Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioral and Brain Sciences, 11* (1), 1–14.

Treisman, A. (1988). Features and Objects: The Fourteenth Bartlett Memorial Lecture. *Quarterly Journal of Experimental Psychology, 40A*, 201–237.

in connectionist and neural computation theory. Vol. 2: Analogical Connections (pp. 113–180), Norwood, NJ: Ablex. 1994.

2

Component Processes in Analogical Transfer: Mapping, Pattern Completion, and Adaptation*

Keith J. Holyoak
Laura R. Novick
Eric R. Melz

1. INTRODUCTION

Analogical transfer from a source to a target analog depends on several component processes. Although proposed taxonomies of component processes have differed in detail, there has been broad agreement that it is useful to distinguish at least six major steps: (a) forming mental representations of the source and target analogs, (b) retrieving a potentially useful source analog from memory, (c) finding a mapping (i.e., set of correspondences) between the elements of the source and target, (d) deriving inferences based on the mapping, (e) evaluating and adapting the inferences to satisfy constraints required by the target situation, and (f) learning new generalizations in the aftermath of analogical transfer (e.g., Carbonell, 1983; Gentner, 1983, 1989; Gick & Holyoak, 1980; Hall, 1989; Keane, 1988; Novick, 1988; Novick & Holyoak, 1991). These processes need not be performed in a strict serial fashion; rather, there is reason to believe they operate partially in parallel and with considerable interdependence (cf. Eskridge, this volume).

* Preparation of this chapter was supported by Contract MDA 903-89-K-0179 from the Army Research Institute to K. Holyoak, and by a grant from the Vanderbilt University Research Council and a fellowship from the National Academy of Education to L. Novick. The computational modeling was supported by NSF Biological Facilities Award BBS 87-1420, NSF Grant DIR-9024251, and a grant from the Keck Foundation, all to the UCLA Cognitive Science Research Program. We thank John Barnden, Liane Gabora, and Greg Nelson for their comments on earlier drafts.

In the present chapter, we will leave aside issues involving the crucial processes of analog representation, retrieval, and learning to focus on the "central" processes of analogical mapping, inference, and evaluation/adaptation. When executed appropriately, these processes collectively yield what we will refer to as *analogical transfer*. We will argue that the three processes are in fact psychologically distinct—for example, an analogist may succeed in deriving a reasonable mapping between a solved source problem and an unsolved target problem, yet be unable to perform the inference and/or evaluation processes required to use the mapping to generate a useful solution to the target problem (also see Novick & Holyoak, 1991, on this point). In addition, we will consider the extent to which these various subprocesses of analogical transfer are amenable to modeling using connectionist-style mechanisms. Our theoretical starting point will be the ACME (Analogical Constraint Mapping Engine) model of mapping developed by Holyoak and Thagard (1989), a hybrid system which combines symbolic representations of knowledge with connectionist-style constraint satisfaction to model human analogical mapping (Holyoak, 1991). We will argue that with some simple extensions, the ACME model can also account for basic post-mapping analogical inferences; but that without much more extensive augmentation, the more open-ended processes of evaluation and adaptation lie beyond the model's capabilities. More generally, the differential success of humans at various stages of analogical transfer may suggest the need to integrate connectionist mechanisms with rule-based reasoning (cf. Nelson, Thagard, & Hardy, this volume).

1.1. Differentiating Mapping from Postmapping Processes

The central focus of theory and research on analogy has been the process of analogical mapping: the identification of a set of orderly correspondences between the elements of the source and target (e.g., Gentner, 1983; Falkenhainer, Forbus, & Gentner, 1989; Holyoak & Thagard, 1989). There are compelling reasons why mapping is viewed as the sine qua non of analogical reasoning. The fundamental intuition underlying the concept of analogy is that two situations can resemble each other by virtue of a kind of "configural" similarity based on systematic role correspondences, even though the specific objects that play corresponding roles in the two analogs, or even the predicates applied to these objects, may lack direct similarities. That is, the existence of a "good" mapping, in the sense of an isomorphism (i.e., consistent one-to-one correspondences), is what in fact defines an analogy. Accordingly, the effectiveness of the retrieval process is largely judged by the extent to which it recovers from memory source analogs that map well to their targets (see Lange & Wharton, 1994). Furthermore, the postmapping process of inference must use the set of correspondences established by the mapping process in order to generate plausible inferences about the target.

Although mapping plays a central role in analogical transfer, transfer requires additional postmapping processes as well. Somewhat unfortunately, psychological research on analogical transfer has seemed to emphasize mapping to such an extent that postmapping processes have been neglected (but see Novick & Holyoak, 1991). Indeed, analogical transfer has sometimes been virtually identified with mapping, and vice versa. For example, Gick and Holyoak (1980) introduced a problem-solving paradigm in which subjects first receive, in an incidental context such as instructions to memorize a story, a source analog involving a problem and its solution. They are then presented with a target problem to solve, first without any hint that the source story is relevant. If subjects fail to give the analogous solution to the target, they are then given a direct hint to try to apply the source story. The proportion of subjects who generate the analogous solution prior to the hint (relative to control subjects who received no analog) has generally been interpreted as a measure of how often people retrieve the source and map it successfully to the target; the additional proportion of subjects who generate the analogous solution only after a hint has been interpreted as a measure of how many failed to spontaneously retrieve the source, even though they were capable of finding the appropriate mapping once they tried. In other words, solution frequency after a hint is provided (which obviates the need to spontaneously retrieve the source) is taken as a measure of ease of mapping.

In fact, however, successful transfer of a problem solution requires not only mapping, but also (at least) some follow-up inference process to generate the analogous solution to the target. For example, consider the analogs used by Gick and Holyoak (1980). The target problem was Duncker's (1945) "radiation problem," in which a doctor must find a way to use rays to destroy an inoperable stomach tumor, without damaging the healthy tissue that the rays must pass through on the way to the tumor. The source problem was provided by a story in which a general needed to get a large army to a fortress located in the center of a country, in order to capture it. Because the entire army could not pass safely along any one road, the general divided the army into small groups and had each group travel along one of several roads that each led to the fortress. The groups converged simultaneously on the fortress, where the combined forces succeeded in capturing it.

What is required to generate an analogous "convergence" solution to the radiation problem, given that the source is available for use? The mapping process presumably must identify at least some of the potential role correspondences between the source and target (e.g., the doctor corresponds to the general, the tumor to the fortress, and the rays to the army). But in and of themselves, such correspondences by no means provide an analogous solution to the target problem. By definition, mapping can identify corresponding elements between the two analogs, such as objects, predicates, and propositions. But in a typical case requiring analogical transfer, as in solving the radiation problem, a solution to the target problem in initially lacking. It is therefore logically impossible to directly "map" the source solution onto the target problem, because the representation of the latter lacks the requisite elements to which the source solution is supposed to map. Not only does the unsolved target necessarily lack propositions describing its solution, but it may also lack objects and/or predicates that are crucial to constructing the analogous solution. In our example, the statement of the radiation problem does not mention the predicate "divide," which is a central element in the convergence solution to the fortress problem, and which must be somehow transferred from the fortress problem to the radiation problem in the course of generating an analogous solution of "dividing the rays."

1.2. Analogical Inference as Pattern Completion

It follows from the above analysis that those subjects in the Gick and Holyoak study who succeeded in generating the convergence solution to the radiation problem must have not only mapped at least some of the corresponding elements in the source and target, but also must have gone on to generate inferences that "filled in" missing information about the target analog. What is striking about the results of these early experiments on analogical problem solving is that the postmapping inferences seemed so easy for subjects as to virtually escape notice. Roughly 80% of subjects succeeded in generating the convergence solution once they received a hint to use the fortress problem, as compared to a mere 10% of control subjects. Furthermore, subjects who talked aloud as they solved the radiation problem typically did not reveal a laborious process of mapping and inference; rather, they usually simply stated the solution in a fairly direct way, with minimal reference to intermediate inference steps. Their performance was thus consistent with the possibility that most of the mental work involved in analogical mapping and inference is often performed rapidly and largely unconsciously.

More generally, one might hypothesize that basic analogical inference accomplishes something similar to what in connectionist networks is termed "automatic pattern completion." If a source and target analog can be successfully mapped, it seems that people are able to readily use the mapping to fill in "gaps" in one (or perhaps both) of the analogs. Of course, there is no reason to assume that the mechanisms responsible for analogical pattern completion are the same as those used in connectionist networks that perform pattern completion. In particular, people are clearly able to make analogical inferences from a single source analog to a single target, without extensive prior training on multiple examples of analogs of the same type, as is required by most connectionist learning procedures. Nonetheless, there may be a natural "cut" between components of analogical transfer that depend only on mapping and pattern completion, versus subsequent components that depend on more elaborate reasoning. We will now consider in more detail both theoretical and empirical arguments in favor of this possibility.

2. ANALOGICAL MAPPING, PATTERN COMPLETION, AND ADAPTATION

2.1. Theoretical Bases for Distinguishing Component Processes

On what bases can we distinguish among mapping and other postmapping components of transfer? Although the exact boundaries undoubtedly blur in actual analogical transfer, it seems possible to distinguish three different components, roughly corresponding to what we think of as mapping, pattern completion (i.e., analogical inference), and further evaluation or adaptation. As nearly all researchers have agreed on the centrality of mapping to analogy (e.g., Gentner, 1983; Holyoak & Thagard, 1989), the crucial issue concerns distinguishing the latter two processes from the mapping process and from each other.

2.1.1. Mapping. Given the initial active representations of the source and target, the mapping process derives a set of correspondences between the elements of the two analogs, where the elements consist of objects, predicates, and propositions. As we have emphasized, mapping can directly establish correspondences only between elements that are present in the initial representations of the two analogs.

2.1.2. Pattern completion. If one wishes simply to verify the analogical status of two isomorphic representations, the mapping process will suffice. In situations of analogical transfer, however, in which information from the source domain must be transferred to the target domain (for example, to provide a solution to the target problem), the mapping process by itself is insufficient for the task at hand. Rather, analogical transfer requires postmapping processes in which new inferences are drawn using the correspondences established by the mapping, in conjunction with the structure of the analogs. Such processes can effectively fill in gaps of missing information when knowledge from a well-understood domain is transferred to an analogous but less well-understood domain. We will refer to the generation of inferences based on analogical mapping as *analogical pattern completion*.

The simplest basic mechanism for analogical pattern completion is *copying with substitution* (CWS), which in some form has been included in all computational models of postmapping analogical transfer (e.g., Burstein, 1986; Carbonell, 1983, 1986; Falkenhainer et al., 1989; Hofstadter & Mitchell, 1988, this volume; Winston, 1980). The simplest form of CWS applies when the mapping process has provided correspondences for all relevant objects and predicates, but some propositions in one or both analogs are left unmapped. The intuition behind this procedure is that in analogical transfer, there is a pressure for propositions in the source to have corresponding propositions in the target, and vice versa. If some proposition exists in one analog, but has no corresponding proposition in the other, and if all of the constituent elements of the existing proposition map to elements in the other analog, we may reasonably conjecture a new proposition in the other analog. Formally, the CWS procedure used to generate propositions in the aftermath of mapping, in the extension of ACME we describe in this chapter, is the following:

- If a proposition P consisting of relation r and objects a and b (notated $P{:}r(a,b)$) exists in the source but does not map to any corresponding proposition in the target, and
- if P's relation and objects have the mappings
  $r->r'$
  $a->a'$
  $b->b'$,
- then create the new proposition $P'{:}r'(a',b')$ in the target.

Although stated for two-place relations, the above procedure can readily be generalized to apply to propositions based on n-place predicates, including logical connectives and higher order predicates that take propositions as arguments (Falkenhainer et al., 1989).

