesting or esthetically pleasing than "deeper" analogies between disparate situations. These separable effects of structural consistency and semantic similarity motivate treating the two kinds of constraints as distinct.

Pragmatic Centrality

Another major type of constraint on mapping that many theorists have proposed involves the pragmatic importance of the elements of the two analogs. Some treatments have emphasized the centrality of causal knowledge in determining the most appropriate mapping (Hesse, 1966; Winston, 1980); others have focused on the roles of high-level plans, goals, and functional knowledge (Anderson & Thompson, 1989; Burstein, 1986; Carbonell, 1983, 1986; Kedar-Cabelli, 1985). Although these models have important differences, they all share the view that the analogist uses explicit or implicit knowledge about the purpose the analogy is intended to serve to help direct the mapping process.

Although few would dispute that pragmatic knowledge influences the use of analogy, there remains disagreement as to the locus of its influence. Pragmatic considerations clearly weigh heavily in the initial selection of a plausibly useful source analog and in the subsequent transfer process. But do pragmatic considerations affect the mapping process itself? Gentner (1989, p. 215) proposes an architecture for analogy in which "plans and goals influence our thinking *before* and *after* the analogy engine" (i.e., the mapping mechanism) "but not during its operation." Gentner argues, very plausibly, that the goal of the analogist will typically have an impact on the representation of the target analog in working memory, which will in turn influence the retrieval of a source analog. Furthermore, the goal structure of a stored source problem may influence the mapping process indirectly by

affecting the degree of structural consistency between the source and the target.

There are reasons to suspect, however, that pragmatic considerations—the analogist's judgments about which elements of the analog are most crucial to achieve a useful mapping—may also have a more direct influence on the mapping process. As noted earlier, the general form of analogical transfer is to find correspondences among elements of the source and of the target, and then construct candidate inferences about the target by essentially copying over propositions from the source after substituting the appropriate corresponding elements from the target domain. This form of transfer is very flexible, and allows analogies to be used in an exploratory fashion to derive unanticipated candidate inferences about the target (Gentner, 1989). The cost of this flexibility, however, is that the inference process is not goal directed; there is no guarantee that candidate inferences constructed in this fashion will be relevant to the analogist's purpose in using the analogy. In many uses of analogy, such as problem solving and explanation, the analogist has an implicit or explicit question in mind when trying to derive a mapping, and therefore intends to use the source analog to derive inferences that will provide specific information about the target. For example, Holyoak and Thagard (1989) suggest that in analogical problem solving, people may aim to generate a mapping sufficient to transfer useful subgoals.

When the analogist is trying to make a particular type of inference or to answer a specific question, a more goal-directed form of transfer is possible. In particular, if the target representation contains variables representing missing knowledge that the analogy is intended to provide, then the analogist may selectively favor possible correspondences that would allow these variables to be appropriately instantiated. For example, if a person is trying to answer the question, "What was the likely cause of the stock market crash of 1987?" by mapping the circumstances of that year to those involved in the crash of 1929, then it would be useful to favor correspondences that allow instantiation of the unknown cause of the crucial event in the target over correspondences that could not do so. Mappings guided by such questions will in effect generate goal-relevant candidate inferences directly, rather than depending upon the unconstrained generation and assessment of all possible inferences in the aftermath of mapping.

In addition to considerations of computational efficiency in guiding transfer, there is some suggestive empirical evidence indicating that pragmatic knowledge may influence the mapping process directly, rather than solely by affecting the representations over which the mapping is derived. Brown et al. (1986) found that young children who were directed to attend to the goal structure of problems were better able to transfer solutions to analogous problems than were children who were not so directed. These investigators found, however, that the latter children were nonetheless able to

recall the critical goal elements when directly asked to do so. Their poor transfer performance thus seemed not to be due to simple failure to remember the goal elements (i.e., failure to include the goal elements in their representations of the source problems). Rather, the difficulty seemed related to selective attention, involving failure to view the goal elements as especially important during the mapping process. The theory to be proposed here assumes that the judged pragmatic importance of elements of analogs can directly constrain the mapping process, in addition to influencing earlier and later stages in the use of analogy.

Statement of Theory

The theory here assumes, as do many others, that analogical mapping can be viewed as a process of finding correspondences between elements of two structures. In propositional representations, the elements will include propositions, predicates, and objects. In other kinds of representations, such as pictorial ones, different types of elements may have to be mapped.

The major assertion of the theory is that mapping is governed by constraints of the three basic types discussed earlier: structural, semantic, and pragmatic. None of these constraints is absolute; rather, they provide "pressures" (in the sense of Hofstadter, 1984) that guide the emergence of a global mapping as a consequence of numerous local decisions about element correspondences.

Constraint 1. The structural constraint of *isomorphism* favors mappings that satisfy the consistency criterion in formula (1) and are one-to-one. Structural consistency requires that if a proposition in the target corresponds to a proposition in the source, then the constituent predicates and arguments of the paired propositions should also correspond. One-to-one mapping requires that each target element should correspond to only one element of the source, and that no two target elements should correspond to the same source element.

Constraint 2. Semantic similarity supports possible correspondences between elements to the degree that they have similar meanings.

Constraint 3. Pragmatic centrality favors correspondences that are pragmatically important to the analogist, either because a particular correspondence between two elements is presumed to hold, or because an element is judged to be sufficiently central that *some* mapping for it should be found.

The similarities and differences between the present theory and the structure-mapping theory developed by Gentner and colleagues (Falkenhainer et al., 1986; Gentner, 1983, 1989) can be characterized in terms of the above principles. The basic assumption that mapping involves finding structural correspondences is common to both approaches, as well as to virtually all

other theoretical treatments. The isomorphism constraint here embodies the principles of structural consistency and one-to-one mapping employed by Falkenhainer et al.; however, in the present theory these principles are interpreted as pressures rather than as requirements. Isomorphism is also related to Gentner's *systematicity principle*, which states that mappings between higher-order relations (i.e., predicates such as "cause" and "implies" that take propositions as arguments) constrain mappings between first-order relations (i.e., predicates such as "kill" that take objects as arguments), which in turn constrain object mappings. Gentner's systematicity principle characterizes one important type of information that can be used to identify isomorphic structures.

The constraint of semantic similarity provides a stronger distinction between Gentner's theory and the one described here, which postulates semantic pressures interacting with the isomorphism constraint. By treating semantic similarity as a pressure distinct from isomorphism, the restriction that multiplace relations must be identical in order to be mapped is eliminated. As a consequence, this theory is able to provide a mechanism for mapping purely formal analogies that lack any identical or even similar relations. At the same time, the mapping process is guided by similarity information if it is available. Nonetheless, this difference between the two approaches should not be overstated. If identity of predicates is understood as a limiting case of semantic similarity, then the semantic similarity constraint here can be viewed as a weakening of Gentner's principle of finding correspondences between identical relations. The mapping model described by Burstein (1986) also allows mapped relations to be similar rather than identical. In the present theory, degree of similarity leads to preferences, rather than strict requirements, in identifying optimal correspondences.

The constraint of pragmatic centrality presented here departs most clearly from Gentner's theory, which maintains that people use only structural information in mapping. Gentner's approach emphasizes that attention to pragmatic considerations is restricted to stages of analogical reasoning occurring before and after the mapping process.

Further comparisons of the theory presented here with that of Gentner can be made at the level of the computational *implementations* of these theories. Below, the implementation of the theory in the program ACME will be contrasted with the implementation of Gentner's theoretical approach in the SME program (Falkenhainer, Forbus, & Gentner, 1986, in press).

The mapping component is of course only one piece of an overall processing system for analogical reasoning. In addition to a natural-language interface, prior processes of analogical retrieval and selection are assumed that (a) propose a plausible source-target pair, and also may optionally provide (b) information about the degree of semantic similarity between pairs of source-target predicates, and (c) information about the pragmatic importance of elements of each analog. The present theory makes no assumptions

about the nature of the processes that compute similarity and importance information. The similarity computation may be based upon decomposition of meanings into identities and differences (Hesse, 1966; Tversky, 1977); the importance computation may be based upon some form of causal or explanation-oriented analysis (Kedar-Cabelli, 1985), prior expectations, or instruction from a teacher. For the present purposes it will simply be assumed that the mapping component can receive a numerical index of the degree of semantic similarity between two predicates and of the pragmatic centrality of elements of the analogs. In general, this theory of mapping can be stated independently of any strong theory of similarity, memory retrieval, causal analysis, or of other subprocesses of analogical inference. The theory thus defines a mapping mechanism that can be potentially integrated within broader theories describing additional stages of analogical reasoning.

ACME: A COOPERATIVE ALGORITHM FOR MAPPING

The algorithm for evaluating mappings is suggested by Marr and Poggio's (1976) treatment of stereoscopic matching, which was based upon a *cooperative* algorithm, "...so-called because of the way in which local operations appear to cooperate in forming global order in a well-regulated manner" (Marr, 1982, p. 122). A cooperative algorithm is a procedure for parallel satisfaction of a set of interacting constraints. In the Marr and Poggio algorithm, a network of nodes is established, in which each node represents a possible pair of matched points, and excitatory and inhibitory connections between nodes represent constraints. The network is then allowed to run in order to find a globally optimal set of match hypotheses.

More generally, Marr (1982) argued that cooperative methods capture two principles that appear to govern fluent information processing: (1) the principle of *graceful degradation*, according to which degrading the input data should allow computation of a partial answer, and (2) the principle of *least commitment*, which requires avoiding doing something that may later have to be undone. Theorists working within the connectionist framework have argued that cooperative methods may be applicable to human memory retrieval and higher level reasoning (Rumelhart, Smolensky, McClelland, & Hinton, 1986). Several properties of an information-processing task can provide cues that a cooperative algorithm may be appropriate. A cooperative algorithm for parallel constraint satisfaction is preferable to any serial decision procedure when: (a) a global decision is composed of a number of constituent decisions, (b) each constituent decision should be based upon multiple constraints, (c) the outcome of the global decision could vary depending upon the order in which constraints are applied and constituent decisions are made, and (d) there is no principled justification for preferring

any particular ordering of constraints or of constituent decisions. (For a philosophical discussion of the importance of parallel computation, see Thagard, 1986.)

Analogical mapping using constraints exhibits all of these features. Hofstadter (1984) pioneered the use of a cooperative process model for analogical mapping. In the Copycat model, a global mapping emerges from the parallel evaluation of evidence for interacting local hypotheses about element correspondences. Similarly, a cooperative algorithm for mapping analogies has been formulated and implemented here into a COMMON LISP program called ACME (Analogical Constraint Mapping Engine). ACME constructs a network of units representing mapping hypotheses and relaxes into a state representing the best mapping.

Inputs to ACME

In order to apply the isomorphism constraint, a mapping model must have input representations rich enough to distinguish (a) among predicates such as *dog*, constants representing objects such as *Fido*, and identifiers representing propositions such as *dog (Fido)*, and (b) between predicates with different numbers of arguments. For example, *cow* can be represented as a one-place predicate taking one argument, as in *cow (Bossy)*, whereas *loves* is a two-place predicate taking two arguments, as in *loves (John, Mary)*. The algorithm described below, like the SME mapping program of Falkenhainer et al. (1986) takes as input structures consisting of sets of sentences in predicate calculus. No particular devotion to predicate calculus as a representation language (Thagard, 1984) is maintained here; it is used because of its simplicity and familiarity. Other more complex representation languages should be amenable to similar treatment.

Several more specific representational assumptions also deserve mention. Propositions may have hierarchical structure, with some predicates taking propositions as arguments. For example, *cause* is treated as a two-place predicate with propositions representing events or states as its arguments (Gentner, 1983). Functions of n arguments are treated as relations of $n + 1$ arguments holding among n objects and a value. Thus *height (John, 6 feet)* represents the fact that the height of John has the value six feet.

In order to represent queries that serve to focus attention on the pragmatically central aspects of a target analog, ACME allows some arguments to be variables marked with a question mark, as in "value?". Two kinds of queries are distinguished between here:

1. *Cross-structure* queries indicate that the answer to the query should come from seeing the relevant correspondences in the other analog. For example, if I tell you that Ortega is like Castro, and ask you what Ortega's political views are, then you could answer by transferring Castro's political views over to Ortega.

2. *Internal* queries indicate that the answer to a query is to be found within the structure that contains the query, with some element already present in the structure answering the query. In comparing the stock market crash of 1987 to the crash of 1929, you may expect to map a structure describing the 1929 crash to a structure describing the 1987 crash. In asking what caused the latter, you would expect to find some element of the 1987 structure that could provide the cause. This case differs from a cross-structure query in that you do not expect some element of the 1929 situation to answer your question for you directly. Rather, you hope that one result of mapping the 1987 and 1929 situations will be to fill in the relevant element in the 1987 structure using another component of that structure. ACME uses two question marks to indicate internal queries. For example, *cause (?proposition?, crash-of-'87)* represents the question, "What aspect of the situation was the cause of the crash of 1987?" As described below, internal queries are treated as variables that allow support to be passed to specific elements that might fill the role of the variable.

Each possible hypothesis about a possible pairing of an element from the source with a corresponding element of the target is assigned to a node or *unit*. Each unit has an *activation level*, ranging between some minimum and maximum values, which indicates the plausibility of the corresponding hypothesis, with higher activation indicating greater plausibility. Inferential dependencies between mapping hypotheses are represented by *weights* on *links* between units. Supporting evidence is given a positive weight, and disconfirmatory evidence is given a negative weight.

Setting Up a Mapping Network

The operation of the ACME program will now be described in greater detail. Figure 2 provides a schematic illustration of the kind of input that is provided to the program and of the kind of mapping network that the program builds. The input to the program consists of predicate-calculus representations of the source and target analogs, plus optional information about semantic similarity and pragmatic importance. It is assumed that a mapping may be computed either from a target analog to a source or vice versa. It is conjectured that the direction of mapping will vary depending upon the use of the analogy and the knowledge of the analogist. If the source is much more familiar than the target, then it may be best to try to map source elements to target elements. On the other hand, if the source is much more complicated than the target or if the target contains highly salient elements, then the analogist may attempt to map from the target to the source. In the example in Figure 2, the target is mapped to the source.

When given two structures as input, ACME automatically generates a network in accord with the constraints postulated by the theory. Each of the

INPUT

TARGET	SOURCE	Semantic Weights For:	Pragmatic Information:
T1 A(a)	S1 M(m)	A=M C=O	IMPORTANT D
T2 B(b)	S2 N(n)		
T3 C(a,b)	S3 O(m,n)		
T4 D(b,a)	S4 P(n,m)		

PARTIAL RESULTING NETWORK

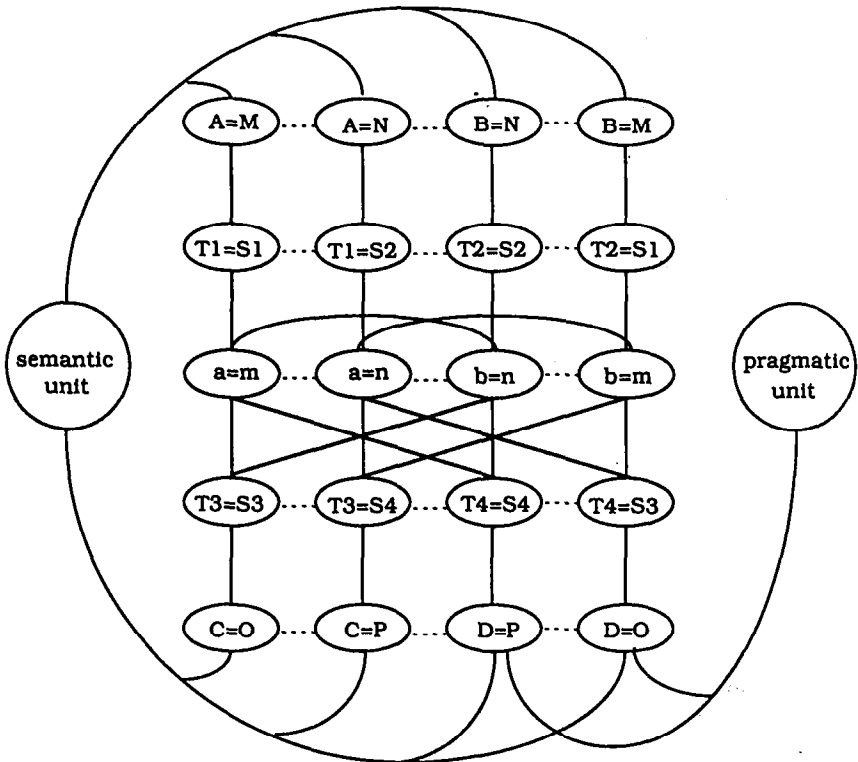


Figure 2. A schematic example of an ACME mapping network. Numbered capital letters represent proposition identifiers, unnumbered capital letters represent predicates, and lowercase letters represent objects. Solid lines represent excitatory connections and dotted lines represent inhibitory connections. (See text for further explanation.)

analogs in the simple abstract example depicted in Figure 2 includes four propositions, 2 one-place predicates, 2 two-place predicates, and 2 objects. The first step in building a mapping network is to construct *mapping units* corresponding to each possible hypothesis about pairings between elements. Two restrictions are used to limit the number of units formed. First, the

type restriction limits possible correspondences to those between elements of the same basic type: propositions to propositions, n -place predicates to n -place predicates, and objects to objects.³ Second, the *part-correspondence* restriction allows pairings only between propositions within corresponding major parts of the two analogs. The latter restriction only applies if the analogs are in fact divided into major parts. In the examples dealt with in this article, the part-correspondence restriction is only implemented for problem analogies. Problems can be divided into the basic parts of initial state, goal state, solution constraints, operators, and solution steps (Carbonell, 1983; Newell & Simon, 1972). ACME assumes that propositions describing initial states map to propositions describing initial states, propositions describing goals to propositions describing goals, and so on. Similarly, it would be possible to restrict mappings involving story structures to propositions within such corresponding major parts as the setting and resolution.

Neither of these restrictions on unit formation in the current implementation of ACME should be considered definitive. The type restriction can be viewed as a means of setting a low threshold of minimal similarity that is required in order to consider setting up mapping unit linking two elements. The part-correspondence restriction can be viewed as a type of serial application of the isomorphism constraint on mapping, in which prior correspondences between high-level parts are used to constrain more detailed correspondences. Possible modifications of these restrictions will be discussed in the General Discussion.

The mapping units in Figure 2 thus represents all possible pairings of elements of the same basic type. For each two-place predicate in the target, for example, units are established for each possible pairing with a two-place source predicate (e.g., $C = O$ and $C = P$). The part-correspondence restriction is not illustrated in this simple example. Units corresponding to pairings between one-place predicates (e.g., $A = M$), pairings between objects (e.g., $a = m$), and pairings between propositions (e.g., $T1 = S1$) are also constructed.

As the units are established, links are formed between them to implement the constraint of structural consistency. All links are symmetrical, with the same weight regardless of direction. For example, excitatory links (represented by solid lines) are formed between $T1 = S1$ and $A = M$, $T1 = S1$ and

³ ACME does not distinguish between object constants and variables representing classes of objects; hence the type restriction does not preclude mapping an object to a variable. Also note that mapping units may sometimes be formed for propositions with unequal numbers of predicates if such a mapping is suggested by a mapping of higher-order relations. For example, if *cause* ($T1$, $T2$) were mapped to *cause* ($S3$, $S2$), then the mapping unit $T1 = S3$ would be formed even if $T1$ has one argument and $S3$ has two arguments. In this case, however, the type restriction would preclude forming a mapping unit linking the predicates of $T1$ and $S1$.

$a = m$, and $A = M$ and $a = m$. All excitatory links have a default excitation weight given by parameter e . For predicates that take more than one argument, the argument mappings support each other, as in the link between $a = m$ and $b = n$ which is suggested by both $T3 = S3$ and $T4 = S4$. Weights on links are additive, so that the weight between $a = m$ and $b = m$ will be double the value of e . Links are also created between predicate-mapping units and their corresponding argument-mapping units. Each potential correspondence between propositions thus generates an interconnected subnetwork of mutually consistent correspondences among elements of the propositions.⁴ Figure 2 only shows a subset of the links that would be formed for this example. Figure 3 depicts the full set of excitatory links that would be created in the course of mapping two propositions with two-place predicates, $T3$ and $S3$.

After all the units have been formed, inhibitory links (represented in Figure 2 by dashed lines) with weights equal to parameter i are formed to connect all units that represent alternative mappings for the same element.

⁴ In an earlier version of ACME, propositions were not mapped directly using proposition identifiers; rather, mapping units were based solely upon correspondences between predicates and between their arguments. Structural consistency was enforced by placing excitatory connections from *predicate* T =*predicate* S units to *object* T =*object* S units. In many instances this simpler architecture is satisfactory. However, it leads to error in certain cases. For example, consider the following fragments from target and source analogs:

T1: D(a, b)	S1: M(x, y)
	S2: N(x, z)
	S3: N(w, y).

Suppose that other evidence provides high activation to the object-mapping units $a = x$ and $b = y$. Intuitively, this information should suffice to establish that predicate D maps to M rather than to N . However, without mapping units for the proposition identifiers (e.g., $T1 = S1$), ACME erroneously would provide slightly higher activation to $D = N$ than to $D = M$. The reason is that although excitatory links will be formed from $a = x$ and $b = y$ to $D = M$ (based upon the possible mapping of $T1$ and $S1$), these winning object-mapping units will also each have excitatory links to $D = N$. A link from $a = x$ to $D = N$ will be formed on the basis of the possible correspondence of $T1$ with $S2$, and a link from $b = y$ to $D = N$ will be formed on the basis of the possible correspondence of $T1$ with $S3$. Since, in addition, two possible proposition correspondences favor $D = N$ whereas only one favors $D = M$, the overall result is a preference for linking D to N rather than to the intuitively correct choice, M .

Such errors arise because the mapping network fails to capture a crucial aspect of structural consistency, which depends upon mapping propositions as integral units. The present architecture of ACME avoids errors of this sort by introducing proposition-mapping units which have excitatory connections both to the corresponding predicate-mapping units and to the object-mapping units, in addition to forming direct excitatory connections between the latter types of units. In the above example, the proposition-mapping unit $T1 = S1$ will be favored over its two competitors, which in turn ensures that $D = M$ is preferred to $D = N$. We thank Dawn Cohen for identifying a chemistry analogy that first brought this problem with the earlier architecture to our attention.

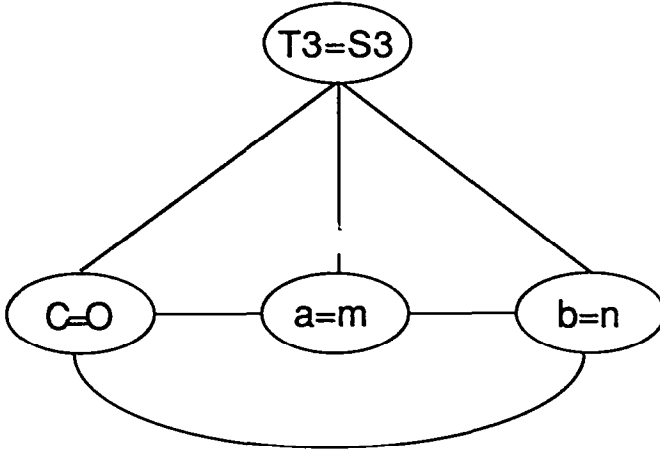


Figure 3. The subnetwork of excitatory connections among mapping units formed in the course of mapping propositions T3 and S3 (see Figure 2)

Thus the units $C=O$ and $C=P$ will be mutually inhibitory, as will $C=P$ and $D=P$. For clarity, Figure 2 depicts only a subset of all the links actually constructed, omitting, for example, the inhibitory links between $T2=S1$ and $T1=S1$.

In addition to the units representing mapping hypotheses, the network includes two special units. The *semantic unit* is used to convey information about the system's prior assessment of the degree of semantic similarity between each pair of meaningful concepts in the target and source, and the *pragmatic unit* similarly is used to convey information about the pragmatic importance of possible correspondences. The semantic-similarity constraint is enforced by placing excitatory links from the semantic unit to all units representing mappings between predicates. The weights on these links are made proportional to the degree of semantic similarity between the mapped concepts. Similarly, the pragmatic-centrality constraint is represented by weights on links connecting the pragmatic unit to relevant mapping units.

The list of semantic weights provides numerical values that reflect the degree of semantic similarity between target-source predicate pairs.⁵ Semantic similarity values, which range from a minimum value *min* representing

⁵ ACME represents objects by semantically empty constants; similarity of objects is represented indirectly by similarity of the mapped predicates that apply to the objects. It is possible, however, that people arrive at judgments of object-object similarity prior to the mapping stage, perhaps using additional information besides predicates available to the mapping process. The program could easily be extended to allow similarity weights to directly reflect object-object as well as predicate-predicate similarities.

no similarity to a maximum value *smax* automatically given to identical predicates, are given to ACME by statements of the form:

(SIMILAR <concept-1> <concept-2> <degree>).

If no similarity degree is given, then ACME assumes the minimum value. Thus in Figure 2, semantic weights are only explicitly provided for the units $A=M$ and $C=O$, which represent units linking concepts with some degree of similarity that lies between the minimum value and identity.

Important elements are indicated by inputs of the form

(IMPORTANT <element>)

which sets up links with weights equal to parameter $p1$ from the pragmatic unit to all units concerning the mapping of the element. Figure 2 shows D as an IMPORTANT element, so that there are excitatory links from the pragmatic unit to both of the units concerning how D could be mapped.

ACME also allows pragmatic support to be provided to a particular correspondence that the analogist may presume to hold. The input

(PRESUMED <hypothesis>)

sets up a mapping from the pragmatic unit directly to the unit representing that hypothesis, with a weight equal to parameter $p2$. Thus IMPORTANT provides support to the class of possible mappings for an important element, whereas PRESUMED provides support to a particular mapping hypothesis.

In addition to its use of pragmatic weights, ACME also represents pragmatic information related to questions for which an answer is sought by special query variables. Table 1 provides a summary of ACME's algorithm for forming a mapping network.

Running the Network

The manner in which the network is run to arrive at a solution is a straightforward application of constraint-satisfaction methods that have been investigated extensively in other applications (see Rumelhart et al., 1986). To initialize the network, the activation levels of the semantic and pragmatic units are fixed at 1, and the activations of all other units are set to some minimal value. On each cycle of activity, all units (except the semantic and pragmatic units) have their activation levels updated on the basis of the activation levels and weights associated with neighboring units and links. The updating procedure is based upon that suggested by Grossberg (1978). The activation level of unit j on cycle $t+1$ is given by

$$a_j(t+1) = a_j(t)(1-d) + enet_j(max - a_j(t)) + inet_j(a_j(t) - min)$$

where d is a decay parameter, $enet_j$ is the net excitatory input, and $inet_j$ is the net inhibitory input (a negative number), with $min = -1$ and $max = 1$.

TABLE 1

Summary of the ACME Algorithms for Forming a Mapping Network and
Performing Constraint Satisfaction

Consider a mapping between structures T and S. Let prop_{Ti} be the i th proposition of structure T, and let pred_{Ti} be the predicate in the i th proposition of T, and let arg_{Tik} be the object or proposition corresponding to the k th argument of the i th proposition in T, with analogous definitions of prop_{Sj} , pred_{Sj} , and arg_{Sjk} .

I. Setting Up A Mapping Network

- A. For each proposition prop_{Ti} in T, consisting of $(\text{pred}_{Ti} (\text{arg}_{Ti1} \text{arg}_{Ti2} \dots \text{arg}_{Tin}))$, for each proposition prop_{Sj} in S, consisting of $(\text{pred}_{Sj} (\text{arg}_{Sj1} \text{arg}_{Sj2} \dots \text{arg}_{Sjm}))$, if prop_{Ti} and prop_{Sj} are in the same part (e.g., goal) of their respective structure, and have the same number of arguments (i.e., $n=m$), then: (1) construct the units $\text{prop}_{Ti}=\text{prop}_{Sj}$, $\text{pred}_{Ti}=\text{pred}_{Sj}$, and each $\text{arg}_{Tik}=\text{arg}_{Sjk}$; (2) construct links between $\text{prop}_{Ti}=\text{prop}_{Sj}$ and $\text{pred}_{Ti}=\text{pred}_{Sj}$; (3) construct links between $\text{prop}_{Ti}=\text{prop}_{Sj}$ and each $\text{arg}_{Tik}=\text{arg}_{Sjk}$; (4) construct links between $\text{pred}_{Ti}=\text{pred}_{Sj}$ and each $\text{arg}_{Tik}=\text{arg}_{Sjk}$; (5) construct links between each pair of $\text{arg}_{Tik}=\text{arg}_{Sjk}$.
 Note 1: These are excitatory links with equal weights set by a default excitation parameter e .
 Note 2: A unit will not be formed if it would be redundant with a unit formed previously, but the weight on a link is incremented by e for each proposition that supports it.
- B. Construct inhibitory links between any two units that represent incompatible mappings, with a negative weight set by a default inhibition parameter l .
 Note 3: If a mapping unit connects an internal-query variable to some other element (i.e., $?query? = \langle \text{element} \rangle$), then construct excitatory (rather than inhibitory) links to other units representing possible mappings to $\langle \text{element} \rangle$.
- C. For each predicate-mapping unit $\text{pred}_{Ti}=\text{pred}_{Sj}$, construct a link from the semantic unit based on the degree of semantic similarity between the two predicates, with a weight ranging from a minimum value sm_{in} if there is no similarity to a maximum value sm_{ax} if the predicates are identical.
- D. For each element (predicate, object, or proposition) listed as IMPORTANT, construct a link from the pragmatic unit to each unit concerning a mapping for that element, with a weight equal to a parameter $p1$ for pragmatic centrality for IMPORTANT mappings.
- E. For each unit listed as PRESUMED, construct a link from the pragmatic unit with a weight equal to a parameter $p2$ for pragmatic centrality for PRESUMED mappings.
- To summarize, A (1-5) and B implement the structural constraint of isomorphism, C implements the semantic similarity constraint, and D and E (along with Note 3) implement the pragmatic centrality constraint.

II. Running the Network

The algorithm for synchronously updating the units in the network is:

- Clamp the activations of the semantic and pragmatic units at the maximum value.
- Set activations of all other units to an initial value (e.g., .01).

At each cycle,

1. For each unit u , calculate the new activation of u in accord with the equations in text, considering each unit u' linked to u .
 2. Set the activation of u to the new activation.
-

The value of $enet_j$ is equal to $\sum_i w_{ij} o_i(t)$ for $w_{ij} > 0$, and the value of $inet_j$ is equal to the same quantity when $w_{ij} < 0$. The quantity $o_i(t)$ is the output on cycle t of a unit with activation a_i , with $o_i(t) = \max(a_i(t), 0)$. On each cycle updating is synchronous across all units (see Table 1). Activation values are constrained to range from *min* to *max*.⁶

The degree to which the activation levels of units satisfy the constraints imposed by the weights on links is approximated by a measure termed G , defined as

$$G(t) = \sum_i \sum_j w_{ij} o_i(t) o_j(t).$$

The value of G can be interpreted as a rough index of the overall fit of the emerging mapping to the constraints of isomorphism, similarity, and pragmatic centrality.⁷

Comparison With Other Simulation Models of Analogy

As noted earlier, many previous simulation models of analogical mapping have been proposed (see Hall, 1989; Thagard, 1988b). Other models have included structural, semantic, and pragmatic constraints on mapping, but no single model has integrated these constraints as ACME does. The most closely related previous simulation is the SME program (Falkenhainer et al.,

⁶ Various alternative variants of the updating procedure were explored. Performance is improved for some examples if the minimum and maximum activation values are made asymmetric ($min = -.3$), as advocated by Grossberg (1978). The formula employed in McClelland and Rumelhart's (1981) model of word recognition was also used, in which the activation level of unit j on cycle t is given by

$$a_j(t+1) = a_j(t)(1-d) + \begin{cases} net_j(max - a_j(t)) & \text{if } net_j > 0 \\ net_j(a_j(t) - min) & \text{otherwise,} \end{cases}$$

where $min = -1$ and $max = 1$. The net input to unit j , net_j , is equal to $\sum_i w_{ij} o_i(t)$; $o_i(t)$ is the output on cycle t of a unit with activation a_i , with $o_i(t) = \max(a_i(t), 0)$. The Grossberg rule considers excitatory and inhibitory inputs separately in adjusting activations, whereas the McClelland and Rumelhart rule sums all inputs before making the adjustment. Although both rules yield similar results for most of the examples, the Grossberg rule proved more effective in some of the more complex cases.