This version of CWS requires that the mapping process has provided correspondences for all predicates and objects that participate in unmapped propositions. But as we have seen in the fortress/radiation problem analogy, it is possible that the source may contain predicates or objects that have no direct correspondence to any element of the initial target representation (e.g., the operator "divide" in the solution to the fortress problem has no immediate match in the representation of the unsolved radiation problem). In such cases, basic CWS must be augmented with procedures for retrieving or generating additional predicates or objects in the more impoverished analog (cf. Kokinov, this volume). To a first approximation, augmented CWS can operate by identifying an unmapped proposition P as before, and then:

- if P contains an unmapped predicate or object, postulate an "image" predicate or object in the other analog, and
- then proceed with CWS as usual.

CWS augmented with predicate/object generation ("copy with substitution and generation," or "CWSG") does not be itself fully determine the actual content of the generated image elements. It does, however, automatically bind each image object or image predicate into roles parallel to those of its corresponding source object or predicate across all relevant propositions that are generated.

CWSG thus produces a description of a "desired" image object or predicate, where the detail of the description will depend on the richness of the set of generated propositions in which the image element appears.

This description may trigger a memory search for a specific object or predicate corresponding to that description (Kokinov, this volume), or the description may serve as the basis for postulating a previously unknown concept. We will consider the generation of a description of an image element to be the final output of the process of analogical pattern completion, with subsequent search and evaluation processes considered part of the later adaptation stage. Element generation thus "straddles the line" between pattern completion and subsequent adaptation.

2.1.3. Adaptation. Whereas mapping involves a consideration of the relationships between active representations of the source and target analogs, and pattern completion considers the mapping plus the structure of each analog, there comes a point in analogical transfer at which attention must focus directly on the *unique* aspects of the target. That is, analogical mapping and pattern completion in effect propose plausible inferences about the target, but the pragmatic test of whether the analogy is useful hinges on whether these inferences are in fact sufficient to achieve the analogist's goals with respect to the target domain. The evaluation/adaptation stage may reject inferences that prove to be erroneous, or add additional supplemental information derived from the target domain itself rather than by analogy to the source. Such supplemental information about the target domain may be supplied by memory retrieval in the aftermath of pattern completion (as in the case of finding an actual instantiation of an object or predicate for which CWSG has produced a description), or by additional inferences about the target situation. These target-based inferences may be integrated with analogy-based inferences to achieve the applicable goals for the target domain, such as finding a solution.

The distinction between the roles of pattern completion and adaptation is closely related to the difference between isomorphism of the source and target domains versus isomorphism of the *representations* of the source and target domains. Pattern completion embodies the tacit assumption that the underlying source and target domains are in fact isomorphic. Thus, if the active representations are not isomorphic, pattern completion will attempt to fill the apparent gaps, thus (re)constructing the underlying isomorphism assumed to hold between the domains. But of course, there is no guarantee that the underlying domains are in fact fully isomorphic; indeed, this will seldom be the case for domains of realistic complexity. Adaptation will be required whenever the underlying structures of the source and target domains are not completely isomorphic. If the source has elements that lack any correspondents in the target, erroneous inferences about the target may be generated (unless pragmatic knowledge can be used at earlier stages to block the generation of inferences based on irrelevant aspects of the source). If the target domain has elements that do not correspond to anything in the source, then additional target-generated inferences will be required to supplement those generated by analogy. The process of adaptation is "open-ended" in the sense that it depends on general reasoning procedures and memory-search processes that go beyond the information provided by the analogy. Thus, in many cases, we would expect a sharp "break" in transfer performance when adaptation is required: whereas mapping and pattern completion are guided by the structural relationships between source and target, adaptation depends upon knowledge of the target domain that goes beyond (and in fact may contradict) what can be generated by using the source-target relations. We now consider empirical evidence that bears on this hypothesis.

2.2. Empirical Evidence for Distinguishing Component Processes

Although ease of mapping can be inferred from ease of transfer, it does not follow that difficulty of mapping can be inferred from difficulty of transfer. In the latter case, subjects may have constructed the appropriate mapping, encountering difficulty instead with the later adaptation process. Because the typical dependent measure in the experimental literature has been success of transfer rather than of mapping, the ease or difficulty of the mapping process is uncertain. The resulting ambiguity concerning the contribution of mapping to the difficulty of analogical transfer creates problems for assessing the role of adaptation. Nevertheless, the existing data are consistent with the hypothesis that mapping and pattern completion are easier to perform than adaptation.

2.2.1. Evidence that Mapping and Pattern Completion are Relatively Easy to Perform. Work by Reed (1987) provided evidence that mapping and pattern completion are fairly easy for "isomorphic" algebra word problems of two types that are common in early algebra courses: "mixture" problems, in which multiple entities of different concentrations or values are mixed together to form an entity of some intermediate concentration or value; and "work" problems, in which two or more agents work together to complete a task that each could complete alone. Tables 2.1 and 2.2 present examples of the problems that Reed used, along with the correspondences required to solve a mapping task that Reed administered, and the percent correct obtained by subjects for each question. Subjects (students enrolled in undergraduate psychology courses) were given six numerical quantities from the source problem (e.g., "12*hr*," "1/8 *tank/hr*," "4.5 *pt.* × 8% *acid*"), and for the first of these they were told the corresponding quantity in the target problem. Their task was to match the remaining five source quantities to the corresponding target quantities. The subscript after each target correspondent listed in Tables 2.1 and 2.2 indicates whether it can theoretically be computed simply by: (a) constructing a mapping, (b) mapping plus pattern completion, or (c) adaptation that goes beyond pattern completion. For isomorphic problems, all the required correspondences can be computed

Table 2.1.
"Isomorphic" and "Similar" Work Problems

Source (Pipe Problem)

A small pipe can fill an oil tank in 12 hours and a large one can fill it in 8 hours. How long will it take to fill the tank if both pipes are used at the same time?

Solution equation: (1/12)h + (1/8)h = 1

"Isomorphic" Target (Typing Problem)

Ann can type a manuscript in 10 hours and Florence can type it in 5 hours. How long will it take them if they both work together?

Solution equation: (1/10)h + (1/5)h = 1 [76% correct]

"Similar" Target (Tank Problem)

A small pipe can fill a water tank in 20 hours and a large pipe can fill it in 15 hours. Water is used at a rate that would empty a full tank in 40 hours. How long will it take to fill the tank when both pipes are used at the same time, assuming that water is being used as the tank is filled?

Solution equation: (1/20)h + (1/15)h − (1/40)h = 1 [20% correct]

Percent Correct on Mapping Task
(boldface indicates information given to subjects)

Source (Pipe)	Isomorphic Target (Typing)	Similar Target (Tank)
8 hr	**5 hr**	**15 hr**
12 hr	10 hr (91%)[1]	20 hr (86%)[1]
1/8 tank/hr	1/5 ms/hr (91%)[2]	1/15 tank/hr (84%)[2]
1/12 tank/hr	1/10 ms/hr (88%)[2]	1/20 tank/hr (84%)[2]
(1/8 tank/hr) × (h hr)	(1/5 ms/hr) × (h hr) (88%)[2]	(1/15 tank/hr) × (h hr) (81%)[2]
(1/12 tank/hr) × (h hr)	(1/10 ms/hr) × (h hr) (93%)[2]	(1/20 tank/hr) × (h hr) (60%)[2]

Note: Footnote numbers refer to the transfer stage theoretically required to compute the correspondence: 1 = mapping; 2 = pattern completion; 3 = adaptation.

Adapted from "A Structure-Mapping Model for Word Problems," by S.K. Reed, 1987, *Journal of Experimental Psychology: Learning, Memory and Cognition, 13.* Reprinted by permission.

either by mapping alone (e.g., 12 *hr* = 10 *hr*, where the mapped terms are each directly provided by the problem statements), or by pattern completion (e.g., 1/8 *tank/hr* = 1/5 *ms/hr*, where the source element appears in the solution equation and the corresponding target element must be constructed by CWSG). Across four isomorphic source/target pairs, an average of 79% of the subjects correctly matched each quantity (Reed, 1987, Experiment 4). (Tables 2.1 and 2.2 report the mean percent correct on the mapping questions, and on an equation-transfer task described below, for the representative examples given in that table. We will use these particular examples in an ACME simulation reported below.)

Table 2.2.
"Isomorphic" and "Similar" Mixture Problems

Source (Nurse Problem)

A nurse mixes a 6% boric acid solution with a 12% boric acid solution. How many pints of each are needed to make 4.5 pints of an 8% boric acid solution?

Solution equation: .06a + (.12)(4.5-a) = (.08)(4.5)

"Isomorphic" Target (Grocer Problem)

A grocer mixes peanuts worth $1.65 a pound and almonds worth $2.10 a pound. How many pounds of each are needed to make 30 pounds of a mixture worth $1.83 a pound?

Solution equation: 1.65a + (2.10)(30-a) = (1.83)(30) [36% correct]

"Similar" Target (Alloy Problem)

One alloy of copper is 20% pure copper and another is 12% pure copper. How much of each alloy must be melted together to obtain 60 pounds of alloy containing 10.4 pounds of copper?

Solution equation: .20a + (.12)(60-a) = 10.4 [0% correct]

Percent Correct on Mapping Task
(boldface indicates information given to subjects)

Source (Nurse)	Isomorphic Target (Grocer)	Similar Target (Alloy)
6% acid	**$1.65**	**20% copper**
12% acid	$2.10 (81%)[1]	12% copper (91%)[1]
8% acid	$1.83 (84%)[1]	10.4/60 copper (2%)[3]
4.5 pt	30 lbs (86%)[1]	60 lbs (81%)[1]
4.5 − a pt	30 − a lbs (74%)[2]	60 − a lbs (77%)[2]
(4.5 pt) × (8% acid)	(30) × ($1.83) (81%)[2]	10.4 lbs (0%)[3]

Note: Footnote numbers refer to the transfer stage theoretically required to compute the correspondence: 1 = mapping; 2 = pattern completion; 3 = adaptation.
Adapted from "A Structure-Mapping Model for Word Problems," by S.K. Reed, 1987, *Journal of Experimental Psychology: Learning, Memory and Cognition, 13.* Reprinted by permission.

As noted above, Holyoak's work with Duncker's (1945) radiation problem also provided evidence for the ease of mapping and pattern completion (also see Keane, 1988). Gick and Holyoak (1980, 1983) found in five experiments that when subjects were told to solve this problem using an earlier story about a general capturing a fortress, 80% were successful in producing the analogous "convergence" solution. Holyoak and Koh (1987) found a transfer rate of 78% when the source story concerned the repair of a filament inside a lightbulb. In contrast, only about 10% of subjects produced the convergence solution to the tumor problem when no story was presented. Note that although the goals and solution constraints for all three problems are analogous (goal = attack a centrally located target; constraint = a large force sent down a single path will result in undesired destruction), the problems are not completely isomorphic. For example, in the fortress problem the force itself (i.e., the army) will be destroyed if the entire army marches down a single road. In the tumor problem use of a high-intensity ray from a single machine will kill the healthy tissue surrounding the tumor, but it will have no deleterious effect on the ray. Thus, these results indicate that a complete isomorphism is not a prerequisite for successful mapping and pattern completion.

What about studies suggesting that mapping is difficult? In many cases, the results are consistent with the hypothesis that the difficulty lies in the subsequent adaptation process. In his mapping experiment described above, Reed (1987) also had subjects determine corresponding quantities for source/target pairs that were "similar" but not isomorphic (see Tables 2.1 and 2.2). For these pairs, the equation for the source problem had to be modified before it could be used to solve the target problem. For example, in the source mixture problem stated in Table 2.2, 6% and 12% acid solutions were mixed to yield 4.5 pints of an 8% solution. The following equation can be used to determine how many pints of each solution were used: .06*a* + (.12)(4.5-*a*) = (.08)(4.5). Note that the amount of acid in the mixture is obtained by multiplying the concentration of the mixture (.08) by the number of pints in the mixture (4.5). In a similar but nonisomorphic target problem, two alloys of 20% and 12% pure copper were melted together to obtain 60 pounds of a new alloy containing 10.4 pounds of copper. To solve this problem, the procedure for finding the amount of pure ingredient in the mixture must be adapted from that of multiplying a concentration times a total quantity to that of retrieving the relevant amount from the problem statement (i.e., a correct equation is: .20*a* + (.12)(60-*a*) = 10.4). Note that on the surface, the required solution equation for the similar target problem is simpler than that for the source problem, because the amount of pure ingredient in the mixture is given directly in the target problem (10.4), unlike the case for the source where it must be calculated as a proportion of the mixture ((.08)(4.5)). But in terms of the analogical relationship between the two quantities, the "simpler" part of the target equation cannot be derived by mapping and pattern completion; hence, it requires adaptation.

Consistent with the hypothesis that mapping and pattern completion are relatively easy to perform, when given the correspondence "6% acid maps to 20% copper," an average of 83% of the subjects produced each of the following correct mappings: 12% *acid* = 12% *copper* and 4.5 *pt.* = 60 *lb.*, both derivable by mapping alone, and 4.5-*a pt.* = 60-*a lb*, derivable by pattern completion. But, consistent with the hypothesis that adaptation is difficult, none of the subjects correctly mapped 4.5 *pt.* × 8% *acid* onto 10.4 *lb.*, and only 2% succeeded in mapping 8% *acid* onto 10.4/60 *copper.*

Not all of Reed's "similar" problems resulted in difficulties on the mapping task. Note that for the similar work target in Table 2.1, the nonisomorphism results from the addition of an extra variable in the target (rate of water use) that is not present in the source. Because the change from source to target consists entirely of an addition of elements, every source element in fact has a correspondent in the target that can be computed by mapping alone or by mapping plus pattern completion. Thus, for this similar target problem, mapping accuracy was uniformly high for all source quantities. Thus, even for nonisomorphic problems, the mapping task was not difficult as long as adaptation was not required to generate the appropriate correspondents in the target.

Another study that provided evidence distinguishing adaptation from mapping was performed by Novick and Holyoak (1991). A reinterpretation of their data also distinguishes pattern completion from the other two transfer processes. Table 2.3 presents two problems that were used in their research (and also by Novick, 1988) to investigate analogical transfer in mathematical problem solving. College students first studied the "garden" problem, plus a solution to it based on finding multiples of the lowest common multiple of several numbers. They then attempted to solve the target "band" problem using the garden problem as a source analog. In addition, some subjects were explicitly asked to state the source correspondents for various key concepts or numbers in the band problem. For example, the band members should map onto plants, the number of members in a row or column onto the number of plants of a kind, and the successful divisor (5, which leaves a zero remainder) onto the successful divisor (6) in the garden problem.

Note that the two problems have many surface dissimilarities (e.g., band members have no obvious resemblance to plants), contain some misleading similarities (e.g., the divisor 5 in the band problem should map onto 6, not 5, in the garden problem), and are far from isomorphic: for example, (a) the band problem involves two people who consider a single total number of band members, whereas the garden problem involves three people who consider two different possible total numbers of plants; (b) the solution constraint in the garden problem is to find the smallest possible total, whereas the constraint in the band problem is to find a possible total within a given range; and (c) the solution to the garden problem is based on the second multiple of the LCM, whereas the solution to the band problem is based on the sixth multiple. There is, therefore, good reason to expect that mapping these two analogs would be challenging. Nonetheless, the college students tested by Novick and Holyoak (1991) achieved over 80% accuracy in providing the correct mappings for the key concepts and numbers. Oral protocols collected from some subjects revealed few overt signs of the mapping process, consistent with the use of an easily executed and relatively fast mapping mechanism.

Novick and Holyoak (1991) also obtained evidence that pattern completion is performed relatively easily. In two experiments, they found that of those subjects

Table 2.3.
Analogous Mathematical Word Problems

Garden Problem (Source)

Mr. and Mrs. Renshaw were planning how to arrange vegetable plants in their new garden. They agreed on the total number of plants to buy, but not on how many of each kind to get. Mr. Renshaw wanted to have a few kinds of vegetables and ten of each kind. Mrs. Renshaw wanted more different kinds of vegetables, so she suggested having only four of each kind. Mr. Renshaw didn't like that because if some of the plants died, there wouldn't be very many left of each kind. So they agreed to have five of each vegetable. But then their daughter pointed out that there was room in the garden for two more plants, although then there wouldn't be the same number of each kind of vegetable. To remedy this, she suggested buying six of each vegetable. Everyone was satisfied with this plan. Given this information, what is the fewest number of vegetable plants the Renshaws could have in their garden?

Solution: Since at the beginning Mr. and Mrs. Renshaw agree on the total number of plants to buy, 10, 4, and 5 must all go evenly into that number, whatever it is. Thus, the first thing to do is to find the smallest number that is evenly divisible by those 3 numbers, which is 20. So the original number of vegetable plants the Renshaws were thinking of buying could by any multiple of 20 (that is, 20 or 40 or 60 or 80, etc.). But then they decide to buy 2 additional plants, that they hadn't been planning to buy originally, so the total number of plants they actually end up buying must be 2 more than the multiples of 20 listed above (that is, 22 or 42 or 62 or 82, etc.). This means that 10, 4, and 5 will now no longer go evenly into the total number of plants. Finally, the problem states that they agree to buy 6 of each vegetable, so the total number of plants must be evenly divisible by 6. The smallest total number of plants that is evenly divisible by 6 is 42, so that's the answer.

Band Problem (Target)

Members of the West Side High School Band were hard at work practicing for the annual Homecoming Parade. First they tried marching in rows of twelve, but Andrew was left by himself to bring up the rear. The band director was annoyed because it didn't look good to have one row with only a single person in it, and of course Andrew wasn't very pleased either. To get rid of this problem, the director told the band members to march in columns of eight. But Andrew was still left to march alone. Even when the band marched in rows of three, Andrew was left out. Finally, in exasperation, Andrew told the band director that they should march in rows of five in order to have all the rows filled. He was right. This time all the rows were filled and Andrew wasn't alone any more. Given that there were at least 45 musicians on the field but fewer than 200 musicians, how many students were there in the West High School Band?

From "Mathematical Problem Solving by Analogy," by L.R. Novick & K.J. Holyoak, 1991, *Journal of Experimental Psychology: Learning, Memory, and Cognition, 17.* Reprinted by permission.

who attempted to transfer the source solution to the target problem, none incorrectly substituted the band numbers into the operators used to solve the garden source problem.[1] Although numerous errors in transferring the source solution procedure to the target problem were observed once subjects had correctly substituted the numbers, only a minority of them could be attributed to the pattern completion process: 27% of the errors involved generation of some but not all of the required image predicates in the target that corresponded to the source operators.

Note that although the task of pattern completion appears relatively easy to perform, it is not necessarily accurate. When applying a representationally complete source to generate inferences to complete a representationally impoverished target, we can only reasonably expect pattern completion to generate useful inferences if the underlying source and target domains are in fact isomorphic. For example, given a source problem and solution, and a superficially similar target problem, it is a trivial matter to copy the source operators and substitute the corresponding target elements into the source operators to generate a potential target solution. However, if the solution procedure required for the target problem differs at all from the source solution procedure, the solution produced by pattern completion will at least require adaptation, and may even prove useless (Novick, 1988). As we shall see, both human subjects and ACME have little difficulty performing the task of pattern completion; however, the accuracy of the inferences generated by this process depend on the degree to which the source and target domains are in fact isomorphic.

2.2.2. Evidence that Adaptation is Relatively Difficult to Perform. Further support for our hypothesis that the primary difficulty in analogical transfer lies in adaptation comes from a comparison of the results of Reed's (1987) mapping experiment, discussed above, to the results of an experiment in which he asked subjects (students enrolled in a college algebra class) to use the source problems to construct appropriate equations for the "isomorphic" and "similar" target problems. As we noted above, for isomorphic problems the analogous equation for the target can be constructed by mapping and pattern completion, whereas for the similar problems adaptation is also required. Over all the problems Reed used, the transfer (i.e., equation accuracy) rates for the isomorphic and similar target problems were 53% and 11%, respectively (Reed, 1987, Experiment 3), indicating that transfer was indeed much more difficult when adaptation was necessary. Average accuracy rates for other subjects on the mapping task were 79% and 63%, respectively, for the isomorphic and similar targets (Experiment 4). As we saw earlier, the reduction in mapping accuracy for similar target problems was almost entirely due to those specific questions for which adaptation was required to generate the correct correspondent in the target. The correlation between transfer and accuracy on the mapping task across the eight target problems was .77, indicating a clear statistical link between the two tasks. The difficulty observed in mapping and/or generating those quantities in the nonisomorphic problems that were involved in the required procedure adaptation, coupled with the low rate of equation transfer for those problems, provides evidence for the difficulty and importance of adaptation in the solution of mathematical problems by analogy.

[1] Approximately one-third of the subjects failed to provide evidence in their written solution protocols of attempting to solve the band problem by analogy to the garden problem. For these subjects, it is impossible to tell whether they generated the appropriate image predicates in the target problem but then did not know how to execute them, or whether they failed to even generate the analogous operators.

Reed, Dempster, and Ettinger (1985, Experiment 3) provided more direct evidence for the difficulty of adaptation, based on their analysis of the errors subjects (enrolled in a college algebra task) made when they failed to construct the correct equation. For the isomorphic target problems, the errors were evenly distributed among quantity errors (wrong numbers substituted into the correct equation), equation errors (generating an incorrect form of the equation), and failures to generate any equation (11% in each category). For the similar target problems, quantity errors were again rare (11% of solutions), but equation errors were very common. The latter accounted for 51% of all solution attempts, with 64% of those errors representing a failure to attempt adaptation of the source equation for use with the target problem, and 36% representing an incorrect adaptation of the source equation. The difficulty of adaptation was further evidenced by an increase in solution attempts that did not involve the use of equations (31% of solutions).

In the studies by Novick and Holyoak (1991) using the garden and band problems, it was also the case that knowing the correct mapping did not guarantee successful transfer of the solution procedure. As noted above, subjects answered about 80% of the mapping questions correctly. Furthermore, in some conditions subjects were directly told several mappings (either of concepts or of numbers) before they attempted to transfer the solution procedure from the source to the target. The most helpful of these mapping hints was to provide correspondences between the numbers that played the same roles in the solution procedures. (This hint ensures that subjects know the mappings that CWS and CWSG require for substitution.) But even for subjects who received this number-mapping hint, 40–50% (in two different experiments) failed to generate the correct procedure for the target band problem. Furthermore, oral protocols, which as we noted above revealed little direct evidence of the mapping process, consisted largely of laborious efforts to work out the *implications* of the correspondences found between the two analogs, after the initial mapping process was apparently completed.

For those subjects (in all conditions) who provided evidence of attempting to apply the source solution procedure to the band target problem (approximately two-thirds of the subjects in each of two experiments), Novick and Holyoak

(1991) analyzed the source of the errors made. As indicated earlier, only 27% of the errors (collapsed across the two experiments) can be attributed to the pattern completion process. The remaining 73% of the errors are the result of difficulties encountered during adaptation. These difficulties fall into two categories. The largest source of difficulty, accounting for 48% of the errors, was in executing the four major solution operators in the context of the band target problem. The remaining 25% of the errors were due to adaptations required because of the nonisomorphism between the garden and band problems. For example, most of these latter errors (21%) were due to the fact (mentioned earlier) that the band solution is based on the sixth multiple of the LCM, whereas the garden solution is based on only the second multiple. Thus, the compute-multiples operator must be adapted to apply a greater number of times (the solution provided with the source problem showed four applications of that operator). Failure to adapt this operator was very clear in subjects' written protocols, as 88% of the subjects who failed to generate enough multiples for the band problem generated either three or four multiples. In sum, Novick and Holyoak's results support the conclusion that analogical mapping and pattern completion are quite robust, but that adaptation is a far less reliable component of analogical transfer.