For both rules it proved important to impose a zero threshold on the outputs of units, so that units with negative activations do not influence the units to which they are connected. Without this restriction, two units with negative activation levels that have an inhibitory weight on the connection between them will excite each other, yielding counterintuitive results for some examples.

⁷ The formula for G used in ACME is a variant of that used by Rumelhart et al. (1986). The present algorithm operates only on internal weights, and does not involve any external inputs. In addition, outputs rather than activation values are used because of the introduction of a threshold on outputs. Links between units that both have negative activations thus do not affect the value of G . Although there is not a proof that the activation-updating procedure finds a local maximum of G , the measure has been found to be heuristically useful in interpreting the behavior of the program.

1986, in press). SME is designed as a "tool kit" for implementing different possible mapping rules; the comparison here is with SME operating with rules based upon Gentner's (1983, 1989) structure-mapping theory. ACME and SME have several important similarities; indeed, in many respects ACME can be characterized as an extension of SME, even though it operates differently. Both models provide content-independent accounts of the mapping process, and both derive a global "best" mapping from a set of constituent hypotheses about element correspondences (the mapping units of ACME and the "match hypotheses" of SME). Both programs operate on predicate-calculus representations of analogs, and both emphasize the role of proposition mappings in enforcing mappings between corresponding elements of the propositions.

In terms of the basic constraints, ACME's isomorphism constraint is a generalized version of the constraints of structural consistency and one-to-one mapping that are employed by SME. Two differences in ACME's treatment of structural constraints are notable. First, ACME treats isomorphism as a separate constraint from semantic similarity. Whereas SME requires multiplace relations to be identical in order to be mapped, ACME allows mappings between relations with no similarity beyond having the same number of arguments. ACME's more abstract version of the isomorphism constraint allows the program to compute mappings beyond the scope of SME. ACME can exploit its sensitivity to abstract isomorphism to *find* important similarities between predicates, rather than depending upon the similarities being precoded in the initial representations of the analogs. This creative aspect of analogy is not well captured by models of mapping that are more highly dependent upon preexisting similarities or identities.

As we noted earlier, Gentner's systematicity principle, which is implemented in SME, describes one major type of information that can be used to identify isomorphisms: mappings of higher-order relations constrain mappings of first-order relations, which in turn constrain mappings of objects. ACME is also sensitive to systematicity, although such information is viewed symmetrically (e.g., not only do relation mappings constrain object mappings, but object mappings constrain relation mappings). The systematicity principle can be viewed as a special case of the general use of interrelated propositions to identify isomorphisms. In both ACME and SME, interrelated first-order relations also convey information about isomorphism, especially with multiplace relations. (The greater the number of arguments in a pair of mapped propositions, the greater the information provided about argument mappings.) Indeed, ACME (but not SME) can potentially find a unique mapping using no information except patterns of semantically unrelated monadic predicates (attributes), if these are in fact sufficient to create unique assignments of objects to sets of attributes. (An example of a mapping based solely upon attributes will be presented in the

section on applications of ACME to formal isomorphisms.) Thus ACME's sensitivity to structure is by no means dependent upon higher-order relations.

A further difference between ACME and SME involves the "tightness" of structural constraints on mapping. SME begins the mapping process by identifying consistently mappable subsets of the analogs. Any violation of the strong restriction that mapped relations must be identical marks the limit of a consistently mappable subset. The program typically yields several such subsets, ranked in order of "goodness." In contrast, ACME treats the constraints of isomorphism, semantic similarity, and pragmatic centrality as pressures that operate in parallel to find a single mapping that best satisfies all of the converging and competing constraints to some (typically imperfect) degree. The program on any one run finds a single set of "best" mapping units (although relatively high activation levels on other units will convey information about possible alternative mappings). Whereas SME explicitly constructs global mappings and evaluates them, the global mappings selected by ACME are only implicitly determined by means of the links between mapping hypotheses.

More generally, ACME includes semantic and pragmatic constraints on the mapping component, as well as purely structural constraints. ACME prefers mappings between elements that are semantically similar, whereas SME excludes such information as relevant only to stages of analogy outside mapping. To implement the constraint of pragmatic centrality, ACME allows preferences for PRESUMED mappings and for mappings involving IMPORTANT elements. More subtly, it prefers mappings that have the potential to answer queries internal to a structure. Somewhat similarly, SME includes a preference for mappings that generate the greatest number of inferences; although not characterized as such by Falkenhainer et al. (1986), this preference might be construed as a pragmatic factor. However, the program does not consider whether the possible inferences are relevant to the goals of the analogist, as ACME does. In line with the emphasis on the pragmatic aspects of analogy, it is contended here that analogists will tend to prefer mappings that produce the inferences they are interested in, not inferences in general.

In its use of a connectionist architecture ACME has important similarities to the Copycat program developed by Hofstadter (1984; Hofstadter & Mitchell, 1988), which also derives the globally best analogy from the outcomes of parallel competitions among interacting hypotheses about element correspondences. However, the constraints embodied in Copycat, like those used in SME, explicitly exclude pragmatic considerations.

ACME, SME, and Copycat are all much more complex than the mapping schemes implicitly included in simulation models that do not operate on cross-domain analogies (or, in the case of Copycat, on novel relational intradomain correspondences). If one is modeling only analogies within a

highly structured domain, mapping can be virtually trivial (especially for small examples), since essentially the same predicates and arguments will exist in both domains. For example, in Anderson and Thompson's (1989) model of using analogies in programming LISP, each piece of LISP code is represented with an explicit form and function. Mapping consists merely of applying similar forms to code with similar functions. Work on case-based reasoning also tends to be restricted to intradomain analogies (e.g., Hammond, 1986; Kolodner & Simpson, 1988). If analogy is restricted to a single domain in which the same predicates and arguments turn up in both structures to be mapped, mapping becomes a very simple special case of the more complex processes used by ACME and SME for cross-domain analogies.

It should be noted that there are important ties between the theory's treatment of the basic constraints on mapping and ACME's use of an algorithm based upon parallel constraint satisfaction. In an abstract isomorphism, it is impossible to assess whether the mapping between any two propositions in the target and source is valid without considering the mappings of all other propositions, because the validity of any possible correspondence depends upon its relationship to the entire analogical structure. When the additional semantic and pragmatic constraints are included, the degree of interdependence between mapping hypotheses increases even further. Parallel constraint satisfaction is able to deal with this sort of extreme interdependence by allowing all local mapping decisions to emerge together in an incremental fashion.

APPLICATIONS OF ACME

Analogical reasoning can serve many different functions. Major contexts for analogy use include problem solving, when the solution to one problem suggests a solution to a similar one; argumentation, when similarities between two situations are used to contend that what is true in one situation is likely to be true in the other; and explanation, when a familiar topic is used to provide understanding of a less familiar one. In addition, analogical reasoning is also used to understand formal analogies of the sort found in mathematics, as well as metaphors, which can be employed to serve both explanatory and more aesthetic functions. Given the fundamental assumption that all uses of analogy involve the same basic mapping mechanism, it follows that a theory of analogical mapping should apply to a full range of examples.

Table 2 lists the principal analogies to which ACME has been applied, along with the number of units and links that were formed for each. All of these examples are discussed below, except for the chemical analogies presented in Thagard, Cohen, and Holyoak (1989). Because translation of analogies in natural language into predicate-calculus inputs is somewhat arbitrary, these applications do not constitute strict tests of the theory imple-

TABLE 2
Summary of Applications of ACME

Analogs	Number of Units	Number of Symmetric Links
Lightbulb/radiation problems (4 versions) (Holyoak & Koh, 1987)	169-192	1373-1773
Fortress/radiation problems (Gick & Holyoak, 1980)	41	144
Cannibals and missionaries/farmer's dilemma problems (Gholson et al., 1988)	144	973
Contras Inference	95	169
Politics inference (2 versions)	55-67	308-381
Water-flow/heat-flow explanation (2 versions) (Falkenhainer et al., 1986)	62-127	317-1010
Solar system/atom explanation (Falkenhainer et al., 1986)	93	733
Jealous animal stories (6 versions) (Gentner & Toupin, 1986)	125-214	1048-1783
Addition/union	162	1468
Attribute mapping	43	220
Midwife/Socrates (3 versions) (Kittay, 1987)	97-203	534-1702
Chemical analogies (8 different analogies) (Thagard et al., 1989)		

mented in ACME. Nevertheless, they show that ACME is applicable to a wide variety of analogies and is consistent with experimental results that reveal when analogical mapping is difficult for people.

In order to demonstrate that ACME can account for performance on a wide variety of analogies using a consistent set of parameters, all parameters were held constant across the entire set of applications. Unless otherwise stated, all runs employed the Grossberg updating rule with $min = -1$, $max = 1$, and $d = .1$. Weight parameters were $e = .1$, $i = -.2$, $smin = 0$, $smax = .1$, $p1 = .1$, and $p2 = .3$. Mapping units were initialized at activation = .01. Intermediate similarity weights will be noted in those examples when they were used. The pragmatic centrality parameters were only used in a subset of the examples. The sensitivity of the program to variations in the parameter set will be described later.

Problem Analogies

Convergence Analogies. ACME has been applied to a number of problem analogies involving the use of a "convergence" solution, in which several weak forces are applied simultaneously to a centrally located object in order to achieve the effect of a single large force (Gick & Holyoak, 1980, 1983; Holyoak & Koh, 1987). In experimental work using these materials, the target analog has typically been a "radiation problem," in which a doctor

must find a way to use a ray to destroy a stomach tumor without harming the surrounding tissue (Duncker, 1945). Holyoak and Koh (1987) compared the effectiveness of four alternative versions of a source analog based upon a "lightbulb problem," in which a broken filament in a lightbulb must be repaired. The four versions were defined by two types of variations. Surface similarity to the ray used in the target was varied by manipulating whether the force used in the lightbulb problem was a laser (highly similar to a ray) or an ultrasound beam (less similar). Similarity of problem constraints was also varied. The similar constraint in the source was the necessity to avoid breaking the fragile glass bulb surrounding the filament (analogous to avoiding injury to the tissue surrounding the tumor). The dissimilar constraint was that a force of sufficiently high intensity was not available. Table 3 presents predicate-calculus representations of the "laser/fragile-glass" version of the lightbulb problem and of the radiation problem. These, together with similar representations of the other lightbulb versions, were used as inputs to ACME. Each proposition in an analog is represented by a list consisting of a predicate, a list of arguments, and a name for the proposition. In the surface-similar versions ACME was given a similarity weight of .08 for the mapping unit *ray-source = laser*. Pragmatic-centrality weights were not used in any of these examples of problem analogies, since all elements of the representations were selected to be important. In each run the possible mapping hypotheses were limited by the part-correspondence restriction (i.e., goal elements must map to goal elements, and so on).

Because of their complexity, a figure showing the full network created using the above input, with its 192 units and 1773 links, cannot be presented. However, Figure 4, reproduced directly from a screen dump of a graphics program running with ACME, shows the nodes connected to a typical unit, *ray-source = laser*. Thick lines indicate excitatory links and thin lines indicate inhibitory links; the weight of the link is specified by a number midway in the line. Beneath each node is a truncated number indicating the activation of the named unit. The network can be browsed to show the connectivity of another node simply by clicking on that node.

Table 4 presents the output of ACME after mapping the radiation problem with the laser/fragile-glass version of the lightbulb problem. This and all other tables of outputs to be presented gives the optimal mappings obtained after the network has settled at an asymptote, defined as the first cycle at which no unit has its activation adjusted by more than .001. For this example the network settles after 31 cycles. The asymptotic value of G is also printed, along with the best mapping for each object, each predicate, and each proposition used as the argument of a higher-order relation. (No propositions appear as arguments in this example.) The program defines the "best" mapping of an element as the corresponding mapping unit with highest activation level, regardless of absolute magnitude; however, it is natural to interpret cases in which the most active unit is very weak (e.g.,

TABLE 3
Predicate-Calculus Representations of Lightbulb Problem
(Laser, Fragile-Glass and Insufficient-Intensity Versions)
and Radiation Problems

LIGHTBULB PROBLEM (source)

Start: (laser (obj_laser) f1)
 (bulb (obj_bulb) f2)
 (filament (obj_filament) f3)
 (surround (obj_bulb obj_filament) f4)
 (outside (obj_laser obj_bulb) f5)

For good-constraint version, add:

(can-produce (obj_laser obj_beams_high) fg6)
 (high-intensity (obj_beams_high_ obj_filament) fg8)
 (can-destroy (obj_beams_high obj_bulb) fg9)
 (can-produce (obj_laser obj_beams_low) fg10)
 (low-intensity (obj_beams_low) fg11)
 (cannot-fuse (obj_beams_low obj_filament) fg12)
 (cannot-destroy (obj_beams_low obj_bulb) fg13)

For poor-constraint version, add instead:

(cannot-produce (obj_laser obj_beams_high) fp6)
 (high-intensity (obj_beams_high) fp7)
 (can-fuse (obj_beams_high obj_filament) fp8)
 (can-produce (obj_laser obj_beams_low) fp10)
 (low-intensity (obj_beams_low) fp11)
 (cannot-fuse (obj_beams_low obj_filament) fp12)

Goals: (fuse (obj_laser obj_filament) f21)

For good-constraint version, add:

(not-destroyed (obj_bulb) fg22)

For poor-constraint version, add instead:

(can-produce (obj_laser obj_beams_high) fp22)

RADIATION PROBLEM (target)

Start: (ray-source (obj_ray) d1)
 (tissue (obj_tissue) d2)
 (tumor (obj_tumor) d3)
 (surround (obj_tissue obj_tumor) d4)
 (outside (obj_ray obj_tissue) d5)
 (can-produce (obj_ray obj_rays_high) d6)
 (high-intensity (obj_rays_high obj_tumor) d8)
 (can-destroy (obj_rays_high obj_tumor) d8)
 (can-destroy (obj_rays_high obj_tissue) d9)
 (can-produce (obj_ray obj_rays_low) d10)
 (low-intensity (obj_rays_low) d11)
 (cannot-destroy (obj_rays_low obj_tumor) d12)
 (cannot-destroy (obj_rays_low obj_tissue) d13)

Goals: (destroy (obj_ray obj_tumor) d21)
 (not-destroyed (obj_tissue) d22)

SIMILARITY: (similar ray-source laser .08)

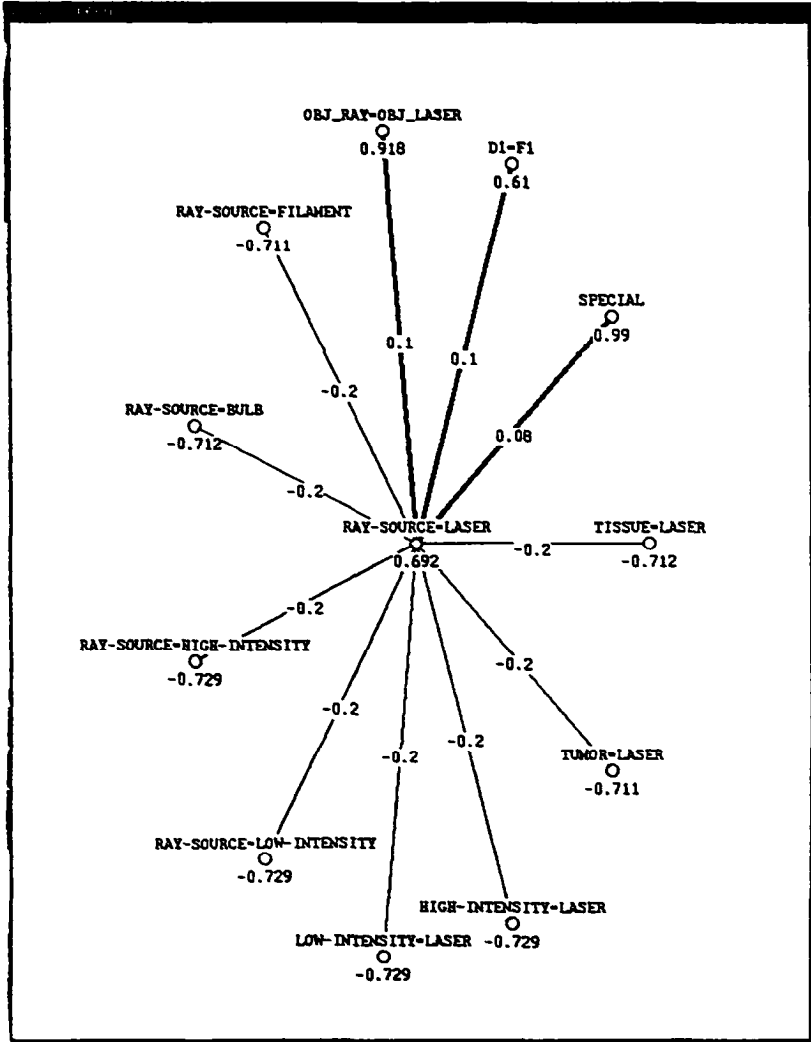


Figure 4. Output of a graphics program running with ACME, showing the subnetwork of excitatory and inhibitory connections to the unit *ray-source=laser*

activation less than .20) as target elements that have no good map in the source. (There are no such unmapped elements in this example.)