Although adaptations seem more difficult to articulate for nonmathematical domains, the same principles should apply. For the tumor problem, transfer after a hint is attenuated when the source and target are nonisomorphic in their goals and/or solution constraints (although, as indicated earlier, other deviations from isomorphism seem not to impair analogy use). Consider the parade story used by Gick and Holyoak (1980, Experiment 2). Like the attack-fortress story described earlier, this story had a fortress with roads leading out from it in all directions. The goal was for a general to stage an impressive parade of soldiers that could be seen throughout the country. Thus, whereas the goal in the attack-fortress story and the tumor problem was to concentrate a large force at a particular central location, the goal in the parade story was to distribute a large force over a wide area. The solution constraint in the parade story was that if the parade was not impressive enough, the general would be demoted to private. In the attack-fortress story and the radiation problem, the constraint was that sending a large force down a single route would cause unwanted destruction. Adaptation rather than pattern completion is required to determine that the differing goals and solution constraints nevertheless imply analogous solution procedures. In line with this reasoning, the transfer rates after a hint for the attack fortress and parade groups were 76% and 49%, respectively. Holyoak and Koh (1987) found a similar result for versions of the lightbulb story that manipulated the similarity of the solution constraint to that in the tumor problem (also see Keane, 1988).

Finally, we consider a study by Reed, Ernst, and Banerji (1974) in which subjects solved the missionaries and cannibals (MC) and jealous husbands (JH) problems, with the order counterbalanced across subjects. In each problem, three people of each of two types must cross a river using a boat that holds two

people. However, the solution constraints for the two problems differ. For MC, the cannibals cannot outnumber the missionaries on either side of the river. For JH, a wife cannot be in the presence of any man unless her husband is also present. The solution to the latter problem is more constrained because husbands and wives are paired, whereas the various missionaries or cannibals are interchangeable. When subjects received the second problem, they were told mapping between the problems (*husbands* = *missionaries*; *wives* = *cannibals*). Nevertheless, solution of the second problem was facilitated only for the direction JH to MC. In this order, the implications of the pairing constraint for determining legal moves can be imported to MC or ignored. In either case, solution of MC should be facilitated. Transfer from MC to JH, however, requires adapting the familiar solution constraint and, therefore, the solution procedure to account for the pairing of husbands and wives. For a similar reason, Gholson, Eymard, Long, Morgan, and Leeming (1988) found better transfer from the "fox/goose/corn" problem to MC than vice versa.

2.2.3. Are Mapping and Pattern Completion ever Difficult? A review of the literature on analogical transfer reveals only one type of situation for which mapping and/or pattern completion appears to be difficult: when novices must overcome misleading surface similarities between the source and target, as in the "cross-mapping" manipulation used by Ross (1987, 1989) and by Gentner and Toupin (1986). Ross used mathematical problems involving elementary probability theory. In the similar-role condition, the source and target involved similar objects that played corresponding roles in the two problems (e.g., both problems involved assigning computers to offices). In the different-role condition, the source and target again involved similar objects, but those objects played different roles in the two problems (e.g., the source involved assigning offices to computers). The correct formula was provided with each target problem, so subjects simply had to instantiate the formula correctly. Nevertheless, instantiation (i.e., substitution) accuracy was greater in the similar-role than the different-role condition (approximately 67% vs. 40%, respectively). Gentner and Toupin (1986) found similar detrimental effects of cross-mapping for children reenacting simple stories with new characters.

Although these results suggest that mapping is difficult when similar objects play different roles in the source and target, the difficulty may interact with expertise. Misleading surface features may be particularly problematic for novices, such as Ross's subjects, who are unsure of the important structural features in the domain and therefore highly weight similarity information (e.g., Chi, Feltovich, & Glaser, 1981; Schoenfeld & Herrmann, 1982; Silver, 1981). In an explicit comparison of different expertise levels, Novick (1988) found that when appropriate structural information is present in a transfer situation, novices have greater difficulty than experts in ignoring misleading similarity information. Gentner and Toupin (1986) found similar results for their comparison of different age groups: when relevant structural information was provided (their "systematic" condition), younger children (aged 4–6) but not older children (aged 8–10) were adversely affected by the cross-mapping manipulation.

Because pattern completion has not been distinguished from mapping and adaptation in any empirical studies of transfer, the factors that determine its degree of difficulty are not known. We would venture two predictions. As is the case for mapping, one important factor may be the solver's level of expertise in the source and target domains. In addition, because it should be easier to create new target propositions when all of the appropriate object and predicate mappings are known (i.e., when inferences require only CWS) than when some of the analogous objects and/or predicates must be generated in the target domain (i.e., when inferences require CWSG), we would expect the success of pattern completion to be dependent on the success of the mapping process. Using a mapping task and algebra word problems similar to those used by Reed (1987), Novick (1992) found support for both of these predictions concerning the substitution component of pattern completion.

We now attempt to show how these theoretically and empirically motivated distinctions among component processes in analogical transfer can be modeled within a hybrid symbolic-connectionist system. Our starting point is the ACME model of mapping proposed by Holyoak and Thagard (1989).

3. A SYMBOLIC-CONNECTIONIST MODEL OF ANALOGICAL MAPPING AND PATTERN COMPLETION

We now describe the ACME model of mapping as it has been extended with CWS and CWSG procedures for analogical pattern completion, and report several computational tests and simulations of some of the relevant empirical results described above. We begin with a computational experiment that tests the robustness of the system, and at the same time provides transfer results that can be compared to those obtained using a standard connectionist learning algorithm, back-propagation (Rumelhart, Hinton, & Williams, 1986).

Although several models of analogical transfer that include the CWS principle have been proposed, no systematic tests have been reported of the robustness of such systems given impoverished analogs as inputs. Indeed, no systematic tests of robustness have been reported for models of either mapping or postmapping transfer processes. Models of mappings have typically been applied to representations that are nearly isomorphic; it is unclear how well the systems could map less orderly representations. In natural settings, use of analogy typically involves situations in which at least one of the analogs—the novel target—is imperfectly understood. A model of human analogical transfer must be sufficiently robust as to be able to identify systematic correspondences between analogs despite gaps in the initial representations, and then proceed to generate plausible inferences to fill those gaps.

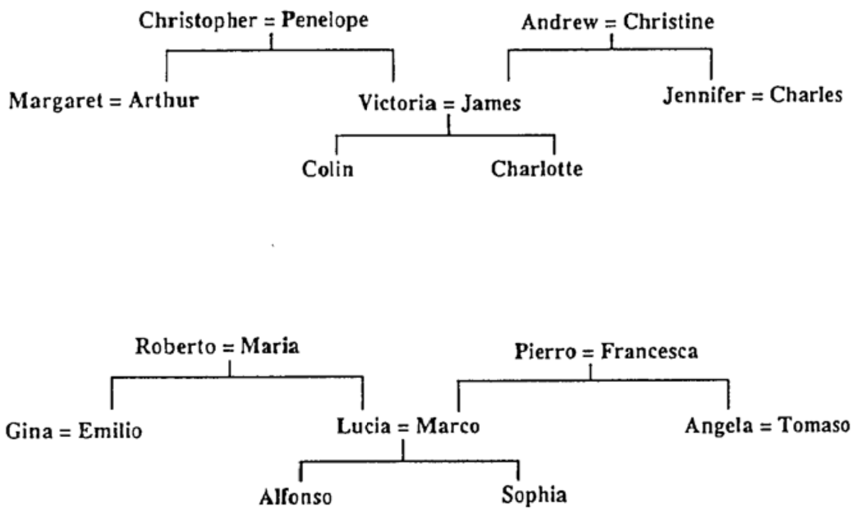
In this section we report tests (described by Melz & Holyoak, 1991) of the robustness of the ACME with CWS transfer system. We performed a number of computational experiments in which we randomly deleted information from one or both of two originally isomorphic analogs, and observed the degree to which the system could reconstruct the damaged analogs. We compare our transfer results with those produced by a back-propagation learning system proposed by Hinton (1986). We first present the example that we used to test the robustness of our transfer mechanism, and describe the system for which it was originally constructed, Hinton's (1986) back-propagation model of relational learning. Then we describe ACME and its CWS pattern-completion algorithm, which uses the output of the mapping process to transfer knowledge between two analogs. Finally, we present a series of computational experiments designed to explore the robustness of the proposed transfer mechanism.

3.1. Learning Family Trees with Back-Propagation

Hinton (1986) described a back-propagation network that he trained on propositional representations of family trees. The primary purpose of Hinton's study was to determine whether the hidden units of the network could develop intuitively meaningful representations of abstract features of a corpus of propositions. The basic family trees that the network learned, and which were also used in the present study, are depicted in Figure 2.1. As is visually apparent, these English and Italian families have an isomorphic structure (e.g., Christopher enters into the same pattern of kinship relations as does Roberto). Using 12 common relational terms (father, mother, husband, wife, son, daughter, brother, sister, uncle, aunt, nephew, and niece), each family can be described by a set of 56 propositions about relationships among the 12 individuals. In Hinton's project, propositions of the form (person1 relation person2) (e.g., Emilio has_father Roberto)) were translated into a connectionist representation and presented to the network. The input layer consisted of 24 units representing localist encodings of each of the people who could fill the person1 role (12 Italians and 12 English), and 12 localist encodings of the relations (e.g., has_mother, has_uncle, etc.). The 24 person1 units were connected to a 6-unit hidden layer, ad the 12 relation units were connected to a separate 6-unit hidden layer. These two discrete hidden layers were both connected to a further 12-unit hidden layer, which in turn projected to another 6-unit hidden layer. This final hidden layer projected to an output layer representing the 24 possible fillers of the person2 role.

The network was trained by clamping the appropriate person1 and relation units in the input layer, and person2 unit(s) in the output layer, and adjusting the weights using back-propagation. Training was based on approximately 96% of the possible propositions, with 1,500 sweeps through the training set. After training, clamping person1/relation pairs in the input layer could accurately activate the correct person2 unit(s) in the output layer (including multiple fillers

Figure 2.1.
Isomorphic family trees. (From "Learning Distributed Representations of Concepts," by G.E. Hinton, 1986, *Proceedings of the Eighth International Conference of the Cognitive Science Society,* Hillsdale, NJ: Erlbaum. Reprinted by permission.) Note that "=" signifies the relation "is married to."



of the person2 role, as in the case of a person with two aunts). For example, if the input units for "Emilio" and "has_father" were clamped, the output unit "Roberto" would be turned on. The hidden units in the network were able to abstract useful features, such as nationality and generation, from the training set. When tested on its ability to complete the 4% of the propositions that had not been used in training, the network was correct on 100% of these in one run and 75% in a second run.

3.2. Analogical Mapping and Pattern Completion in ACME

3.2.1. ***Mapping.*** Holyoak and Thagard (1989) framed the problem of analogical mapping in terms of parallel satisfaction of multiple constraints that jointly determine the optimal correspondences between elements of the source and target analogs. Their ACME model receives as input a source analog and a target analog represented in predicate-calculus notation. Each analog consists of a set of propositions, where each proposition consists of an *n*-place predicate, a list of constituent objects of the predicate, and a proposition label. For example, the fact that Emilio's father is Roberto would be represented by the proposition:

(has_father(Emilio Roberto)I1).