ACME also displays any additional possible mappings for each element, defined as units other than the best, which have activations greater than a threshold set at .20. In most cases each element has a single clear best mapping, but the occasional secondary mappings are also of interest. For example, the predicate *can-destroy* in the radiation problem has *can-destroy* in

the target problem as its best map (activation = .58), but has a secondary mapping to *can-fuse* (activation = .39). The reason for this dual mapping is apparent at the level of propositions (see Table 3). The high-strength ray can destroy the tissue just as the high-strength laser can destroy the bulb (i.e., $d9 = fg9$), but at the same time the high-strength ray can destroy the tumor just as the high-strength laser can fuse the filament ($d8 = fg8$). ACME finds the intuitively correct best mapping for both source propositions ($d9 = fg9$ has activation = .69, and $d8 = fg8$ has activation .68). However, at the level of predicate mappings *can-destroy = can-destroy* is preferred to *can-destroy = can-fuse* because of the greater semantic similarity of the concepts in the former mapping.

This analogy demonstrates that ACME can successfully map relations without requiring that they be coded as identical or even similar (e.g., *destroy = fuse*, activation = .71). One could argue, of course, that these relations overlap at the level of their underlying semantic structures, as both are causatives. However, this is a very abstract similarity; intuitively, the differences between the two predicates (e.g., one involves destruction and the other construction) are at least as salient as their similarities. The operation of ACME suggests how the overall structure of an analog can force dissimilar predicates into correspondence. The mapping thus leads to (rather than results from) discovery of underlying similarities. Once two predicates have been mapped, their similarities are highlighted while their differences are forced into the background.

The output provided in Table 4 does not reflect the history of activation of units as the network settles. Such information can be displayed, however, by using a graphics program that runs simultaneously with ACME and graphs the activation values of selected units. Figure 5 depicts activation histories of all the units concerning the mappings of ray-source, tissue, and tumor. In each graph, the x-axis is for time over 60 cycles of updating, and the y-axis shows activations ranging between 1 and -1, with the horizontal line indicating activation of 0. For example, the unit *ray-source = laser* rises steadily and then asymptotes, while the other units concerning the mapping of ray-source asymptote at negative activation values. This graph illustrates the characteristic stability of the asymptotic activation levels of individual units.

Table 5 presents results of running ACME on four versions of the convergence problem. The top row of the table presents the number of "cycles to success" for each version, defined as the number of cycles required for each correct mapping unit (for objects, predicates, and propositions used as arguments) to exceed the activation of its nearest competitor. It has generally been found that cycles to success provides a useful measure of relative difficulty of analogical mappings. Holyoak and Koh (1987) measured the percent of undergraduates who produced the convergence solution to the radiation problem after reading one of the four versions of the lightbulb problem, both before a hint to use the source was given, and in total, after a

TABLE 4
 Activation Values of Best Mappings of Radiation Problem to Lightbulb Problem
 (Laser/Fragile-Glass Version)*

Network has settled by cycle 31.
 Test: TEST0 Total times: 32
 Thu May 5 15:17:40 EDT 1988
 Laser analogy: basic, good constraint.
 Units not yet reached asymptote: 0
 Goodness of network: 4.84
 Calculating the best mappings after 32 cycles.
 Best mapping of RAY-SOURCE is LASER. 0.69
 Best mapping of TISSUE is BULB. 0.59
 Best mapping of TUMOR is FILAMENT. 0.59
 Best mapping of SURROUND is SURROUND. 0.77
 Best mapping of OUTSIDE is OUTSIDE. 0.77
 Best mapping of CAN-PRODUCE is CAN-PRODUCE. 0.88
 Best mapping of HIGH-INTENSITY is HIGH-INTENSITY. 0.71
 Best mapping of CAN-DESTROY is CAN-DESTROY. 0.58
 Mapping with CAN-FUSE is also possible: 0.39
 Best mapping of LOW-INTENSITY is LOW-INTENSITY. 0.71
 Best mapping of CANNOT-DESTROY is CANNOT-DESTROY. 0.58
 Mapping with CANNOT-FUSE is also possible: 0.39
 Best mapping of DESTROY is FUSE. 0.71
 Best mapping of NOT-DESTROYED is NOT-DESTROYED. 0.71
 Best mapping of OBJ_RAYS_LOW is OBJ_BEAMS_LOW. 0.88
 Best mapping of OBJ_RAYS_HIGH is OBJ_BEAMS_HIGH. 0.88
 Best mapping of OBJ_TUMOR is OBJ_FILAMENT. 0.90
 Best mapping of OBJ_TISSUE is OBJ_BULB. 0.91
 Best mapping of OBJ_RAY_ is OBJ_LASER. 0.92

* Values are activations of units after settling. This table and all other tables of mapping results are taken directly from ACME outputs, except that all numbers have been rounded to 2 decimal places.

hint was provided. For comparison with ACME's mapping results, these data are provided at the bottom of Table 5. Since ACME is modeling mapping only, not retrieval, the more relevant comparison is with the number of solutions after a hint was given. ACME is able to find the correct set of mappings in all four cases. Cycles to success does not differ as a function of surface similarity. All versions of the analogy have such extensive structural correspondences that the extra weight from the semantic unit to the mapping unit corresponding to the more similar predicate pair, *ray-source = laser*, has no measurable impact upon mapping difficulty. As the data from the Holyoak and Koh (1987) study indicate, people are also able to derive the mapping equally well in the laser and ultrasound conditions once a hint is provided.

Cycles to success does increase slightly for the two poor-constraint versions. Most seriously, proposition d22 in the radiation problem, which expresses the major constraint of not destroying the healthy tissue, has no

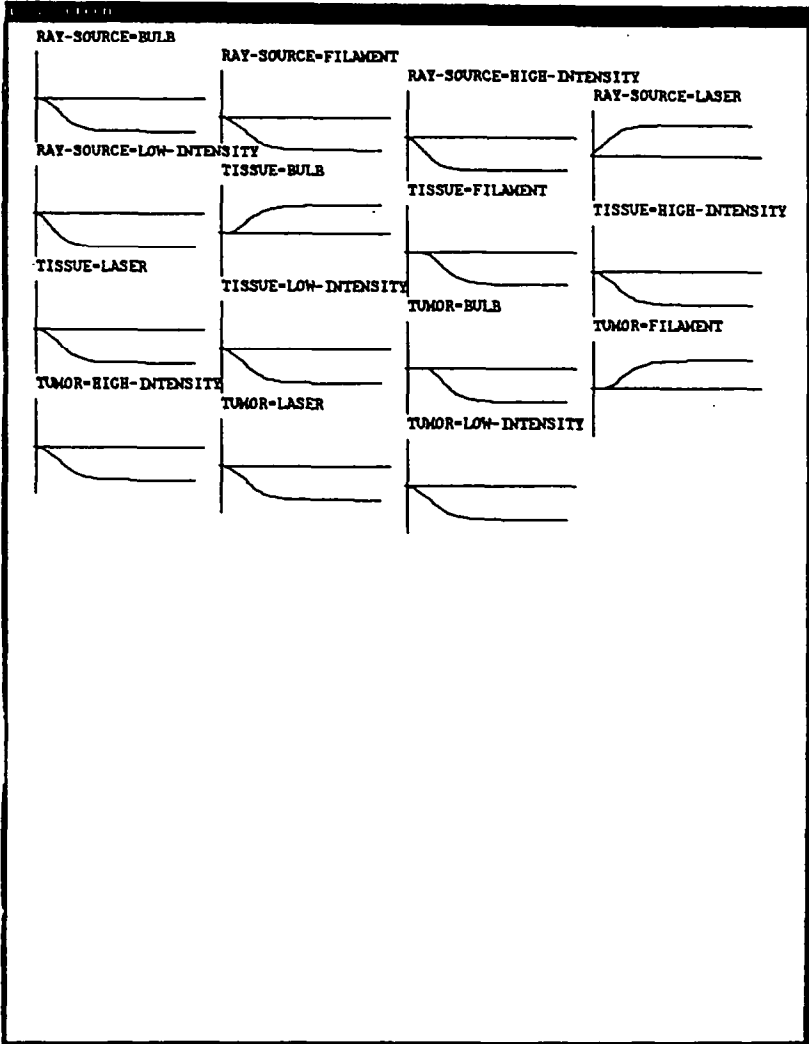


Figure 5. The activation history of several selected mapping units over 60 cycles of updating

map at all in the poor-constraint versions of the lightbulb problems. This breakdown of the analogy is likely to be a major part of the reason transfer was impaired for human subjects in the poor-constraint conditions of the Holyoak and Koh (1987) experiment.

ACME has also been applied to another convergence analogy, in which the target is the radiation problem and the source is the "fortress problem" used by Gick and Holyoak (1980). In the latter problem a general divides his army into small groups and has them converge simultaneously on a fortress

TABLE 5
Ease of Mapping Four Versions of Lightbulb/Radiation Problem Analogy

	Laser/ Fragile-glass	Ultrasound/ Fragile-glass	Laser/ Insufficient- Intensity	Ultrasound/ Insufficient- Intensity
Cycles to Success	3	3	5	5
Percent Convergence				
Solutions Prior to Hint*	69	38	33	13
Percent Convergence				
Solutions with Hint*	75	81	60	47

* Data from Holyoak and Koh (1987)

to capture it. ACME is also able to find the appropriate mapping for this additional convergence analogy.

Homomorphs of Missionaries-and-Cannibals. ACME's performance on the convergence analogies indicates that it can find useful mappings in the absence of a strict isomorphism. Humans can sometimes show problem-solving transfer between homomorphs that require more extensive one-to-many or many-to-one mappings. Once such example has been provided by Gholson, Eymard, Long, Morgan, and Leeming (1988). In the "missionaries-and-cannibals" problem, it is necessary to get a number of missionaries and cannibals across a river in a boat without leaving any missionaries outnumbered, and hence eaten, by cannibals. The homomorphic "farmer's dilemma" problem requires a farmer to take a fox, a goose, and some corn across a mountain in his wagon without having the fox eat the goose or the goose eat the corn. These problems are not isomorphic since multiple missionaries and multiple cannibals map equally well to a single corn and a single fox, respectively. Gholson et al. (1988) found that third- and fourth-grade children who had learned a solution to the farmer's dilemma problem were facilitated in subsequently solving a simple version of the missionaries-and-cannibals problem (with three missionaries and two cannibals). However, no transfer was obtained in the reverse direction, from missionaries-and-cannibals to farmer's dilemma. A similar transfer asymmetry was obtained with adult subjects by Reed, Ernst, and Banerji (1974), using other missionaries-and-cannibals homomorphs.

Table 6 shows predicate-calculus representations of the two problems in the form in which they were given to ACME. These representations are simplified in that the constraints on which objects can be left together are not explicitly included; however, the representations contain the basic information required to indicate that each missionary and each cannibal have the same properties, so that the optimal mapping should be a homomorphism. Given that ACME forms inhibitory links in order to encourage one-to-one

TABLE 6
 Predicate-Calculus Representations of Missionaries-and-Cannibals
 and Farmer's Dilemma Problems

 MISSIONARIES-AND-CANNIBALS

Start: (missionary (obj_missionary1) m1)
 (missionary (obj_missionary2) m2)
 (missionary (obj_missionary3) m3)
 (cannibal (obj_cannibal1) m4)
 (cannibal (obj_cannibal2) m5)
 (boat (obj_boat) m6)
 (carries (obj_boat vbl_rower vbl_passenger) m7)
 (eat (obj_cannibal1 obj_missionary1) m8)
 (eat (obj_cannibal1 obj_missionary2) m9)
 (eat (obj_cannibal1 obj_missionary3) m10)
 (eat (obj_cannibal2 obj_missionary1) m11)
 (eat (obj_cannibal2 obj_missionary2) m12)
 (eat (obj_cannibal2 obj_missionary3) m13)

Goals: (cross_river (obj_missionary1) m14)
 (cross_river (obj_missionary2) m15)
 (cross_river (obj_missionary3) m16)
 (cross_river (obj_cannibal1) m17)
 (cross_river (obj_cannibal2) m18)
 (not_eaten (obj_missionary1) m19)
 (not_eaten (obj_missionary2) m20)
 (not_eaten (obj_missionary3) m21)

FARMER'S DILEMMA:

Start: (farmer (obj_farmer) f1)
 (fox (obj_fox) f2)
 (goose (obj_goose) f3)
 (corn (obj_corn) f4)
 (wagon (obj_wagon) f5)
 (carries (obj_wagon obj_farmer vbl_thing) f6)
 (eat (obj_fox obj_goose) f7)
 (eat (obj_goose obj_corn) f8)

Goals: (cross_mountain (obj_farmer) f9)
 (cross_mountain (obj_fox) f10)
 (cross_mountain (obj_goose) f11)
 (cross_mountain (obj_corn) f12)
 (not_eaten (obj_goose) f13)
 (not_eaten (obj_corn) f14)

mappings, it is not obvious that the program could find many-to-one or one-to-many mappings. However, as the outputs presented in Table 7 demonstrate, ACME in fact sorts out the relations among the elements of the two problems very well, both when mapping from missionaries-and-cannibals to farmer's dilemma (many-to-one mappings, Table 7A) and from farmer's dilemma to missionaries-and-cannibals (one-to-many mappings, Table 7B).

TABLE 7

Results of Mapping Missionaries-and-Cannibals to Farmer's Dilemma and Vice Versa

A. Missionaries-and-Cannibals to Farmer's Dilemma:

Network has settled by cycle 41.
 Test: TEST22 Total times: 42
 Mon May 2 17:09:08 EDT 1988
 Mapping cannibals & missionaries (3) to farmer's dilemma.
 Units not yet reached asymptote: 0
 Goodness of network: 2.26
 Calculating the best mappings after 42 cycles.
 Best mapping of OBJ_MISSIONARY1 is OBJ_CORN. 0.37
 Mapping with OBJ_GOOSE is also possible: 0.26
 Best mapping of OBJ_MISSIONARY2 is OBJ_CORN. 0.37
 Mapping with OBJ_GOOSE is also possible: 0.26
 Best mapping of OBJ_MISSIONARY3 is OBJ_CORN. 0.37
 Mapping with OBJ_GOOSE is also possible: 0.26
 Best mapping of OBJ_CANNIBAL1 is OBJ_FOX. 0.52
 Mapping with OBJ_GOOSE is also possible: 0.21
 Best mapping of OBJ_CANNIBAL2 is OBJ_FOX. 0.52
 Mapping with OBJ_GOOSE is also possible: 0.21
 Best mapping of OBJ_BOAT is OBJ_WAGON. 0.80
 Best mapping of VBL_PASSENGER is VBL_THING. 0.75
 Best mapping of VBL_ROWER is OBJ_FARMER. 0.75
 Best mapping of NOT_EATEN is NOT_EATEN. 0.78
 Best mapping of CROSS_RIVER is CROSS_MOUNTAIN. 0.82
 Best mapping of EAT is EAT. 0.90
 Best mapping of CARRIES is CARRIES. 0.79
 Best mapping of BOAT is WAGON. 0.57
 Best mapping of CANNIBAL is FOX. 0.60
 Best mapping of MISSIONARY is CORN. 0.61

B. Farmer's Dilemma to Missionaries-and-Cannibals:

Network has settled by cycle 41.
 Test: TEST23 Total times: 42
 Mon May 2 17:11:11 EDT 1988
 Mapping farmer's dilemma to cannibals & missionaries (3).
 Units not yet reached asymptote: 0
 Goodness of network: 2.26
 Calculating the best mappings after 42 cycles.
 Best mapping of FARMER is BOAT. -0.52
 Best mapping of FOX is CANNIBAL. 0.60
 Best mapping of GOOSE is MISSIONARY. -0.14
 Best mapping of CORN is MISSIONARY. 0.60
 Best mapping of WAGON is BOAT. 0.57
 Best mapping of CARRIES is CARRIES. 0.79
 Best mapping of EAT is EAT. 0.90
 Best mapping of CROSS_MOUNTAIN is CROSS_RIVER. 0.82
 Best mapping of NOT_EATEN is NOT_EATEN. 0.78
 Best mapping of VBL_THING is VBL_PASSENGER. 0.75
 Best mapping of OBJ_WAGON is OBJ_BOAT. 0.80
 Best mapping of OBJ_CORN is OBJ_MISSIONARY3. 0.37
 Mapping with OBJ_MISSIONARY2 is also possible: 0.37
 Mapping with OBJ_MISSIONARY1 is also possible: 0.37

TABLE 7 (Continued)

Best mapping of OBJ_GOOSE is OBJ_MISSIONARY3. 0.26
Mapping with OBJ_CANNIBAL2 is also possible: 0.20
Mapping with OBJ_CANNIBAL1 is also possible: 0.20
Mapping with OBJ_MISSIONARY2 is also possible: 0.26
Mapping with OBJ_MISSIONARY1 is also possible: 0.26
Best mapping of OBJ_FOX is OBJ_CANNIBAL2. 0.52
Mapping with OBJ_CANNIBAL1 is also possible: 0.52
Best mapping of OBJ_FARMER is VBL_ROWEN. 0.75

Thus in the former direction each missionary maps equally well to the corn, and each cannibal maps equally well to the fox. The goose is a weaker secondary map for both the missionaries and the cannibals; at the level of predicates, however, the only successful mapping units are *missionary = corn* and *cannibal = fox*. ACME is able to find many-to-one mappings at the level of objects because the positive evidence favoring each of several object-object mappings (e.g., each cannibal to the fox) provides total excitation that exceeds the inhibition generated by inhibitory links between the competing units. In ACME the structural constraint of one-to-one mapping, like that of structural consistency, is treated as a pressure rather than a requirement.

The mappings found by ACME may also help account for the asymmetry in transfer between the two problems observed by Gholson et al. (1988). Although subjects may map the problems in either direction, they of course only know the solution to one of them at the time of transfer. For simplicity, let us assume subjects were mapping from target to source, so that the mapping from missionaries-and-cannibals to farmer's dilemma (Table 7A) represents the basis for transferring knowledge about the farmer's dilemma to the various characters of missionaries-and-cannibals. It is clear from examining the solutions required for the two problems that there are no useful correspondences at the level of moves. However, subjects may have been able to use their experience with the source to help construct appropriate tests for legal moves. The basis for the analogy is that in each problem, to avoid illegal moves it is necessary to check, during the process of move selection, that objects are neither threats to, nor threatened by, other objects. If the source problem is the farmer's dilemma, then the mapping in Table 7A gives relatively clear guidance about the test required for each object in the missionaries-and-cannibals problem: each cannibal should be viewed as a potential threat (like the fox), and each missionary should be viewed as potentially threatened (like the corn). This is the direction in which Gholson et al.'s subjects were able to transfer successfully.

In contrast, subjects who receive the missionaries-and-cannibals problem as the source will have less adequate guidance in dealing with the farmer's dilemma characters. As Table 7B indicates, the fox maps to each of the cannibals, and the corn maps to each of the missionaries; the predicate map-

pings *fox = cannibal* and *corn = missionary* emerge unambiguously. These mappings will correctly lead subjects to decide to view the fox as a threat and the corn as threatened. Note, however, that the mapping provides no clear guidance about how to treat the goose. The goose maps weakly to each of the missionaries and cannibals; the predicate *goose* has no good map at all. Intuitively, the goose in the farmer's dilemma problem is sometimes threatened and sometimes a threat; the missionaries-and-cannibals problem provides no clear information about how to deal with such ambiguity. Accordingly, transfer in this direction will be greatly impaired, as Gholson et al. observed.

Pragmatics in Analogical Arguments

In analogical arguments, one implies that what is true of one situation is also true of another situation. Such arguments have a long history in philosophy, being used, for example, to argue for the existence of other minds and the existence of God. (I know that I have a mind because of my experiences, and I infer that you also have a mind because you are similar to me in many other respects. Less metaphysically, I may infer that since my favorite candidate for the United States presidency is like former president John F. Kennedy, my candidate too will win the election.)

Analogical arguments may rely upon an unbiased assessment of similarities, but often people are motivated to reach particular conclusions. It is conjectured here that such motivations may bias people's judgments of similarity, just as Kunda (1987) has shown that generalization can be biased by personal motivation. Motivated analogical arguments would provide striking examples of the pragmatic nature of analogy, since judgments about what corresponds to what will be biased by the conclusion the analogy is intended to produce.

To illustrate the possible role of biases in analogical argumentation, ACME was used to simulate a very simple example of a motivated use of analogy, involving the issue of assessing the nature of the Contras' attempt to overthrow the government of Nicaragua in 1987. Table 8 shows the input representations provided to the program. The "Contras" structure contains only the minimal information that the Contras aim to overthrow the government of Nicaragua, leaving open the question of whether the U.S. should support them, and whether they should be viewed as terrorists or freedom fighters. The unknown but desired information about the Contras is signalled by variables representing cross-structure queries, such as *support?*. The "Others" analog contains substructures corresponding to Hungarians wanting to overthrow their communist government, who are categorized as freedom fighters, and to the PLO wanting to overthrow the Israeli government, who are categorized as terrorists. The Contras structure has the same degree of structural and semantic similarity to each of the

TABLE 8
 Predicate-Calculus Representations for Contras Example

CONTRAS	(country (Nicaragua) c1)
	(govern (Sandinistas Nicaragua) c2)
	(aim-to-overthrow (Contras Sandinistas) c3)
	(terrorists? (Contras Contra-terror?) c4)
	(freedom-fighters? (Contras Contra-freedom?) c5)
	(should-support? (US Contras support?) c6)
OTHERS	
	(country (Hungary) h1)
	(govern (communists Hungary) h2)
	(aim-to-overthrow (Hungarians communists) h3)
	(freedom-fighters? (Hungarians freedom-fighters-yes) h5)
	(terrorists? (Hungarians terrorists-no) h4)
	(should-support? (US Hungarians support-yes) h6)
	(country (Israel) i1)
	(govern (Israelis Israel) i2)
	(aim-to-overthrow (PLO Israelis) i3)
	(terrorists? (PLO terrorists-yes) i4)
	(freedom-fighters? (PLO freedom-fighters-no) i5)
	(should-support? (US PLO support-no) i6)
PRESUMED:	support?=support-yes

Hungarian and Israeli substructures, and maps to them equally well if no additional information is provided.

Suppose, however, that the analogist is motivated to support the Contras. This person will want *support?* to correspond to *support-yes*. To represent this type of bias in the mapping process, ACME's input in Table 8 includes the information that this is a PRESUMED mapping. Results of running ACME are shown in Table 9. The result of ACME's bias toward *support?* = *support-yes* is not only that the network indeed settles upon the mapping *support?* = *support-yes*, but that the Contras map to Hungarians rather than to PLO, so that *Contra-terror?* maps to *terrorists-no* and *Contra-freedom?* maps to *freedom-fighters-yes*. A very different result is produced if ACME is told that it is desirable to reach the conclusion that the Contras should not be supported: the Contras then map to the PLO and take on their characteristics. Thus a single entering bias can potentially alter the entire mapping that will be found.

Also modeled was a similar case of motivated analogy using three possible analogs. Suppose you are inferring some of the policies of your favorite political candidate. Then you will be motivated to infer that the candidate has properties that will make election possible. If you believe that a moderate will have the best chance of being elected, then you will be prone to view

TABLE 9
Output for Contras Example

Network has settled by cycle 103.
 Test: TEST7 Total times: 104
 Mon May 2 14:46:02 EDT 1988
 Support the contras?
 Units not yet reached asymptote: 0
 Goodness of network: 2.53
 Calculating the best mappings after 104 cycles.
 Best mapping of NICARAGUA is HUNGARY. 0.75
 Best mapping of SANDINISTAS is COMMUNISTS. 0.73
 Best mapping of CONTRAS is HUNGARIANS. 0.75
 Mapping with PLO is also possible: 0.42
 Best mapping of CONTRA-TERROR? is TERRORISTS-NO. 0.68
 Best mapping of CONTRA-FREEDOM? is FREEDOM-FIGHTERS-YES. 0.68
 Best mapping of SUPPORT? is SUPPORT-YES. 0.77
 Best mapping of US is US. 0.81
 Best mapping of SHOULD-SUPPORT? is SHOULD-SUPPORT?. 0.84
 Best mapping of FREEDOM-FIGHTERS? is FREEDOM-FIGHTERS? 0.77
 Best mapping of TERRORISTS? is TERRORISTS?. 0.77
 Best mapping of AIM-TO-OVERTHROW is AIM-TO-OVERTHROW. 0.78
 Best mapping of GOVERN is GOVERN. 0.76
 Best mapping of COUNTRY is COUNTRY. 0.69

your candidate as most similar to another moderate; accordingly, you will infer that your candidate is more analogous to a known moderate politician than to a conservative or liberal. ACME models this behavior when it maps a target politician to an "Others" analog consisting of a liberal, a moderate, and a conservative. If the target is about equally similar to the three alternatives in structure and in semantic properties, then treating the mapping of "politics-value" to "moderate" as PRESUMED is sufficient to cause the target to map to the moderate candidate and take on that politician's characteristics.

Note, however, that such pragmatic biases can only *select* among mappings that are reasonably close in terms of the structural and semantic constraints. Pragmatic biases will not suffice, for example, to view an arch conservative as a moderate simply because moderates are believed to be electable. In a second version of the "politics" simulation, the target politician is represented as having the same properties as the known conservative, and then the program was run with a pragmatic weight indicating that the target was a moderate, as in the version described above. In early cycles the target mapped most strongly to the moderate politician, but eventually the structural and semantic pressures dominated. At asymptote, ACME mapped the target politician to the conservative despite its pragmatic bias. Thus although pragmatic considerations can guide the mapping process, they cannot overwhelm other information.

Pragmatics in Explanatory Analogies

ACME has been applied to the explanatory analogies discussed by Gentner (1983, 1989) and Falkenhainer et al. (1986): the analogy between the flow of water caused by differential pressure and the flow of heat caused by differential temperature; and the analogy between the motion of planets around the sun and of electrons around an atomic nucleus. These examples allow a close comparison of the ACME and SME programs, since Falkenhainer et al. (1986) describe the representations used as input to SME in sufficient detail so that essentially the same information could be provided about each analog to ACME.

Table 10 presents predicate-calculus representations of the water-flow/heat-flow analogy that were used as inputs to ACME. Two versions were used. The basic version is based directly upon an example used as a test of SME by Falkenhainer et al. (1986). This version is of interest because it represents an analogy in which considerably more information is known about the source (water flow) than about the target (heat flow), as is often the case when a novel situation is explained in terms of a more familiar one. Despite surface representational differences, our predicate-calculus representations encode essentially the same information as that which Falkenhainer et al. provided to the SME program. The major exception is that ACME received an explicit representation of the pragmatic purpose of using the analogy. The internal query *?proposition?* in proposition h11 of the heat analog represents the unknown cause of heat flow.

Table 11A presents the activation levels of selected mapping hypotheses after the network settles in 80 cycles. Falkenhainer et al. constructed the example to include three impediments to a successful map from water flow to heat flow. ACME, like SME, is able to handle all three difficulties. First, the attribute *clear* in the water-flow analog has no acceptable map in the heat-flow analog, as is evident from the fact that no mapping unit for *clear* has an activation above 0. Second, the information that both water and coffee are liquids and have a flat top tends to encourage water to map to coffee rather than to heat. Despite this misleading similarity information, the structural information encoded in the network enables ACME, by cycle 3, to provide higher activation to the unit representing the hypothesis that water maps to heat. As the values in Table 11 indicate, the mapping from water to heat emerges as a clear victor over the alternative possibility of mapping water to coffee.

The third and most serious impediment to a successful map is the irrelevant information concerning the diameters of the beaker and vial, encouraging the map of diameter to pressure in competition with the correct map of temperature to pressure. SME selects the correct map on the basis of Gentner's principle of systematicity, interpreted as a preference for mappings that yield the greatest number of possible inferences. In contrast, the preferability of the temperature-pressure map is viewed here as largely a

TABLE 10
 Predicate-Calculus Representation of Water-Flow and Heat-Flow Analogs,
 and Extended Versions

WATER-FLOW (source)

(liquid (obj_water) w1)
 (flat-top (obj_water) w2)
 (clear (obj_beaker) w3)
 (diameter (obj_beaker obj_val1) w4)
 (diameter (obj_via1 obj_val2) w5)
 (greater (obj_val1 obj_val2) w6)
 (pressure (obj_beaker obj_val3) w7)
 (pressure (obj_via1 obj_val4) w8)
 (greater (obj_val3 obj_val4) w9)
 ; flow: from x to y of w via z
 (flow (obj_beaker obj_via1 obj_water obj_pipe) w10)
 ; pressure difference causes flow:
 (cause (w9 w10) w11)

Extended version adds:

(volume (obj_beaker obj_val5) w11)
 (volume (obj_via1 obj_val6) w12)
 (greater (obj_val5 obj_val6) w13)
 ; diameter difference causes volume difference
 (cause (w6 w13) w14)

HEAT-FLOW (target)

(liquid (obj_coffee) h1)
 (flat-top (obj_coffee) h2)
 (temperature (obj_coffee obj_val13) h7)
 (temperature (obj_ice_cube obj_val14) h8)
 (greater (obj_val13 obj_val14) h9)
 (flow (obj_coffee obj_ice_cube obj_heat obj_bar) h10)
 ; what causes heat flow?
 (cause (?proposition? h10) h11)

Extended version adds:

(volume (obj_coffee obj_val15) h11)
 (volume (obj_ice_cube obj_val16) h12)
 (greater (obj_val15 obj_val16) h13)

pragmatic matter of the intended use of the analogy. If water flow is being used to explain heat flow, then aspects of water systems that affect its flow (pressure differences rather than diameter differences) should be favored in mapping.