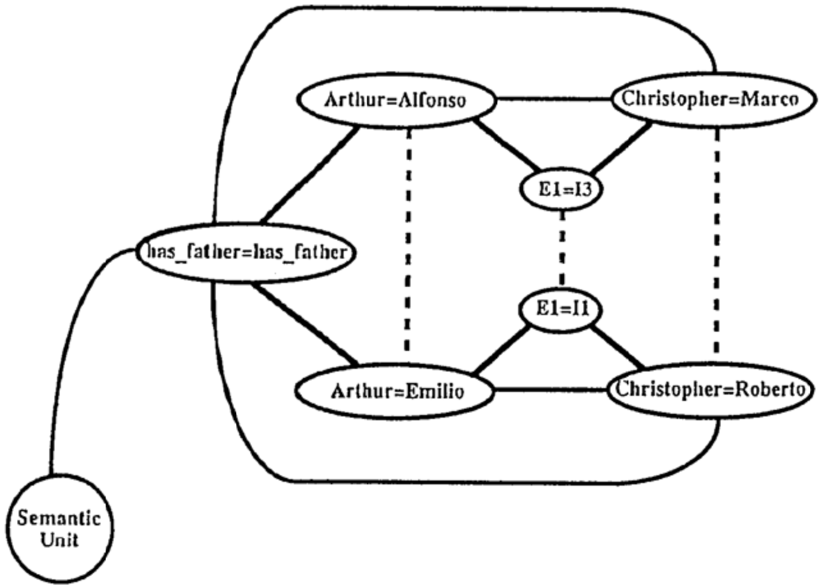
The full representations of each family would consist of 56 propositions, each involving a two-place relation.

The central constraint on analogical mapping embodied in the ACME model is the pressure toward *isomorphism.* Isomorphism requires that: (a) the elements of the target and source analogs map one-to-one, and (b) if a proposition in the target maps to a proposition in the source, the constituent predicates and objects of the target proposition must map onto the corresponding elements of the source. To enforce the isomorphism constraint, ACME constructs a network in which nodes represent elemental mapping hypotheses and weighted links between the nodes represent constraints between mapping hypotheses. Mapping hypotheses are created for correspondences of elements of the same logical type (i.e., propositions, *n*-place predicates, and objects). If the analogs can be divided into major constituent units, or "fields" (e.g., a problem representation might be divided into propositions describing the initial state, the goals, and the solution), then the above restrictions on unit formation are applied separately for each field type. (For the family-tree problem, each analog is treated as a single undifferentiated field.) Symmetric excitatory links are created between mapping hypotheses that are consistent with each other, and symmetric inhibitory links are created between hypotheses that are inconsistent. For each potential proposition mapping, an excitatory cluster of units is formed by creating excitatory links between the proposition mappings and the corresponding mappings of the propositions' predicates and arguments. Inhibitory links are placed between alternative mappings of the same element, in order to enforce the pressure for one-to-one mappings.

Figure 2.2 depicts a portion of the network ACME creates when presented with the representations of the English and Italian families as source and target analogs, respectively. Two excitatory clusters based on the proposition mappings *E1*=*I1* and *E1*=*I3* are shown, where *E1* is "Arthur has_father Christopher," *I1* is "Emilio has_father Roberto," and *I3* is "Alfonso has_father Marco." These proposition mappings inhibit each other because they represent competing hypotheses about the mapping of *E1* (similarly, the mappings *E1*=*I1* and *E2*=*I1* are mutually inhibitory). In addition, the object mappings associated with the conflicting proposition mappings are inconsistent, so they too inhibit each other. (Excitatory and inhibitory links are represented by solid and dashed lines, respectively, in the figure.)

In addition to isomorphism, ACME postulates that the degree to which predicates in the two analogs are semantically similar is an important factor affecting mapping. ACME implements the constraint of *semantic similarity* by connecting any mapping unit relating two semantically similar predicates to a special "semantic unit" that has its activation clamped to the maximum value. Units relating identical predicates are automatically connected to the semantic unit with relatively high values of excitation. In the experiments reported in this paper, only mappings of identical predicates have connections to the semantic

Figure 2.2.
Partial mapping network, mapping the English family to the Italian family. Note that "=" signifies the relation "maps to."



unit. In Figure 2.2, for example, a connection exists between the semantic unit and the predicate mapping *has_father = has_father*.

A third mapping constraint incorporated in ACME concerns the degree to which elements or element correspondences are considered to be pragmatically relevant to the analogist's goals. If an element is deemed to be especially important, all mapping hypotheses involving that element are linked to a clamped "pragmatic unit." In addition, any correspondence (e.g., *EI = II*) that is assumed to be known in advance of the mapping process may be linked to the pragmatic unit.

Each of the nodes in the network has an activation value, which is allowed to continuously vary between a minimum and a maximum value (e.g., −.3 and 1). To run the network, the semantic and pragmatic units are clamped to an activation of one. All other units are initialized at some minimal activation (e.g., .01), which allows the settling process to begin even for analogies that lack semantic and pragmatic links. The network is then relaxed using the activation updating rule 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 $min = -.3$ and $max = 1$. The value of $enet_j$ is equal to $\Sigma w_{ij} o_i(t)$ 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 $o_i(t)$ = maximum($a_i(t)$, 0). Activation updates are synchronous, and the updating algorithm is currently implemented in *LISP on a CM2 Connection Machine.

3.2.2. Pattern Completion. Once a set of mappings for objects, predicates, and propositions has been obtained by relaxing the system, we invoke the simple CWS pattern-completion procedure described earlier to generate candidate inferences based on the mappings and the structure of the analogs. Pattern completion is accomplished by an explicit symbolic algorithm that operates on the output of the relaxation algorithm. The criteria ACME uses to generate a proposition based on an unmapped proposition P is that the best mapping of P must have an activation below some threshold value, and the predicate and objects of P must have activations above the threshold. For all the simulations reported in this chapter, this threshold was chosen to be .2. Note that the above procedure for inference generation is inherently symmetrical; the new proposition can be added to either the source or the target analog. We investigated whether transfer performance is in fact symmetrical when the input representations are degraded.

3.3. Transfer Tests of ACME with CWS

The family-tree problem has several virtues as the basis for computational tests of an analogical transfer model. First, the two complete family structures are in fact isomorphic, so analogical mapping and pattern completion should be possible. Second, the full representations are a well-specified set of propositions, so we can quantify the degree to which analogs have been corrupted by eliminating propositions from the input representations. By deleting propositions from the inputs, we can systematically reduce the degree to which the input representations (as distinct from the underlying family structures) are isomorphic, and examine the robustness of the mapping and pattern completion mechanisms. Third, because Hinton (1986) investigated generalization by back-propagation using essentially the same problem, we can make a rough comparison of the degree to which missing information can be restored by analogical pattern completion to the success of generalization after learning by back-propagation.

3.3.1. Mapping the Intact Family Trees. The first requirement was to demonstrate that ACME could in fact map the two analogs if the complete representations (i.e., 56 propositions for each family) were provided as inputs. This is a nontrivial computational problem simply because the mapping network formed is very large (3,424 mapping units interconnected by 381,224 symmetrical links), due to the fact that all propositions involve two-place relations (so that any proposition in one analog could potentially map to any proposition in the other). The major parameter values used were .005 for decay, excitation, and similarity of identical predicates, and −.16 for inhibition. The network settled into a stable asymptotic state after 196 cycles of activation updating, producing a complete and correct set of correspondences between elements of the two structures. Table 2.4 presents the winning mapping units for predicates and people. As shown in the table, all the correct mappings had asymptotic activations close to the maximum possible value of one.

*3.3.2. **Reconstruction of Damaged Analogs With Identical Predicates.*** Having established that ACME can map the intact analogs, we next performed a series of computational experiments in which we randomly deleted

Table 2.4.
ACME Solution for Family-Tree Problem With Intact Analogs

English Family	Italian Family
Corresponding Predicates, with Asymptotic Activations of Winning Mapping Units	
HAS_FATHER	HAS_FATHER (0.95)
HAS_MOTHER	HAS_MOTHER (0.95)
HAS_HUSBAND	HAS_HUSBAND (0.92)
HAS_WIFE	HAS_WIFE (0.92)
HAS_SON	HAS_SON (0.95)
HAS_DAUGHTER	HAS_DAUGHTER (0.95)
HAS_BROTHER	HAS_BROTHER (0.89)
HAS_SISTER	HAS_SISTER (0.89)
HAS_UNCLE	HAS_UNCLE (0.94)
HAS_AUNT	HAS_AUNT (0.94)
HAS_NEPHEW	HAS_NEPHEW (0.95)
HAS_NIECE	HAS_NIECE (0.95)
Corresponding Persons, with Asymptotic Activations of Winning Mapping Units	
CHARLES	TOMASO (0.94)
MARGARET	GINA (0.94)
CHRISTINE	FRANCESCA (0.94)
PENELOPE	MARIA (0.94)
JENNIFER	ANGELA (0.96)
ANDREW	PIERRO (0.94)
CHARLOTTE	SOPHIA (0.92)
COLIN	ALFONSO (0.92)
JAMES	MARCO (0.96)
VICTORIA	LUCIA (0.96)
ARTHUR	EMILIO (0.96)
CHRISTOPHER	ROBERTO (0.94)

propositions from the family-tree representations provided as inputs to ACME. Deletion of a proposition implies that no mapping units are formed for it. We then used ACME to map the damaged representations, after which we applied the CWS pattern-completion mechanism to attempt to reconstruct the complete analogs. The results of our first series of experiments on analog reconstruction are presented in Figure 2.3.

In our first experiment, we randomly deleted propositions from both analogs, and observed the proportion of deleted propositions that were correctly or incorrectly created by the transfer mechanism previously described. The results of this experiment are shown in Figure 2.3A. The abscissa represents the proportion of propositions that were deleted from the entire set of propositions in the two analogs. The ordinate represents the proportion of the deleted propositions that were either created correctly (labeled "correct") or incorrectly (labeled "commission error"). Each data point on this and subsequent graphs represents the average of the results of two runs. ACME was able to reconstruct 100% of the missing propositions when 4% had been deleted, and 53% of those missing when 10% had been deleted, without making any commission errors. Correct restorations diminished to 23% when the deletion rate was increased to 20–30%, and at higher levels of deletion correct inferences were essentially eliminated. Commission errors were very infrequent even at the highest levels of damage to the analogs.

The above experiment involved symmetrical damage to the two analogs, with bidirectional transfer between the two analogs. In contrast, naturalistic analogical transfer typically involves asymmetric transfer from a well-understood source to a poorly understood target. To more closely approximate the naturalistic asymmetry of analogical transfer, we ran a second experiment in which we restricted proposition deletions to only a single analog (in this case, it happened to be the Italian family). As the results in Figure 2.3B clearly indicate, transfer was far better than in the previous experiment. Full recovery of deleted propositions was possible at deletion rates of up to 50%, and even at a deletion rate of 70% ACME was able to recover 43% of the missing propositions.

The difference in robustness between the two deletion procedures is extreme indeed. For example, deleting 60% of the propositions in one analog produces the same quantity of missing information as does deleting 30% of the propositions across both analogs. Yet the former procedure yields almost perfect recovery (Figure 2.3B), whereas the latter procedure allows recovery of almost none of the missing information (Figure 2.3A). We next explored potential structural explanations for this difference in robustness as a function of whether deletions were made from one or two analogs.