In the representation of the heat-flow analogy in Table 10, the information-seeking purpose of the analogy is captured by the proposition (*cause (?proposition? h10) h11*) in the heat-flow representation, where “*?proposition?*” represents the unknown cause of heat flow. This internal-query variable signals that the purpose of the mapping is to identify an actual

TABLE 11
Output for Heat-Flow Example

A. Basic version:

Network has settled by cycle 80.
Test: TEST2 Total times: 81
Thu Dec 8 09:41:16 EST 1988
Analogy between water flow and heat flow
Units not yet reached asymptote: 0
Goodness of network: 2.84
Calculating the best mappings after 81 cycles.
Best mapping of LIQUID is LIQUID. 0.53
Best mapping of FLAT-TOP is FLAT-TOP. 0.53
Best mapping of CLEAR is LIQUID. -0.02
 tied with FLAT-TOP.
Best mapping of DIAMETER is TEMPERATURE. 0.04
Best mapping of GREATER is GREATER. 0.76
Best mapping of PRESSURE is TEMPERATURE. 0.78
Best mapping of FLOW is FLOW. 0.82
Best mapping of CAUSE is CAUSE. 0.76
Best mapping of W9 is H9. 0.78
 Mapping with ?PROPOSITION? is also possible: 0.78
Best mapping of W10 is H10. 0.85
Best mapping of OBJ_PIPE is OBJ_BAR. 0.79
Best mapping of OBJ_VAL4 is OBJ_VAL14. 0.79
Best mapping of OBJ_VAL3 is OBJ_VAL13. 0.79
Best mapping of OBJ_VIAL is OBJ_ICE_CUBE. 0.85
Best mapping of OBJ_VAL2 is OBJ_VAL14. 0.02
Best mapping of OBJ_VAL1 is OBJ_VAL13. 0.02
Best mapping of OBJ_BEAKER is OBJ_COFFEE. 0.86
Best mapping of OBJ_WATER is OBJ_HEAT. 0.79

B. Extended version:

Network has settled by cycle 92.
Test: TEST6 Total times: 93
Thu Dec 8 10:02:42 EDT 1988
Extended map from water to heat.
Units not yet reached asymptote: 0
Goodness of network: 4.08
Calculating the best mappings after 93 cycles.
Best mapping of LIQUID is LIQUID. 0.52
Best mapping of FLAT-TOP is FLAT-TOP. 0.52
Best mapping of CLEAR is LIQUID. -0.004
 tied with FLAT-TOP.
Best mapping of DIAMETER is TEMPERATURE. 0.10
Best mapping of GREATER is GREATER. 0.84
Best mapping of PRESSURE is TEMPERATURE. 0.74
Best mapping of FLOW is FLOW. 0.83
Best mapping of CAUSE is CAUSE. 0.76
Best mapping of VOLUME is VOLUME. 0.84
Best mapping of W6 is H9. -0.14
Best mapping of W14 is H14. 0.70

continued

TABLE 11 (Continued)

Best mapping of OBJ_VAL6 is OBJ_VAL16. 0.81
Best mapping of OBJ_VAL5 is OBJ_VAL15. 0.81
Best mapping of W9 is H9. 0.78
Mapping with ?PROPOSITION? is also possible: 0.78
Best mapping of W10 is H10. 0.85
Best mapping of OBJ_PIPE is OBJ_BAR. 0.80
Best mapping of OBJ_VAL4 is OBJ_VAL14. 0.75
Best mapping of OBJ_VAL3 is OBJ_VAL13. 0.75
Best mapping of OBJ_VIAL is OBJ_ICE_CUBE. 0.89
Best mapping of OBJ_VAL2 is OBJ_VAL14. 0.09
Best mapping of OBJ_VAL1 is OBJ_VAL13. 0.09
Best mapping of OBJ_BEAKER is OBJ_COFFEE. 0.89
Best mapping of OBJ_WATER is OBJ_HEAT. 0.80

proposition in the heat situation that can fill the empty argument slot. Since *w9* concerning the greater temperature has this desired feature by virtue of its appearance in the proposition (*cause (w9 w10) w11*), whereas *w6* concerning the greater diameter does not, units for mapping the former are preferred to units for mapping the latter. Because the mapping of *w9* to *h9* is therefore preferred to the mapping of *w6* to *h9*, the mapping of pressure to temperature is preferred to the mapping of diameter to temperature. (Note that because *?proposition?* is treated as a variable that supports rather than competes with specific values such as *h9* (see Table 1, Note 3), ACME reports both *?proposition?* and *h9* as the best maps for *w9*.)

Although SME and ACME both find the appropriate mappings in this example, some minor extensions serve to differentiate the performance of the two programs. For SME it is crucial that the identical predicate "flow" be used in the water-flow and heat-flow situations, or else this correspondence would not be found. For ACME this identity is not essential. The basic version of the analogy was also run with the predicates "water-flow" and "heat-flow," respectively, substituted for "flow" in the two analogs. The two distinct predicates were given a minimal similarity weight. Given that these predicates were the only four-place predicates in the analogs, and hence forced to map by the type restriction, this variation is not a serious challenge for ACME. The resulting asymptotic solution, which is reached after 78 cycles, is virtually identical to that shown in Table 11A; the mapping unit *water-flow* = *heat-flow* asymptotes at an activation of .80.

As noted above, SME prefers the mapping of pressure to temperature over the mapping of diameter to temperature for a different reason than ACME does. Whereas ACME bases its choice on the fact that only the former mapping is directly relevant to the purpose of the analogy, SME bases its choice upon the fact that only the former allows a causal inference to be constructed, because only temperature change is the cause of something in the water-flow situation. These two criteria can be separated by a

small extension of the two analogs, as indicated in Table 10. In the extended version of the water-flow situation, it is assumed the analogist has noticed that the greater diameter of the beaker relative to the vial is the cause of the former having greater volume. It is further supposed that the analogist has observed in the heat-flow situation that the volume of the coffee exceeds that of the ice cube. If the internal query is not included in the latter representation, then the systematicity principle provides no basis for choosing between the two competing mappings involving temperature in this extended version. For, if pressure maps to temperature, than it is possible to infer that temperature differences cause flow; whereas if diameter maps to temperature, it is possible to infer that temperature change is the cause of the observed differences between the volumes of the coffee and the ice cube. The latter inference is of course erroneous, but worse, it is irrelevant to the analogist's presumed goal of understanding heat flow. In contrast, as the output in Table 11B indicates, ACME's internal-query mechanism continues to produce a clear preference for mapping pressure to temperature in the extended version of the analogy.

ACME is also able to produce the appropriate mapping for the solar-system/atom analogy that Falkenhainer et al. (1986) used to test SME. Since this analogy does not provide any additional complexities (the network settles with the correct solution after 40 cycles), the results will not be described further.

ACME has also been applied to analogies used by chemistry teachers to explain difficult concepts to students, but these results are reported elsewhere (Thagard et al., 1989).

Competing Constraints in Mapping Story Analogs

Additional evidence concerning ACME's ability to account for empirical evidence relating to the effect of systematicity on mapping is provided by a study performed by Gentner and Toupin (1986). This experiment investigated the effects both of systematicity and *transparency*: the degree to which similar objects serve similar functions in the analogy. Gentner and Toupin presented two groups of children, aged 4–6 years and 8–10 years, with a series of simple stories. After the child had acted out one version of a story with props, the experimenter asked him or her to act out the same story with different characters.

Table 12 presents a simplified version of one of these stories that served as the basis for a simulation by ACME, and Table 13 presents the actual predicate-calculus representation provided to the program. As indicated in Table 14, each source story was used across children in either a "systematic" or a "nonsystematic" form. The systematic version differed from the nonsystematic version in that it added additional information relevant to the causes of events in the story (e.g., the cat's jealousy caused its anger). Transparency was varied by manipulating the similarity of the animals in the

TABLE 12
 Precis of a "Jealous Animal" Story as Used in ACME Simulation,
 In Systematic and Nonsystematic Versions

The cat was jealous.

(Nonsystematic version: The cat was strong.)

The cat was friends with a walrus.

The walrus played with a seagull.

The cat was angry.

(Systematic version: Because the cat was jealous and the walrus played with the seagull, the cat was angry.)

The cat was reckless.

(Systematic version: Because the cat was angry, it was reckless.)

The cat got in danger.

(Systematic version: Because the cat was reckless, it got in danger.)

The seagull saved the cat.

(Systematic version: Because the seagull saved the cat, the cat was friends with the seagull.)

TABLE 13
 Predicate-Calculus Representation of a Jealous Animal Story:
 Similar Objects/Similar Roles (Systematic and Unsystematic Versions)

(cat (obj_cat) b1)
 (walrus (obj_walrus) b2)
 (seagull (obj_seagull) b3)
 (friends (obj_cat obj_walrus) b4)
 (played (obj_walrus obj_seagull) b5)
 (angry (obj_cat) b6)
 (reckless (obj_cat) b7)
 (endangered (obj_cat) b8)
 (save (obj_seagull obj_cat) b9)

Systematic version adds:

(jealous (obj_cat) b10)
 (befriend (obj_cat obj_seagull) b11)
 (conjoin-event (b10 b5) b12)*
 (cause (b12 b6) b13)
 (cause (b6 b7) b14)
 (cause (b7 b8) b15)
 (cause (b9 b11) b16)

Unsystematic version adds instead:

(strong (obj_cat) b10)

* The interpretation of a conjoin-event is that two events are conjoined to make a third event. This device is needed so that cause can remain a two-place relation despite conjunctive causes.

TABLE 14
Mapping Conditions for "Jealous Animal" Stories

Test	Similar Objects/ Similar Roles	Dissimilar Objects	Similar Objects/ Dissimilar Roles
dog	cat	camel	seagull
seal	walrus	lion	cat
penguin	seagull	giraffe	walrus

various roles. In the example used in the simulation, the target analog involved a dog, seal, and penguin. In the S/S (similar objects/similar roles) condition, the source analog involved similar characters playing similar roles (cat, walrus, and seagull). In the D (dissimilar objects) condition, all the characters were quite different from those in the target (camel, lion, and giraffe). In the cross-mapped S/D (similar objects/dissimilar roles) condition, similar characters were used, but these played different roles than did the corresponding characters in the target (seagull, cat, and walrus).

Gentner and Toupin found that both systematicity and transparency affected the accuracy with which children enacted the target stories. The two effects interacted, in that performance was uniformly good, regardless of systematicity, in the S/S condition. As the transparency of the mapping decreased from the S/S to the D and the S/D conditions, performance declined, and the advantage of the systematic over the unsystematic version increased. The positive impact of systematicity was more pronounced for the older group of children.

In order to stimulate these results, predicate-calculus representations of the stories were used as inputs to ACME (see Table 13). If the similarity of the characters in the source and target was high, the similarity weight for the corresponding predicate-mapping unit was set at .06; if the similarity was low, the weight was set at .01. Units for pairings of identical predicates were given similarity weights of .1, the value of *smax* in all the reported simulations. Table 15 presents the cycles to success (the first cycle at which the appropriate winning units emerged for all objects, predicates, and propositional arguments) in each of the six conditions. Values of cycles to success increased very slightly from the S/S to D conditions, and more dramatically to the cross-mapped S/D condition. Only in the latter condition did systematicity have an effect on cycles to success, as the network failed to settle on the correct mapping in the unsystematic S/D condition. In this most difficult condition, the network was unable to overcome the misleading impact of semantic similarity. Further exploration revealed that the correct solution can be obtained in the S/D condition if either the value of *i* is reduced from $-.2$ to $-.1$ or the value of *min* is raised from -1 to $-.3$; however, cycles to success remains higher in the unsystematic than the systematic S/D con-

TABLE 15
Results of ACME Runs for Six Versions of "Jealous Animal" Stories

Version	Cycles to Success
Systematic:	
S/S	3
D	4
S/D	38
Nonsystematic:	
S/S	3
D	4
S/D	no success

dition. Although ACME seems to show relatively fewer differences between the S/S and D conditions than did Gentner and Toupin's subjects, the program does capture the substantially greater difficulty of the cross-mapped S/D condition, and the role of systematicity in overcoming misleading semantic similarity.

As noted above, Gentner and Toupin found that the younger children benefited less from high systematicity than did the older children. The authors suggested that focus on systematicity increases with age. In terms of the present theory, it is possible that with age children learn to place greater weight on isomorphism, and less on the similarity constraint. It is also possible, however, that the younger children in the Gentner and Toupin (1986) study simply failed to grasp some of the causal structure provided in the systematic stories, and hence encoded the source stories imperfectly. Thus the lesser benefit they derived from the systematic versions need not imply insensitivity to the isomorphism constraint.

Finding Isomorphisms Without Semantic or Pragmatic Information

A Formal Isomorphism. As pointed out in the comparison of ACME with SME, ACME is able to use structural information to map predicates that are not semantically identical or even similar. In fact, if two analogs are isomorphic, it should be possible to derive an appropriate mapping even in the complete absence of semantic or pragmatic information. Table 16 presents a formal analogy between addition of numbers and union of sets that was used to demonstrate this point. Both addition and union have the abstract mathematical properties of commutativity, associativity, and the existence of an identity element (0 for numbers, the empty set Φ for sets). ACME was given predicate-calculus representations of these two analogs, with no identical elements (note that number equality and set equality are given distinct symbols), and with all semantic weights set equal to the minimal value. This analogy is quite complex, as many propositions have the

TABLE 16
Formal Isomorphism Between Addition of Numbers and Union of Sets

Property	Addition	Union
Commutativity:	$N1 + N2 = N2 + N1$	$S1 \cup S2 = S2 \cup S1$
Associativity:	$N3 + (N4 + N5) = (N3 + N4) + N5$	$S3 \cup [S4 \cup S5] = [S3 \cup S4] \cup S5$
Identity:	$N6 + 0 = N6$	$S6 \cup \emptyset = S6$
Predicate-Calculus Representations:		
NUMBERS:	<pre>(sum (num1 num2 num10) n1) (sum (num2 num1 num11) n2) (num_eq (num10 num11) n3) (sum (num5 num6 num12) n3) (sum (num4 num12 num13) n5) (sum (num4 num5 num14) n6) (sum (num14 num6 num15) n7) (num_eq (num13 num15) n8) (sum (num20 zero num20) n9)</pre>	
SETS:	<pre>(union (set1 set2 set10) s1) (union (set2 set1 set11) s2) (set_eq (set10 set11) s3) (union (set5 set6 set12) s4) (union (set4 set12 set13) s5) (union (set4 set5 set14) s6) (union (set14 set6 set15) s7) (set_eq (set13 set15) s8) (union (set20 empty-set set20) s9)</pre>	

same predicate (*sum* or *union*), and many symbols representing intermediate results must be sorted out. Note that the representations given to the program did not explicitly group the components of each analog into three distinct equations. In the absence of any semantic or pragmatic information, only weights based upon isomorphism, coupled with the type restriction, provided information about the optimal mapping.

As the output in Table 17 indicates, ACME settles to a complete solution to this formal mapping problem after 59 cycles. The model is thus able to derive a unique mapping in the absence of any overlap between the elements of the source and target. ACME's ability to deal with such examples is crucially dependent upon its parallel constraint-satisfaction algorithm.

Isomorphism without Explicit Relational Predicates. The best mapping for the addition/union analogy, as for all the examples considered so far, involves a rich set of relational correspondences. It is important to understand, however, that the structural constraint of isomorphism is not strictly

TABLE 17
Output after Running Addition/Union Analogy

Network has settled by cycle 59.
 Test: TEST0 Total times: 60
 Mon May 2 10:40:03 EDT 1988
 Analogy between numbers and sets.
 Units not yet reached asymptote: 0
 Goodness of network: 3.31
 Calculating the best mappings after 60 cycles.
 Best mapping of NUM10 is SET10. 0.79
 Best mapping of NUM2 is SET2. 0.82
 Best mapping of NUM1 is SET1. 0.82
 Best mapping of NUM11 is SET11. 0.79
 Best mapping of NUM12 is SET12. 0.82
 Best mapping of NUM6 is SET6. 0.82
 Best mapping of NUM5 is SET5. 0.82
 Best mapping of NUM13 is SET13. 0.79
 Best mapping of NUM4 is SET4. 0.82
 Best mapping of NUM14 is SET14. 0.82
 Best mapping of NUM15 is SET15. 0.79
 Best mapping of NUM20 is SET20. 0.66
 Best mapping of ZERO is EMPTY-SET. 0.66
 Best mapping of NUM_EQ is SET_EQ. 0.57
 Best mapping of SUM is UNION. 0.83

dependent upon explicit relational predicates in the analogs. In fact, ACME can identify isomorphisms between analogs that lack not only higher-order relations, but any relational predicates at all. To illustrate this point, Table 18A presents predicate-calculus versions of two completely arbitrary analogs, each involving three objects and three monadic predicates. Without any semantic or pragmatic information, ACME settles to a unique solution to this minimalist mapping problem after 71 cycles. The obtained mapping is given in Table 18B.