One possible explanation for the greater robustness of pattern completion after deletion from a single analog is that when random deletions occur across both analogs, then corresponding propositions (e.g., *E1* and *I1*) may both be deleted, in which case the CWS mechanism is guaranteed to fail (because there

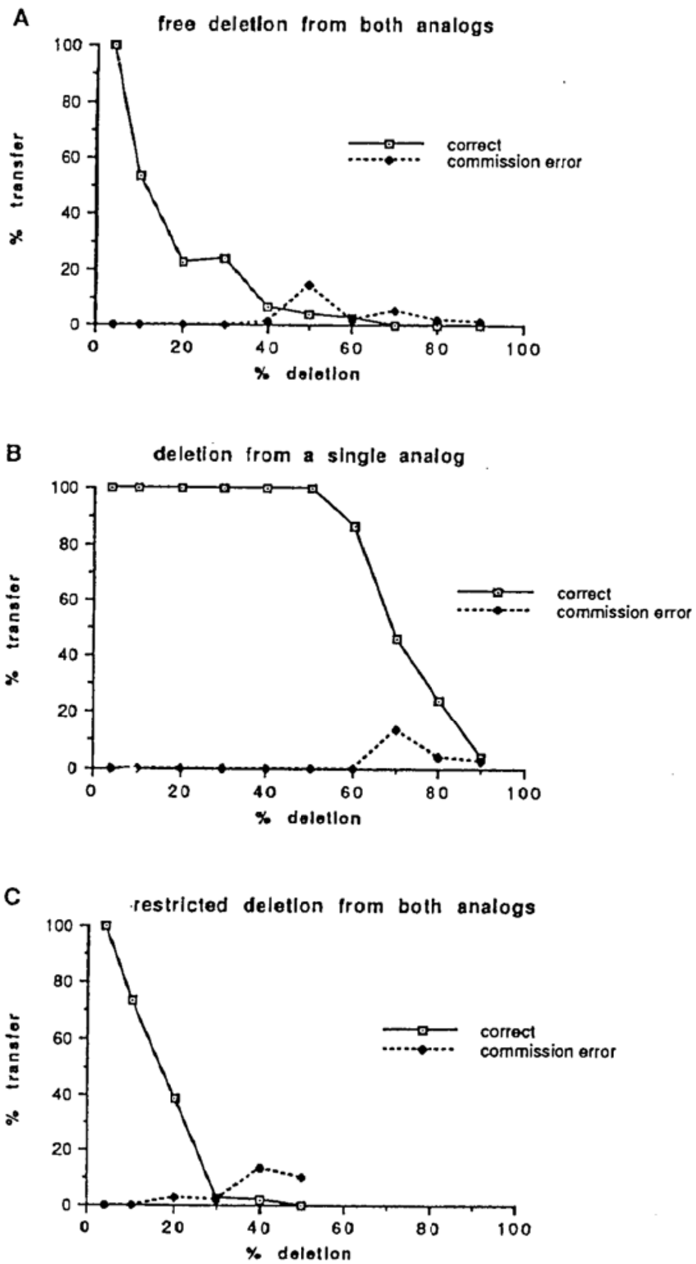


Figure 2.3.
Proposition reconstruction for three computational experiments.

will be no proposition from which to generate an analogous inference). (Of the deleted propositions, the proportion of propositions for which we can expect such an event to occur is simply p, the proportion of the total propositions that are deleted.) In contrast, if one analog is left intact, it is guaranteed that one member of each proposition pair is available (namely, the proposition in the source). To test the effect of this structural advantage for the latter procedure, we introduced a third deletion scheme that allowed propositions to be deleted from both analogs, with the restriction that at most one proposition from each pair of corresponding propositions could be deleted. The results, shown in Figure 2.3C, indicate that although this procedure produces somewhat more robust transfer than does free deletion from both analogs (Figure 2.3A), it remains much worse than when deletions are performed from only a single analog (Figure 2.3B). These results suggest that some other structural factor must account for the greater robustness of transfer when deletions are restricted to a single analog.

Another structural factor that varies when deletions are made from one versus two analogs involves the possible generation of incorrect proposition mappings. In the case where deletion is restricted to a single structure, for any proposition such as *I5* in the target structure that is deleted, the strongest (incorrect) mapping of its corresponding source proposition (*E5*), for example, *E5* = *I10*, will be inhibited by the correct mapping unit, here *E10* = *I10*. Because *E10* = *I10* represents the mapping of corresponding propositions, it will have a high activation and will drive the incorrect mapping *E5* = *I10* well below the threshold required for generating transfer candidates. Hence, Proposition *E5* in the source will be left unmapped, making it a candidate for generation of an inference about the target by CWS.

The single-structure-deletion case may be contrasted with the case in which proposition deletions occur in both analogs. In this case, incorrect proposition mappings do not necessarily experience the devastating inhibition described above. For example, consider the case in which Proposition *E10* is deleted from the source and *I5* is deleted from the target. If *I10* and *E5* share a common relation or object, they may produce a reasonably strong mapping. Because neither *E10* = *I10* nor *E5* = *I5* exists, the mapping *E5* = *I10* faces no serious competition, and thus is able to produce an activation level above the transfer threshold, preventing subsequent transfer from *E5*.

It follows that more errors in proposition mappings can be expected when deletions are made from two analogs rather than one. An indeed, this prediction was confirmed in our ACME simulations. When deletions were made from both analogs (in both the unrestricted and restricted deletion experiments), errors in proposition mappings began to appear at the 10% deletion level (an average of three errors), rising quickly to an average of more than 25 such errors at the 40% and 50% deletion levels. In contrast, when deletions were made from only one analog, proposition mismappings did not occur at all until the 70% deletion level, which produced an average of just four such errors. Thus, the greatly increased frequency of proposition mismappings that results when deletions are made in both analogs appears to be the main reason for the reduced robustness of pattern completion in that condition.

3.3.3. Analogical Pattern Completion Compared to Generalization by Back-Propagation. These computational experiments with the family-tree problem reveal that ACME with a CWS pattern-completion mechanism was able to produce accurate and robust transfer when deletions were restricted to a single analog. If as many as 50% of the propositions in the target analog were deleted, the system was able to recreate all of the missing information without error; significant recovery was obtained even if as many as 70% of the target propositions were deleted. Transfer was much more impaired when deletions were made from both analogs, rather than just the target. Melz and Holyoak (1991) reported an additional set of experiments in which the semantic pressure supporting mappings of identical elements is removed. The basic pattern of results for the experiments reported here was found, with only a mild reduction in the robustness of transfer. Hence, transfer with CWS appears to depend primarily on the structural configuration of the analogs, rather than any superficial similarity between them. These computational results lead to the prediction that transferring knowledge from a well-understood source to a poorly understood target will be easier than transferring knowledge between two moderately understood analogs, because mismappings are more likely to occur in the latter case. As far as we know, this prediction has not yet been directly tested for human analogical transfer.

It is of interest to compare the performance of ACME with Hinton's (1986) model of learning by back-propagation as applied to the family-tree problem. Our system appears better able to recover implicit missing information for the family tree problem. Moreover, its superior performance is based on only a single presentation of the problem, rather than on 1,500 training trials. It is important to note, however, that obvious and significant differences exist between the two systems. On one hand, ACME requires that the propositions about the two families be explicitly separated into target and source analogs, whereas Hinton's system received all propositions intermixed, and in fact *learned* that the distinction between English and Italian people was an important regularity. On the other hand, Hinton's generalization task involved giving the system the first argument and relation and asking it to generate the second argument, whereas ACME was asked to generate entire new propositions without any explicit partial cues. In addition to these differences in the transfer task performed by the two systems, the general aims of each system are quite different. Hinton's system is primarily intended to abstract general features from a body of propositions. ACME, on the other hand, has no such generalization capability, but rather conjectures the existence of unstated information based solely on structural correspondences between two sets of propositions. It is possible that the two approaches to transfer can be integrated so that they usefully complement each other.

4. ANALOGICAL TRANSFER IN PROBLEM SOLVING

4.1. The Need for Element Generation and Pragmatic Constraints

The ACME-plus-CWS model tested on the family-tree problem is not, by itself, adequate to model analogical problem solving. One basic deficit, noted earlier, hinges on the fact that transfer of a source solution to an unsolved target problem typically requires not only substitution of mapped elements, but also the generation of additional target objects and predicates to fill corresponding roles to previously unmapped objects and predicates in the source solution. For example, if an operator is introduced in the solution to the source, a corresponding operator must be generated in the target representation in order to construct an analogous solution.

To address this requirement, we further augmented the pattern completion mechanism in ACME by allowing the model to generate new elements. As we noted earlier, element generation "straddles the line" between what we are terming "pattern completion" versus "adaptation." In our usage, pattern completion can generate an abstract description of an element to be added to the target representation; however, identifying an actual object or predicate in the target domain that meets this description will require additional analysis of the target domain itself, and hence would be classified as adaptation. For our present purposes, we wished to model the pattern-completion aspect of generation, rather than provide a full account of subsequent adaptation as well. Accordingly, our "copy with substitution and generation" (CWSG) algorithm simply postulates target elements. For objects, nothing more is done: the semantic properties of a generated target object are simply implicit in the predicates that apply to it. For predicates, we made the simplifying assumption that as a default, a generated predicate in the target will be assumed to be identical to the corresponding source predicate. Thus, if a certain operator is used in the source, ACME with CWSG transfers the same operator into the target. Although this assumption is clearly oversimplified (as it is easy to construct cases where the required target predicate is a superordinate or coordinate of the source predicate; see Kokinov, this volume), the "identity default" is adequate for initial explorations. The algorithm can be applied to generate "higher order" propositions that take propositions as arguments; however, this feature is not necessary for the examples discussed here.

A crucial issue that must be addressed when an element-generation algorithm is used is specifying when it is allowed to apply. The ACME-with-CWS model discussed above tends to be conservative in generating transfer inferences because an unmapped source proposition is only used to generate a target inference if all elements of the source proposition have been mapped. But if unlimited CWSG is introduced, then *any* unmapped source proposition will generate a transfer inference, because if any of the elements of the source proposition are not already mapped to target elements, a correspondent in the target will simply be postulated. Such a loose criterion for inference generation will lead to rampant commission errors unless the source is in fact isomorphic to the target domain (or to a subset of the target domain), which is virtually never the case for complex realistic problem domains.

This issue raises the general problem of how pragmatic knowledge can be used to constrain inference generation. The approach we take here is based on the assumption that the problem solver knows something about *what parts* of the source are particularly relevant to the purpose for which the analogy is being used. As we noted earlier, ACME allows analogs to be divided into major constituents, or fields. For problems, the fields might include the initial state, the goals, and the solution. Typically, a problem solver would attempt to establish correspondences between the initial state and the goals of a source and target, and then use the resulting mapping to transfer the source solution to the target. It follows that at the pattern-completion stage, it is the *solution* field of the source that is crucial for generating inferences, to fill the previously empty solution field for the target. Accordingly, in the tests we will report here, we restrict the application of CWSG to the solution field of the source analog.

4.2. Applications of ACME-plus-CWSG to Transfer Between Algebra Word Problems

As an initial test of ACME-plus-CWSG as a model of human analogical problem solving, we used the system to simulate the mapping and equation-transfer results obtained by Reed (1987), which we described earlier, using the work and mixture problems presented in Tables 2.1 and 2.2, respectively. The first step was to formalize the algebra problems in predicate-calculus representations of the sort that ACME takes as inputs. Appendix I, Parts A and B, present the representations constructed for the work and mixture problems, respectively. The specific details of the formalization are rather arbitrary, but we enforced a number of consistent conventions. All variables are given names with the suffix "*"; names of values of numerical variables are additionally distinguished by the prefix "val." Specific numerical values have the prefix "num," and other objects have the prefix "obj." The relation between specific values and their corresponding variables is represented by the special predicate "instantiates." We assume that all predicates stated for variables are inherited by their instantiations (e.g., in the representation of the "pipe" problem, because "num8" is an instantiation of the variable "val_hrs*," and proposition P6 states that this variable is a "time," it follows that num8 is also a time.) Although this assumption allows us to reduce redundancy in the representations, it does not directly affect the operation of ACME, which treats "instantiates" the same as any other predicate for the purpose of mapping.