The basis for the mapping in this example is the fact that there is a unique set of element correspondences such that attributes asserted of each object in the source map consistently to attributes asserted of some object in the target. This information *could* be described in terms of relations; for example, an attribute of Bill ("smart") is *the same as* an attribute of Steve, just as an attribute of Rover ("hungry") is *the same as* an attribute of Fido. The crucial point, however, is that such relational information is computed implicitly by the ACME algorithm, and need not be explicitly coded into the propositional input representations of the analogs.

One might well question whether ACME's ability to derive semantically empty isomorphisms based solely upon monadic predicates is overpowerful relative to human analogical reasoning. Accordingly, a small experiment was performed to find out whether people can find the mapping identified

TABLE 18
Input and Output for Arbitrary Attribute-Mapping Example

A. Input analogs:

Source	(smart (Bill) f1) (tall (Bill) f2) (smart (Steve) f3) (timid (Tom) f4) (tall (Tom) f5)
Target	(hungry (Rover) s1) (friendly (Rover) s2) (hungry (Fido) s3) (frisky (Blackie) s4) (friendly (Blackie) s5)

B. Output:

Network has settled by cycle 71.
 Test: TEST3 Total times: 72
 Wed May 11 10:08:53 EDT 1988
 Abstract similarity.
 Units not yet reached asymptote: 0
 Goodness of network: 0.61
 Calculating the best mappings after 72 cycles.
 Best mapping of SMART is HUNGRY. 0.70
 Best mapping of TALL is FRIENDLY. 0.71
 Best mapping of TIMID is FRISKY. 0.54
 Best mapping of TOM is BLACKIE. 0.70
 Best mapping of STEVE is FIDO. 0.54
 Best mapping of BILL is ROVER. 0.71

by ACME for this example. The five sentences corresponding to the five propositions in each analog (e.g., "Bill is smart") were listed in adjacent columns on a piece of paper. Sentences related to the same individual were listed consecutively; otherwise, the order was scrambled. Across subjects two different orders were used. The instructions simply stated, "Your task is to figure out what in the left set of sentences corresponds to what in the right set of sentences." Subjects were also told that the meaning of the words was irrelevant. The three individuals and three attributes of the analog on the left were listed on the bottom of the page; for each element, subjects were to write down what they believed to be the corresponding element of the analog on the right. Three minutes were allowed for completion of the task.

A group of 8 UCLA students in an undergraduate psychology class served as subjects. Five subjects produced the same set of six correspondences identified by ACME, 2 subjects produced four of the six, and 1 subject was unable to understand the task. These results indicate that finding the isomorphism for this example is within the capability of many college students.

Structure and Pragmatics in Metaphor

To explore the performance of ACME in metaphorical mapping, the program was given predicate-calculus representations of the knowledge underlying a metaphor that has been analyzed in detail by Kittay (1987). The metaphor is derived from a passage in Plato's *Theaetetus* in which Socrates declares himself to be a "midwife of ideas," elaborating the metaphor at length. Table 19 contains predicate-calculus representations based upon Kittay's analysis of the source analog concerning the role of a midwife and of the target analog concerning the role of a philosopher-teacher. Roughly, Socrates claims that he is like a midwife in that he introduces the student to intellectual partners, just as a midwife often serves first as a matchmaker; Socrates helps the student evaluate the truth or falsity of his ideas much as a midwife helps a mother to deliver a child.

This metaphor was used to provide an illustration of the manner in which structural and pragmatic constraints interact in ACME. Table 19 presents predicate-calculus representations of two versions of the metaphor: an isomorphic version based directly upon Kittay's analysis, and a nonisomorphic version created by adding irrelevant and misleading information to the representation of the "Socrates" target analog. The best mappings obtained for each object and predicate in the target, produced by three runs of ACME, are reported in Table 20. The asymptotic activations of the best mappings are also presented. A mapping of "none" means that no mapping unit had an asymptotic activation greater than .20.

The run reported in the first column used the isomorphic version without any pragmatic weights. The network settles with a correct set of mappings after 34 cycles. Thus Socrates maps to the midwife, his student to the mother, the student's intellectual partner to the father, and the idea to the child. (Note that there is a homomorphic mapping of the predicates *thinks_about* and *tests_truth* to *in_labor_with*.) The propositions expressing causal relations in the two analogs are not essential here; deletion of them still allows a complete mapping to be discovered.

A very different set of mappings is reported in the middle column of Table 20 for the nonisomorphic version of the "Socrates" analog. This version provides additional knowledge about Socrates that would be expected to produce major interference with discovery of the metaphoric relation between the two analogs. The nonisomorphic version contains the information that Socrates drinks hemlock juice, which is of course irrelevant to the metaphor. Far worse, the representation encodes the information that Socrates himself was matched to his wife by a midwife; and that Socrates' wife had a child with the help of this midwife. Clearly, this nonisomorphic extension will cause the structural and semantic constraints on mapping to support a much more superficial set of correspondences between the two situations. And indeed, in this second run ACME finds only the barest fragments of the intended metaphoric mappings when the network settles

TABLE 19
 Predicate-Calculus Representations of Knowledge Underlying the Metaphor
 "Socrates is a Midwife of Ideas" (Isomorphic and Nonisomorphic Versions)

MIDWIFE (source)

(midwife (obj_midwife) m1)
 (mother (obj_ m2)
 (father (obj_father) m3)
 (child (obj_child) m4)
 (matches (obj_midwife obj_mother obj_father) m5)
 (conceives (obj_mother obj_child) m6)
 (cause (m5 m6) m7)
 (in_labor_with (obj_mother obj_child) m8)
 (helps (obj_midwife obj_mother) m10)
 (give_birth_to (obj_mother obj_child) m11)
 (cause (m10 m11) m12)

SOCRATES (target)

(philosopher (Socrates) s1)
 (student (obj_student) s2)
 (intellectual_partner (obj_partner) s3)
 (idea (obj_idea) s4)
 (introduce (Socrates obj_student obj_partner) s5)
 (formulates (obj_student obj_idea) s6)
 (cause (s5 s6) s7)
 (thinks_about (obj_student obj_idea) s8)
 (tests_truth (obj_student obj_idea) s9)
 (helps (Socrates obj_student) s10)
 (knows_truth_or_falsity (obj_student obj_idea) s11)
 (cause (s10 s11) s12)

Nonisomorphic version adds:

(father (Socrates) s20)
 (poison (obj_hemlock) s21)
 (drink (Socrates obj_hemlock) s22)
 (midwife (obj_soc-midwife) s23)
 (mother (obj_soc-wife) s24)
 (matches (obj_soc-midwife obj_soc-wife Socrates) s25)
 (child (obj_soc-child) s26)
 (conceives (obj_soc-wife obj_soc-child) s27)
 (cause (s25 s27) s28)
 (in_labor_with (obj_soc-wife obj_soc-child) s29)
 (helps (obj_soc-midwife obj_soc-wife) s30)
 (give_birth_to (obj_soc-wife obj_soc-child) s31)
 (cause (s30 s31) s32)

after 105 cycles. Socrates' midwife now maps to the midwife in the source, and Socrates' wife and child map to the source mother and child. Socrates himself simply maps to the father. Most of the other crucial objects and predicates (other than *cause* and *helps*, which map to themselves) have no good mappings. The only major pieces of the intended analogy that survive

TABLE 20
Best Mappings, With Asymptotic Activation Levels, for Objects and Predicates
in Three Versions of the Socrates/Midwife Metaphor

Cycles to Settle	Versions					
	Isomorphic, Nonpragmatic 34		Nonisomorphic, Nonpragmatic 105		Nonisomorphic, Pragmatic 83	
Objects:						
Socrates	obj_midwife	.87	obj_father	.80	obj_midwife	.86
obj_student	obj_mother	.69	obj_mother	.69	obj_mother	.69
obj_partner	obj_father	.81	none		obj_father	.80
obj_idea	obj_child	.90	obj_child	.69	obj_child	.70
*obj_soc-midwife	—		obj_midwife	.84	none	
*obj_soc-wife	—		obj_mother	.69	obj_mother	.69
*obj_soc-child	—		obj_child	.69	obj_child	.65
*obj_hemlock	—		none		none	
Predicates:						
philosopher	midwife	.58	none		midwife	.81
student	mother	.59	none		none	
intellectual_partner	father	.57	none		father	.57
idea	child	.59	none		child	.58
introduces	matches	.77	none		matches	.67
formulates	conceives	.72	conceives	.27	conceives	.31
thinks_about	in_labor_with	.36	none		none	
tests_truth	in_labor_with	.36	none		none	
knows_truth_or_falsity	gives_birth_to	.72	gives_birth_to	.29	gives_birth_to	.31
helps	helps	.77	helps	.79	helps	.80
cause	cause	.84	cause	.84	cause	.84
*poison	—		none		none	
*drink	—		none		none	
*father	—		father	.70	none	
*midwife	—		midwife	.70	none	
*mother	—		mother	.69	mother	.69
*child	—		child	.69	none	
*matches	—		matches	.78	none	
*conceives	—		conceives	.48	conceives	.43
*in_labor_with	—		in_labor_with	.74	in_labor_with	.74
*gives_birth_to	—		gives_birth_to	.46	gives_birth_to	.43

* Elements with an asterisk appeared only in nonisomorphic version. Elements that map to "none" have no mapping unit with activation greater than .20.

are the mappings between the student and the mother and between the idea and the child.

Note, however, that the original statement of the metaphor, "Socrates is a midwife of ideas," provides some direct pragmatic guidance as to the intended mappings. Clearly, Socrates must map to the midwife, and the idea

must map to *something*. This is precisely the kind of knowledge that ACME can represent using pragmatic weights. Accordingly, in a further run the mappings between propositions $s1$ and $m1$ and between the elements of those propositions (i.e., $s1 = m1$, *Socrates* = *obj_midwife*, and *philosopher* = *midwife*) were marked as PRESUMED; and proposition $s4$ and its elements (i.e., $s4$, *obj_idea*, and *idea*) were marked as IMPORTANT. The right column of Table 20 reports the results for the nonisomorphic version of the metaphor after these pragmatic weights were introduced. The pragmatic information was sufficient to allow almost complete recovery of the abstract metaphoric mappings. The network settled after 83 cycles. Socrates again maps to the midwife, and the partner to the father; almost all of the appropriate predicate mappings, such as those between *idea* and *child* and between *introduces* and *conceives*, are also recovered. Note that some of the more superficial mappings of objects, such as between Socrates' wife and the mother, also emerge. The behavior of the program across these versions of the metaphor thus dramatically illustrates both the power and the limitations of purely structural constraints, and the crucial role of pragmatic knowledge in finding abstract mappings in the face of misleading information.

Stability, Sensitivity, and Complexity Analyses

The discussion of ACME's applications is concluded by describing analyses that have been done to answer important questions about the stability, sensitivity, and computational complexity of the system. The chief stability question is whether networks settle into states in which units have stable activation values. Questions about sensitivity concern whether the performance of ACME depends upon specific parameter values or representations. Computational complexity concerns the danger of combinatorial explosion arising with large examples. These analyses show that ACME fares well on all these dimensions.

Localist connectionist networks such as the one used by ACME are sometimes unstable, undergoing oscillations in activation values of units. ACME is not subject to such instabilities for examples above, which in all of their versions have been run until stable activations have been reached by all units. In no case does ACME get a correct mapping and lose it by having an incorrect mapping catch up to, and surpass, the correct one. The stability of ACME is well typified by Figure 5, which illustrates how the activations of units proceed to stable levels. How quickly ACME settles depends upon the values of the parameters, but in no experiment has the time exceeded 220 cycles. The network typically settles in well under 100 cycles.

ACME has several important parameters, and the question naturally arises how sensitive ACME is to particular values for those parameters. The three most crucial ones are decay rate d , excitation e , and inhibition i . All of

the runs reported above have used the same values: $d = .1$, $e = .1$, and $i = -.2$. Sensitivity experiments using the laser/radiation and basic water/heat-flow examples showed that ACME performs satisfactorily over a wide range of values for these parameters. Decay rates from .001 to .2 work equally well, with the higher values producing a faster rate of settling. Higher values for excitation also lead to faster settling; values ranging from .01 to .12 work for all examples. Excitation values higher than .12 are disruptive, however, because the gracefulness of the network depends upon reasonably small updatings of activation. If excitation is above .12, then after dozens of cycles the net input to a unit can exceed 1, producing instability manifested in long settling times. Inhibition values can range all the way to -.9 without causing problems, except that high inhibition can prevent ACME from overcoming an initially promising but incorrect mapping. Although it is not crucial to the functioning of the networks, it was found desirable to have inhibition higher than excitation, because that produces greater separation in the asymptotic values of best and second-best units. In order to have a large impact, the $p2$ parameter for weights from the pragmatic unit for PRESUMED mappings works best at around .3; smaller values produce less differentiation while larger values do not produce significantly more differentiation.

How sensitive is ACME to representation changes? Although several of the examples examined involved isomorphic representations, such examples as the poor-constraint versions of the convergence problems, the missionaries-and-cannibals homomorphs, and the nonisomorphic version of the Socrates/midwife metaphor, show that ACME does not need to be spoon-fed isomorphic representations. The robustness of ACME in the face of representational changes was tested further by both complicating and simplifying existing examples. A bizarre analogy was created by combining the representations of the Contras and the Socrates analogs, and mapping it to a combination of the Hungarian and midwife analogs. ACME quickly partitioned the bizarre analogy into the appropriate sub-analogies. "Ablation" experiments were also performed on the good-constraint laser example, first by deleting every third proposition of the lightbulb problem and second, by deleting every second proposition of the tumor problem. ACME still found the appropriate mappings, except that in the first case, in which the predicate "filament" had been deleted, the tumor was left without any mapping unit with positive activation.

Finally, the computational complexity of ACME was analyzed. Mathematical analysis shows that the algorithm for constructing networks operates in time that is a polynomial (nonexponential) function of the number of elements in the input representations of the two structures that are mapped. Let $m1$ be the number of propositions in Structure 1, and $m2$ be the number of propositions in Structure 2. To simplify the algebra, let m be the higher of $m1$ and $m2$. Similarly, let n be the greater of the numbers of objects in the arguments of Structures 1 and 2. Since there can be only one predicate in

a proposition, m also gives an upper limit on the number of predicates in a structure. Then

the maximum possible number of units mapping propositions is m^*m ;
 the maximum possible number of units mapping predicates is m^*m ; and
 the maximum possible number of units mapping objects is n^*n .

So the maximum possible number of units altogether is

$$2m^2 + n^2.$$

Since there is at most only one link between any two units, if there are u units, then there can be at most $u(u-1)$ possible links, which is less than the number of units squared, or

$$4m^4 + 4m^2n^2 + n^4.$$

The semantic and pragmatic units add an additional maximum $2u$ links. Clearly, the number of links is not an exponential function of the number of elements in the input representations. In practice, the number of units and links are far lower than the calculated second- and fourth-power limits because of the various restrictions used in setting up the networks.

Unfortunately, no mathematical analysis can be given concerning how long it will take the network to settle as a function of number of units, since time of settling is a function of the degree to which the various constraints are satisfied as well as of the number of units. But the networks run so far do not take a long time to settle. Across the more than 20 examples, the greatest number of cycles to settling was 217 using the McClelland and Rumelhart updating rule and low excitation and inhibition values; the same example (the analogy between the radiation problem and the fortress problem) took only 62 cycles to settle with high excitation and decay values using the Grossberg updating rule. On a Sun 3/160 workstation, the runs typically take a few minutes, with most of that time spent in running the network to asymptote after the correct mapping units have already emerged as winners. For Falkenhainer et al.'s (1986) heat- and water-flow analogy, for example, ACME creates the network in less than 2 seconds of CPU time and settles in less than 25 seconds.

The algorithm for updating the network is fully parallel, and hence can theoretically operate in constant time if there is a processor corresponding to each link. A version of ACME in *LISP has been implemented on a 16384-processor CM2 Connection Machine, which takes advantage of the inherent parallelism of the updating algorithm. The largest example tested so far, which is considerably larger than any of the examples mentioned in Table 2, involves mapping representations of synopses of the plays *Hamlet* and *Macbeth* (cf. Winston, 1980). Each of these representations contains about 45 propositions, and the resulting network involves 730 units and 11801 links. The network settles after 402 cycles.

GENERAL DISCUSSION

A new theory of analogical mapping that integrates structural, semantic, and pragmatic constraints on how people map from one structure to another has been described here. The theory has been implemented in a connectionist computer program, ACME, which has been applied to a wide variety of analogies. The examples presented illustrate such capabilities as finding many-to-one and one-to-many mappings, mapping dissimilar relations, identifying purely structural isomorphisms without any semantic or pragmatic information, and using pragmatic knowledge to find useful mappings in the face of misleading structural and semantic resemblances. The program was able to provide qualitative simulations of a number of experimental findings concerning human analogical reasoning.

All of the examples considered here involved finding mappings between analogs at the same level of abstraction. It should be readily apparent, however, that ACME can also map structures that differ in level of abstraction, such as an instance and a schema. For example, rather than mapping a representation of the Contras onto a representation of the PLO, it would be possible to map the Contras onto a schema for "terrorists." Similarly, the same basic mapping process could map the radiation problem either onto the laser problem or onto a more abstract convergence schema (Gick & Holyoak, 1983). The present theory thus provides a unifying account of analogical mapping and mapping to schemas.

Although the constraint-satisfaction theory of analogical mapping appears powerful in its intended domain, many other important issues about analogy remain unsolved. Most notably, the model of mapping needs to be incorporated into a broader theory of all phases of analogical reasoning (Holyoak & Thagard, 1989). Of particular interest is the link between the initial spontaneous retrieval of plausibly useful analogs and the subsequent mapping process. There is evidence that retrieval is more heavily influenced by semantic similarity of predicates than is mapping (Gentner & Landers, 1985; Holyoak & Koh, 1987), although retrieval also seems to be influenced by deeper forms of structural similarity (Holyoak & Koh, 1987; Schank, 1982). Complementary to ACME, a program called ARCS (Analog Retrieval by Constraint Satisfaction) (Thagard, Holyoak, Nelson, & Gochfeld, 1989) has also been developed. ARCS is a constraint-satisfaction model of retrieval that uses semantic, structural, and pragmatic constraints to help find relevant analogs stored in memory. In contrast to ACME, semantic constraints take precedence in ARCS, with the retrieval of analogs being initiated through associations of semantically similar concepts. However, the retrieval process is also guided by structural correspondences and pragmatic import. ARCS represents an advance over ACME in that it is able to compute both semantic similarity and pragmatic centrality from more basic knowledge, rather than requiring semantic and pragmatic weights to be specified

directly. ARCS computes semantic similarity from relations such as superordination and meronymy encoded in an automated thesaurus, WordNet, which is organized as a semantic network (Miller, Fellbaum, Kegl, & Miller, 1988). Pragmatic importance is computed by an analysis of the structure of problems and explanations. The output of ARCS provides a partial mapping of the target to retrieved source analogs; it should be possible to pass the output of ARCS to ACME so that ACME can simply continue the mapping process in greater detail. (See Holyoak & Thagard, in press, for an overview of the ACME and ARCS systems.)

The major issue not addressed here by the theories of mapping and retrieval is re-representation. To find useful analogies between complex analogs, it will often be necessary to interweave the mapping component with strategic manipulation of the representations of the source and target. In order to find homomorphic correspondences involving many elements, for example, it is important to be able to group tentatively elements of each analog into sets, which could then be treated as unitary objects.

More generally, it may often be advantageous to attempt mappings at different levels of abstraction, with mappings found at higher levels then being used to constrain lower level mappings (Holyoak, 1985). The part-correspondence restriction is a first approximation to a more serial and hierarchical component of the mapping process, in which decisions about mappings at a more global level subsequently restrict the possible mappings at a more detailed level of representation. This hierarchical aspect of mapping is analogous to Marr's (1982) model of stereoscopic vision, in which correspondences between images based upon low spatial frequencies are used to constrain correspondences between images based upon higher spatial frequencies. The overall mapping process, in this extended conception, is serial across major representational levels, but more parallel within each level.

The availability of mechanisms for re-representation will make possible more flexible constraints on mapping. For example, ACME's implementation of structural constraints employs a rigid type restriction, requiring that n -place predicates map only to n -place predicates. A richer semantic representation would enable ACME to map predicates with different numbers of arguments so long as the available arguments fall into appropriate semantic categories. Similarly, a richer semantics would allow appropriate mappings between propositions involving converse relations, such as *surround* (x, y) and *enclosed-by* (y, x), which the present model cannot handle.

Finally, it is noted that the general form of the theory proposed here for analogical mapping—a set of constraints satisfiable via a cooperative algorithm—may well be applicable to other high-level cognitive processes. Lehnert (1987), for example, describes a sentence analyzer that uses a constraint network to parse sentences into case-frame meaning relationships. Similarly, Kintsch (1988) proposes a model of discourse comprehension that illustrates a constraint-satisfaction approach. Thagard (1989) and

Thagard and Nowak (1988) show how scientific theories can be evaluated on the basis of principles of explanatory coherence that are implemented in a program that overlaps with ACME. These parallel constraint-satisfaction models belie the widespread interpretation of connectionism as a straightforward revival of associationism. Connectionism has a number of important aspects, which are conceptually quite separable. In particular, whereas connectionist learning theory indeed has a strongly behaviorist flavor, parallel constraint-satisfaction models, which depend upon an analysis of the structure of problems rather than on simple associations, are much more redolent of Gestalt psychology. An abstract isomorphism, which can be computed by constraint satisfaction, is a striking example of a whole that goes beyond the sum of its parts. The parallelism of human information processing, which is so evident in lower level perception and memory retrieval, may extend to important aspects of reasoning and problem solving as well.

■ Original Submission Date: December 13, 1988.

REFERENCES

- Anderson, J.R., & Thompson, R. (1989). Use of analogy in a production system architecture. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning*. London: Cambridge University Press.
- Black, M. (1962). *Models and metaphors*. Ithaca, NY: Cornell University Press.
- Brown, A.L., Kane, M.J., & Echols, K. (1986). Young children's mental models determine transfer across problems with a common goal structure. *Cognitive Development*, 1, 103-122.
- Burstein, M.H. (1986). A model of learning by incremental analogical reasoning and debugging. In R. Michalski, J.G. Carbonell & T.M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (Vol. 2). Los Altos, CA: Kaufmann.
- Carbonell, J.G. (1983). Learning by analogy: Formulating and generalizing plans from past experience. In R. Michalski, J.G. Carbonell & T.M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach*. Palo Alto, CA: Tioga Press.
- Carbonell, J.G. (1986). Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. In R. Michalski, J.G. Carbonell & T.M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (Vol. 2). Los Altos, CA: Kaufmann.
- Coombs, C.H., Dawes, R.M., & Tversky, A. (1970). *Mathematical psychology*. Englewood Cliffs, NJ: Prentice-Hall.
- Darden, L. (1983). Artificial intelligence and philosophy of science: Reasoning by analogy in theory construction. In P. Asquith & T. Nickles (Eds.), *PSA 1982* (Vol. 2). East Lansing, MI: Philosophy of Science Society.
- Duncker, K. (1945). On problem solving. *Psychological Monographs*, 58 (Whole No. 270).
- Falkenhainer, B., Forbus, K.D., & Gentner, D. (1986). The structure-mapping engine. In *Proceedings of the Fifth National Conference on Artificial Intelligence*. Los Altos, CA: Morgan Kaufman.
- Falkenhainer, B., Forbus, K.D., & Gentner, D. (in press). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*.

- Gentner, D. (1982). Are scientific analogies metaphors? In D.S. Miall (Ed.), *Metaphor: Problems and perspectives*. Brighton, England: Harvester Press.
- Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7, 155-170.
- Gentner, D. (1989). The mechanisms of analogical reasoning. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning*. London: Cambridge University Press.
- Gentner, D., & Landers, R. (1985). Analogical reminding: A good match is hard to find. In *Proceedings of the International Conference on Systems, Man and Cybernetics*. Tuscon, AZ.
- Gentner, D., & Toupin, C. (1986). Systematicity and surface similarity in the development of analogy. *Cognitive Science*, 10, 277-300.
- Gholson, B., Eymard, L.A., Long, D., Morgan, D., & Leeming, F.C. (1988). Problem solving, recall, isomorphic transfer, and non-isomorphic transfer among third-grade and fourth-grade children. *Cognitive Development*, 3, 37-53.
- Gick, M.L., & Holyoak, K.J. (1980). Analogical problem solving. *Cognitive Psychology*, 12, 306-355.
- Gick, M.L., & Holyoak, K.J. (1983). Schema induction and analogical transfer. *Cognitive Psychology*, 15, 1-38.
- Grossberg, S. (1978). A theory of visual coding, memory, and development. In E.L.J. Leeuwenberg & H.F.J. Buffart (Eds.), *Formal theories of visual perception*. New York: Wiley.
- Halford, G.S. (1987). A structure-mapping approach to cognitive development. *International Journal of Psychology*, 22, 609-642.
- Halford, G.S., & Wilson, W.H. (1980). A category theory approach to cognitive development. *Cognitive Psychology*, 12, 356-411.
- Hall, R. (1989). Computational approaches to analogical reasoning: A comparative analysis. *Artificial Intelligence*, 39, 39-120.
- Hammond, K.J. (1986). *Case-based planning: An integrated theory of planning, learning, and memory*. Unpublished doctoral dissertation, Department of Computer Science, Yale University, New Haven, CT.
- Hesse, M.B. (1966). *Models and analogies in science*. Notre Dame, IN: Notre Dame University Press.
- Hofstadter, D.R. (1984). The Copycat project: An experiment in nondeterministic and creative analogies. A.I. Laboratory Memo 755). Cambridge, MA: MIT.
- Hofstadter, D.R., & Mitchell, M. (1988). Conceptual slippage and mapping: A report on the Copycat project. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.
- Holland, J.H., Holyoak, K.J., Nisbett, R.E., & Thagard, P. (1986). *Induction: Processes of inference, learning, and discovery*. Cambridge, MA: Bradford Books/MIT Press.
- Holyoak, K.J. (1982). An analogical framework for literary interpretation. *Poetics*, 11, 105-126.
- Holyoak, K.J. (1984). Mental models in problem solving. In J.R. Anderson & S.M. Kosslyn (Eds.), *Tutorials in learning and memory: Essays in honor of Gordon Bower*. San Francisco: Freeman.
- Holyoak, K.J. (1985). The pragmatics of analogical transfer. In G.H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 19). New York: Academic.
- Holyoak, K.J., & Koh, K. (1987). Surface and structural similarity in analogical transfer. *Memory & Cognition*, 15, 332-340.
- Holyoak, K.J., & Thagard, P. (1989). A computational model of analogical problem solving. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning*. London: Cambridge University Press.

- Holyoak, K.J., & Thagard, P. (in press). A constraint-satisfaction approach to analogue retrieval and mapping. In K.J. Gilhooly, M.T.G. Keane, R.H. Logie, & G.Erdos (Eds.), *Lines of thinking*. Chichester, England: Wiley.
- Indurkha, B. (1987). Approximate semantic transference: A computational theory of metaphors and analogies. *Cognitive Science*, *11*, 445-480.
- Kedar-Cabelli, S. (1985). Purpose-directed analogy. *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*. Irvine, CA: Cognitive Science Society.
- Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, *95*, 163-182.
- Kittay, E. (1987). *Metaphor: Its cognitive force and linguistic structure*. Oxford: Clarendon Press.
- Kolodner, J.L., & Simpson, R.L. (1988). *The MEDIATOR: A case study of a case-based problem solver* (Tech. Rep. No. GIT-ICS-88/11). Atlanta: Georgia Institute of Technology, School of Information and Computer Science.
- Kolodner, J.L., Simpson, R.L., & Sycara, E. (1985). A process model of case-based reasoning in problem solving. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (Vol. 1). San Mateo, CA: Kaufmann.
- Kunda, Z. (1987). Motivation and inference: Self-serving generation and evaluation of causal theories. *Journal of Personality and Social Psychology*, *53*, 636-647.
- Lehnert, W. (1987). Learning to integrate syntax and semantics. In P. Langley (Ed.), *Proceedings of the Fourth International Workshop on Machine Learning*. Los Altos, CA: Kaufmann.
- MacLane, S. (1971). *Categories for the working mathematician*. New York: Springer-Verlag.
- Marr, D. (1982). *Vision*. New York: Freeman.
- Marr, D., & Poggio, T. (1976). Cooperative computation of stereo disparity. *Science*, *194*, 283-287.
- 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.
- Miller, G.A. (1979). Images and models, similes and metaphors. In A. Ortony (Ed.), *Metaphor and thought*. Cambridge, England: Cambridge University Press.
- Miller, G.A., Fellbaum, C., Kegl, J., & Miller, K. (1988). WordNet: An electronic lexical reference system based on theories of lexical memory (Tech. Rep. No. 11). Princeton, NJ: Princeton University, Cognitive Science Laboratory.
- Newell, A., & Simon, H.A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.
- Novick, L.R. (1988). Analogical transfer, problem similarity, and expertise. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *14*, 510-520.
- Palmer, S.E. (1978). Fundamental aspects of cognitive representation. In E. Rosch & B.H. Lloyd (Eds.), *Cognition and categorization*. Hillsdale, NJ: Erlbaum.
- Palmer, S.E. (1989). Levels of description in information processing theories of analogy. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning*. London: Cambridge University Press.
- Polya, G. (1973). *Mathematics and plausible reasoning* (Vol. 1). Princeton, NJ: Princeton University Press.
- Reed, S.K., Ernst, G.W., & Banerji, R. (1974). The role of analogy in transfer between similar problem states. *Cognitive Psychology*, *6*, 436-440.
- Ross, B.J. (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.
- Rumelhart, D.E., Smolensky, P., McClelland, J.L., & Hinton, G.E. (1986). Schemata and sequential thought processes in PDP models. In J.L. McClelland, D.E. Rumelhart & the

- PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the micro-structure of cognition* (Vol. 2). Cambridge, MA: Bradford Books/MIT Press.
- Schank, R.C. (1982). *Dynamic memory*. Cambridge, MA: Cambridge University Press.
- Suppes, P., & Zinnes, J.L. (1963). Basic measurement theory. In R.D. Luce, R.R. Bush & E. Galanter (Eds.), *Handbook of mathematical psychology*. New York: Wiley.
- Tarski, A. (1954). Contributions to the theory of models. *Indagationes Mathematicae*, 16, 572-588.
- Thagard, P. (1984). Frames, knowledge, and inference. *Synthese*, 61, 301-318.
- Thagard, P. (1986). Parallel computation and the mind-body problem. *Cognitive Science*, 10, 301-318.
- Thagard, P. (1988a). *Computational philosophy of science*. Cambridge, MA: Bradford Books/MIT Press.
- Thagard, P. (1988b). Dimensions of analogy. In D. Helman (Ed.), *Analogical reasoning*. Dordrecht, Holland: Kluwer.
- Thagard, P. (1989). Explanatory coherence. *Behavioral and Brain Sciences*, 12, 435-467.
- Thagard, P., Cohen, D., & Holyoak, K.J. (1989). Chemical analogies. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*. San Mateo, CA: Kaufmann.
- Thagard, P., Holyoak, K.J., Nelson, G., & Gochfeld, D. (1989). *Analog retrieval by constraint satisfaction*. Unpublished manuscript, Princeton University, Cognitive Science Laboratory.
- Thagard, P., & Nowak, G. (1988). The explanatory coherence of continental drift. In A. Fine & J. Leplin, (Eds.), *PSA 1988* (Vol. 1, pp. 118-126). East Lansing, MI: Philosophy of Science Association.
- Tourangeau, R., & Sternberg, R.J. (1982). Understanding and appreciating metaphors. *Cognition*, 11, 203-204.
- Tversky, A. (1977). Features of similarity. *Psychological Review*, 84, 327-352.
- Winston, P.H. (1980). Learning and reasoning by analogy. *Communications of the ACM*, 23, 689-703.
- Ziegler, W.P. (1976). *Theory of modelling and simulation*. New York: Wiley.