Mathematical functions such as "quotient-of" are expressed as three-place predicates, in which the first two arguments represent inputs and the third

represents the output. Operators are given the suffix "!"; (e.g., "multiply!" represents an instruction to perform a multiplication). Other details concerning the interpretation of the representations are provided as comments (lines that begin with ";"). Note that the "solution" field is empty in the representations of target problems.

Table 2.5 summarizes the size of the ACME networks formed for the four mapping simulations based on the work and mixture problems, and the number of cycles required for the network to settle in each run. All parameter values in these runs were identical to those employed in the simulations of the family-tree problem described above. Note that cycles to settle remains constant over a roughly two-fold increase in the network size from the work to the mixture problems, indicating that mapping difficulty is largely independent of the size of the representations. Settling time was somewhat longer for the mapping between "similar" mixture problems, which as we will see pose the most serious departure from isomorphism.

Tables 2.6–2.9 present the basic mapping and transfer solutions generated by ACME for each of the four runs. The winning mappings units and their asymptotic activations are given for all predicates, operators, and objects. Note that in each run all the operators and several numerical objects introduced in the source solution (representing computed quantities) are left unmapped (i.e., mapped to "NIL"), simply because the initial representations of the target analogs lack solutions. With these exceptions, the winning correspondences in the first three runs (isomorphic work, similar work, and isomorphic mixture problems; see Tables 2.6 and 2.8) are entirely correct.

Tables 2.7 and 2.9 present the solutions generated by ACME's CWSG pattern-completion algorithm, using the mapping results and the structure of the source analog. The left column gives the source proposition from which each target proposition was constructed (see Appendix I for details on the source propositions), and the middle and right columns give the corresponding constructed inferences for the isomorphic and similar targets, respectively. (The inferences are not given proposition labels.) Recall that the basic pattern completion

Table 2.5.
Network Size and Settling Time for ACME Solutions to Reed's (1987) Algebra Analogs

	Work Problems		Mixture Problems	
	Isomorphic (Pipe--> Typing)	Similar (Pipe--> Tank)	Isomorphic (Nurse--> Grocer)	Similar (Nurse--> Alloy)
units	421	553	959	904
links	4691	7091	18521	16485
cycles to settle	175	176	185	232

Table 2.6.
Results of ACME Mapping For Work Problems Used by Reed (1987)

Source (Pipe)	Isomorphic Target (Typing)	Similar Target (Tank)
Predicate Mappings with Asymptotic Activations		
PIPE	PERSON (0.73)	PIPE (0.49)
TANK	MANUSCRIPT (0.58)	TANK (0.71)
COMBINE	COMBINE (0.81)	COMBINE (0.81)
FILL-TIME	TYPE-TIME (0.95)	FILL-TIME (0.95)
TIME	TIME (0.70)	TIME (0.71)
FILL-RATE	TYPE-RATE (0.77)	FILL-RATE (0.81)
RATE	RATE (0.69)	RATE (0.70)
QUOTIENT-OF	QUOTIENT-OF (0.81)	QUOTIENT-OF (0.83)
NUMBER	NUMBER (0.87)	NUMBER (0.87)
ONE	ONE (0.69)	ONE (0.69)
INSTANTIATES	INSTANTIATES (0.96)	INSTANTIATES (0.96)
TWELVE	TEN (0.54)	TWENTY (0.54)
EIGHT	FIVE (0.58)	FIFTEEN (0.58)
KNOWN	KNOWN (0.70)	KNOWN (0.70)
PORTION-COMPLETED	NIL	NIL
PROPORTION	NIL	NIL
EQUAL	NIL	NIL
Operator Mappings		
DIVIDEI	NIL	NIL
MULTIPLYI	NIL	NIL
ADDI	NIL	NIL
Object Mappings with Asymptotic Activations		
NUM_H	NUM_H (0.89)	NUM_H (0.90)
•NUM8	NUM5 (0.97)	NUM15 (0.97)
•NUM12	NUM10 (0.89)	NUM20 (0.89)
NUM1	NUM1 (0.85)	NUM1 (0.87)
VAL_1/HRS*	VAL_1/HRS* (0.88)	VAL_1/HRS* (0.88)
OBJ_PIPE*	OBJ_PERSON* (0.95)	OBJ_PIPE*(0.95)
VAL_HRS*	VAL_HRS* (0.95)	VAL_HRS* (0.95)
OBJ_TWO-PIPES	OBJ_TWO-PEOPLE (0.90)	OBJ_TWO-PIPES (0.90)
OBJ_OIL_TANK	OBJ_MS (0.96)	OBJ_WATER-TANK (0.96)
OBJ_PIPE_LG	FLORENCE (0.91)	OBJ_PIPE_LG (0.90)
OBJ_PIPE_SM	ANN (0.91)	OBJ_PIPE_SM (0.90)
SUM_WHOLE_TASK	NIL	NIL
PRODUCT_PIPE_LG	NIL	NIL
PRODUCT_PIPE_SM	NIL	NIL
VAL_1/HRS_LG	NIL	NIL
VAL_1/HRS_SM	NIL	NIL

Note: Source elements preceded by the symbol "•" indicate those used by Reed (1987) in his mapping task.

algorithm is to: (a) substitute the appropriate target element if one is available from the mapping output; or if the source element was left unmapped, then (b) for predicates/operators, transfer the identical element from source to target; and (c) for objects, postulate a new target object with an arbitrary label.

The solutions generated for the two isomorphic mappings (middle columns of Tables 2.7 and 2.9) are entirely correct. Reed's (1987) subjects also were relatively successful in generating the correct equations for the isomorphic problems. (Human subjects, unlike ACME, were less successful on the mixture than the work isomorphs.) Note that the object-generation component of CWSG, because it introduces a single new target object to replace all occurrences of the unmapped source object, necessarily preserves all analogous variable bindings.

One caveat to the claim that ACME is fully successful in equation generation for the isomorphs should be noted. For the work problems (Appendix I.A), the representations include some information about the "start state" that we assume subjects are likely to infer, but which is not actually included in the problem statement. In particular, the representations introduce the concept of "type rate" for the isomorphic Typing problem (Proposition T7). Because this concept is present in the target representation, ACME is able to correctly map "fill rate" in the source (Pipe problem) to "type rate" in the target (see Table 2.6). The pattern-completion mechanism then simply substitutes the corresponding predicate "type rate" for "fill rate" in the source Proposition P31, along with the appropriate corresponding objects, to create an analogous target proposition (see Table 2.7). Suppose, however, that we did not introduce the concept "type rate" in the start state for the target problem. In that case ACME's CWSG algorithm would simply copy over the concept "fill rate" from the source to the target, which would of course result in an overgeneralization of the concept.

The fact that Reed's (1987) subjects performed quite well on the equation-transfer task for this pair of problems suggests either that: (a) they inferred the concept represented by Proposition T7 while encoding the target problem, or that (b) they were able to use semantic knowledge to appropriately modify "fill rate" when transferring that predicate to the target domain. Regardless of the appropriate explanation in this case, it is clear that a more complete implementation of CWSG will require greater semantic knowledge to guide search for the most appropriate predicate in the target domain to substitute for the generating predicate in the source (cf. Kokinov, this volume).

One aspect of the solution transferred for the work isomorphs is of particular interest because it provides a counterexample to the claim that analogical transfer can be appropriately constrained by transferring only relations from the source to target, and not one-place predicates or objects (Gentner, 1983; Falkenhainer et al., 1989). Note that from source Proposition P27, which calls for division of 1 by 12 to produce a quotient, ACME constructs a target proposition calling for division of 1 by 10 to produce a quotient. The numerator 1 ("num1") is transferred by virtue of the identity mapping, *num1* = *num1*. In this case a particular object—the number 1—is transferred from the source to the target. This example thus contradicts the proposal that analogical transfer always involves ignoring object identities and one-place predicates.

For the similar work analogs, the source problem involves two agents working together to complete a task, whereas the target problem also involves an additional negative contribution (use of water as the pipes fill the tank). Thus, although the correct mappings are obtained for all source elements, the target solution generated is inadequate because it fails to include any representation of the water usage (see Table 2.7). Modifying the equation to reflect this additional constraint would require adaptation, which the current ACME model, like most of Reed's (1987) subjects, is unable to perform.

Finally, Tables 2.8 and 2.9 present the results for the most difficult of the analogy problems that Reed tested, the similar mixture problems. In the source problem, the solution equation calculates the amount of acid in the mixture by multiplying the acid concentration (8%) by the pints of solution (4.5), a step stated as N48 in the representation of the Nurse problem in Appendix I.B. The target Alloy problem is actually simpler, in that the pounds of copper in the mixed alloy (the analog of the amount of acid in the mixed solution) is directly given (10.4). However, this simplification violates isomorphism, and leads both human subjects and ACME to both make mapping errors and to generate an erroneous equation for the target. The object "num8%" in the source is mapped to "num_10.4" in the target, even though the former is a percentage and the latter is a weight (and the predicate "eight-one-hundredths" is mapped to "ten-point-four"). This mismapping arises because each of the mapped elements lacks a good "mate": the target does not provide a percentage corresponding to num8%, and the source does not provide a weight corresponding to num_10.4. Because both are numbers related to the respective mixtures, in the absence of any better possibility they are similar enough to map to each other.

Because of this mismapping, ACME's pattern-completion mechanism produces an erroneous equation, via an inference derived from N48 in the source. To mimic the structure of the source equation, ACME simply substitutes the target object num_10.4 for the source object num_8%. The result is that instead of recognizing that num_10.4 directly represents the weight of copper in the mixed alloy, this value is instead multiplied by another weight, num60. This may seem like a ludicrous mathematical howler, but in fact it is the most common error that human subjects made for this analogy problem in an earlier study by Reed and his colleagues: "They continued to multiply two numbers together (10.4 and 60) even though the quantity of copper in the mixture was directly stated in the test problem" (Reed et al., 1985, p. 122). Thus, ACME's transfer mechanism seems to rather accurately model the performance of human subjects who are able to do analogical mapping and pattern completion, but are unable to evaluate or adapt the solution generated by pattern completion, because they do not fully understand the target domain.

Table 2.7.
Results of ACME Pattern Completion to Generate Equations for Target Work Problems Used by Reed (1987)

Source (Pipe) Prop.	Isomorphic Target (Typing)	Similar Target (Tank)
•P27	(DIVIDE! (NUM1 NUM10 obj202))	(DIVIDE! (NUM1 NUM20 obj375))
P28	(TYPE-RATE (ANN OBJ_MS obj202))	(FILL-RATE (OBJ_PIPE_SM OBJ_WATER-TANK obj375))
P29	(INSTANTIATES (obj202 VAL_1/HRS*))	(INSTANTIATES (obj375 VAL_1/HRS*))
•P30	(DIVIDE! (NUM1 NUM5 obj207))	(DIVIDE! (NUM1 NUM15 obj380))
P31	(TYPE-RATE (FLORENCE OBJ_MS obj207))	(FILL-RATE (OBJ_PIPE_LG OBJ_WATER-TANK obj380))
P32	(INSTANTIATES (obj207 VAL_1/HRS*))	(INSTANTIATES (obj380 VAL_1/HRS*))
P33	(PORTION-COMPLETED (ANN obj212))	(PORTION-COMPLETED (OBJ_PIPE_SM obj385))
•P34	(MULTIPLY! (obj202 NUM_H obj212))	(MULTIPLY! (obj375 NUM_H obj385))
P35	(PROPORTION (obj212))	(PROPORTION (obj385))
P36	(PROPORTION-COMPLETED (FLORENCE obj220))	(PORTION-COMPLETED (OBJ_PIPE_LG obj393))
•P37	(MULTIPLY! (obj207 NUM_H obj220))	(MULTIPLY! (obj380 NUM_H obj393))
P38	(PROPORTION (obj220))	(PROPORTION (obj393))
P39	(ADD! (obj212 obj220 obj230))	(ADD! (obj385 obj393 obj403))
P40	(EQUAL (obj230 NUM1))	(EQUAL (obj403 NUM1))
P41	(NUMBER (obj202))	(NUMBER (obj375))
P42	(NUMBER (obj207))	(NUMBER (obj380))
P43	(NUMBER (obj212))	(NUMBER (obj385))
P44	(NUMBER (obj220))	(NUMBER (obj393))
P45	(NUMBER (obj230))	(NUMBER (obj403))
P46	(PROPORTION (obj230))	(PROPORTION (obj403))

Note: Source elements preceded by the symbol "•" indicate those used by Reed (1987) in his mapping task.

Table 2.8.
Results of ACME Mapping For Mixture Problems Used by Reed (1987)

Source (Nurse)	Isomorphic Target (Grocer)	Similar Target (Alloy)
Predicate Mappings with Asymptotic Activations		
SOLUTION	NUTS (0.81)	ALLOY (0.81)
COMBINE	COMBINE (0.81)	COMBINE (0.81)
ACID-CONCENTRATION	UNIT-PRICE (0.90)	ALLOY-PURITY (0.88)
PERCENTAGE	PRICE (0.58)	PERCENTAGE (0.71)
ACID-CONTENT	TOTAL-PRICE (0.70)	COPPER-CONTENT (0.71)
VOLUME	WEIGHT (0.57)	WEIGHT (0.74)
AMOUNT	LBS (0.90)	LBS (0.90)
PRODUCT-OF	PRODUCT-OF (0.81)	PRODUCT-OF (0.81)
INSTANTIATES	INSTANTIATES (0.97)	INSTANTIATES (0.96)
NUMBER	NUMBER (0.90)	NUMBER (0.90)
SIX-ONE-HUNDREDTHS	ONE-POINT-SIX-FIVE (0.59)	TWENTY-ONE-HUNDREDTHS (0.60)
TWELVE-ONE-HUNDREDTHS	TWO-POINT-ONE-ZERO (0.57)	TWELVE-ONE-HUNDREDTHS (0.71)
EIGHT-ONE-HUNDREDTHS	ONE-POINT-EIGHT-THREE (0.56)	***TEN-POINT-FOUR (0.52)***
GREATER-THAN	GREATER-THAN (0.92)	LESS-THAN (0.81)
FOUR-POINT-FIVE	THIRTY (0.54)	SIXTY (0.57)
KNOWN	KNOWN (0.79)	KNOWN (0.79)
Operator Mappings		
SUBTRACT!	NIL	NIL
MULTIPLY!	NIL	NIL
ADD!	NIL	NIL
Object Mappings with Asymptotic Activations		
NUM_S	NUM_A (0.88)	NUM_B (0.88)
NUM_W	NUM_P (0.88)	NUM_G (0.88)
•NUM_4.5	NUM30 (0.88)	NUM60 (0.88)
•NUM8%	NUM_1.83 (0.92)	***NUM_10.4 (0.79)***
•NUM12%	NUM_2.10 (0.93)	NUM12% (0.91)
•NUM6%	NUM_1.65 (0.97)	NUM20% (0.97)
VAL_PINTS*	VAL_LBS* (0.94)	VAL_LBS* (0.94)
VAL_ACID*	VAL_$* (0.85)	VAL_CU* (0.87)
OBJ_SOLN*	OBJ_NUTS* (0.95)	OBJ_ALLOY* (0.95)
VAL_CONC*	VAL_$/LB* (0.94)	VAL_%* (0.93)
OBJ_SOLN_MIX	OBJ_MIX (0.92)	OBJ_ALLOY_MIX (0.91)
OBJ_SOLN_STRONG	OBJ_ALMONDS (0.92)	OBJ_ALLOY_BAD (0.92)
OBJ_SOLN_WEAK	OBJ_PEANUTS (0.92)	OBJ_ALLOY_GOOD (0.92)
VAL_ACID_M	NIL	NIL
VAL_ACID_S	NIL	NIL
VAL_ACID_W	NIL	NIL

Note: The two bold, italicized mappings for the alloy problem are incorrect. Source elements preceded by the symbol "•" indicate those used by Reed (1987) in his mapping task.

Table 2.9.
Results of ACME Pattern Completion to Generate Equations for Target Mixture Problems Used by Reed (1987)

Source (Nurse) Prop.	Isomorphic Target (Grocer)	Similar Target (Alloy)
•N42	(SUBTRACT! (NUM30 NUM_P NUM_A))	(SUBTRACT! (NUM60 NUM_G NUM_B))
N43	(TOTAL-PRICE (OBJ_PEANUTS obj156))	(COPPER-CONTENT (OBJ_ALLOY_GOOD obj156))
N44	(MULTIPLY! (NUM_1.65 NUM_P obj156))	(MULTIPLY! (NUM20% NUM_G obj156))
N45	(TOTAL-PRICE (OBJ_ALMONDS obj160))	(COPPER-CONTENT (OBJ_ALLOY_BAD obj160))
N46	(MULTIPLY! (NUM_2.10 NUM_A obj160))	(MULTIPLY! (NUM12% NUM_B obj160))
N47	(TOTAL-PRICE (OBJ_MIX obj164))	(COPPER-CONTENT (OBJ_ALLOY_MIX obj164))
•N48	(MULTIPLY! (NUM_1.83 NUM30 obj164))	***(MULTIPLY! (NUM_10.4 NUM60 obj164))***
N49	(ADD! (obj156 obj160 obj164))	(ADD! (obj156 obj160 obj164))
N50	(NUMBER (obj156))	(NUMBER (obj156))
N51	(NUMBER (obj160))	(NUMBER (obj160))
N52	(NUMBER (obj164))	(NUMBER (obj164))
N53	(INSTANTIATES (obj156 VAL_$*))	(INSTANTIATES (obj156 VAL_CU*))
N54	(INSTANTIATES (obj160 VAL_$*))	(INSTANTIATES (obj160 VAL_CU*))
N55	(INSTANTIATES (obj164 VAL_$*))	(INSTANTIATES (obj164 VAL_CU*))

Note: The bold, italicized proposition generated for the alloy problem is incorrect. Source elements preceded by the symbol "•" indicate those used by Reed (1987) in his mapping task.

Finally, let us consider the responses ACME would produce to the mapping questions that Reed (1987) administered to his subjects, the results of which were reported in Tables 2.1 and 2.2. We assume that for elements that can be mapped directly, ACME would report the winning mapping; and for elements that lack direct correspondents, ACME would report the target expression that would be generated during pattern completion. It should be clear from the previous discussion and the results shown in Tables 2.6–2.9 that ACME produces entirely correct responses except for the two mismappings for the similar mixture problems. This pattern of successes and failures is extremely similar to that produced by Reed's subjects, who were usually correct on all except these same two questions, on which they virtually always failed. Thus, ACME with its CWSG transfer mechanism is able to provide a fairly detailed and accurate simulation of human mapping and solution transfer for algebra word problems.

4.3. Simulating Mapping and Transfer for Nonisomorphic Arithmetic Word Problems

The performance of ACME with CWSG in simulating Reed's (1987) findings reveals some of the complexities that arise when we consider situations in which the underlying source and target domains are not completely isomorphic. The success of the CWS and CWSG procedures depends upon the model's tacit assumption that the source and target situations, despite any apparent "gaps" in their representations, are in fact isomorphic. But if the source includes propositions that lack parallels in the target situation, commission errors (i.e., erroneous inferences about the target) are likely to result. For example, for the similar mixture problems discussed in the previous section, CWSG generates the incorrect inference that solving the alloy problem involves multiplying 60 by 10.4. Conversely, if the target requires operators for which no correspondent exists in the source problem, omission errors are likely. For example, this was the case for the tank problem discussed in the previous section, in which CWSG failed to generate an operator to account for the water usage.

It should be clear, then, that successful transfer between nonisomorphic analogs will depend upon additional pragmatic information that: (a) either prevents erroneous inferences from being generated (by blocking the application of CWSG to source propositions identified as irrelevant to the target) or weeds them out after they are generated (by identifying the inferences as incorrect in the target situation), and (b) identifies the unique aspects of the target problem so as to create appropriate new operators. In our simulation of Reed's experiments, ACME's use of CWSG was restricted to the solution field, an example of a pregeneration constraint on inference generation. Posttransfer evaluation of inferences, as well as the creation of target-specific operators, necessarily involves adaptation, a capacity that ACME lacks.

We will now report several additional applications of the ACME system, augmented by its pattern-completion mechanisms, to simulate empirical data on human analogical transfer between mathematical problems. These data involve the garden (source) and band (target) problems presented in Table 2.3, which were used by Novick (1988) and Novick and Holyoak (1991) to investigate analogical problem solving. These problems are both more complex and less isomorphic than any of the examples we have yet discussed. Successful transfer with these problems requires posttransfer evaluation of inferences.

4.3.1. Representations of the Problems Used by Novick and Holyoak (1991). Appendix II, Parts A and B, respectively, present representations of the garden and band problems in the type of predicate calculus notation that serves as the input to ACME. We attempted to represent the central information that is either directly given in the problems (see Table 2.3) or readily inferable. As in the representations of Reed's (1987) problems, each representation has three major fields: a description of the initial problem state, of the goals, and of the solution (if given). Arguments in propositions are either names of objects (e.g., "obj_band" in the band problem denotes the band, "num1" denotes a certain number) or embedded propositions (e.g., Proposition B6 in Appendix II.B states that num1 is the remainder of the division operation denoted by B5). The solution procedure represented in Appendix II.A contains operators, such as "find-lcm!," which could be decomposed into more detailed steps. (As before, procedural operators are distinguished from declarative predicates by an exclamation point.)

These formalizations of the garden and band problems make it clear that the relationship between the two problems is quite complex. The problems are superficially dissimilar and far from completely isomorphic. In particular, the garden problem involves three initial attempts to divide the plants into kinds given a certain possible number of plants, denoted by "num_total_G0," followed by four additional attempts using a revised total, "num_total_G1," which finally results in a solution. In contrast, the band problem involves only a single total number of band members, "num_total_B," and only four attempted divisions; these should map onto the fourth through seventh attempts in the garden problem. The garden and band problems further differ in that the goal of the band problem includes a range restriction (the total must be at least 45 but less than 200) that has no counterpart in the garden problem. In addition, the compute-multiples operator must be applied six times in order to solve the band problem, compared to only twice for the garden problem. There are also several differences of a more incidental kind (e.g., the band problem involves two people, the director and Andrew; whereas the garden problem involves three people, Mr. and Mrs. Renshaw and their daughter). Of course, the overall cover stories involve almost completely different predicates, and the specific numbers involved differ (except for the number 5, which appears in both problems but plays a different role in each).

Except for the knowledge that both problems involve people giving orders or making suggestions, the major shared predicates which could serve as retrieval cues are such mathematical concepts as "divide," "remainder-of," and "number," which are common to numerous mathematical problems with quite diverse structures. The mapping is complex, and transfer of the LCM procedure to the band problem requires that the solution to the garden problem be adapted to account for: (a) the additional required applications of the compute-multiple operator, and (b) the range restriction. On the face of it, then, we would expect that retrieval, mapping, and adaptation could all contribute to the difficulty of using the analogy.

4.3.2. Simulation Results for Mapping. As we noted earlier, Novick and Holyoak (1991) found that their subjects were able to correctly map about 80% of the corresponding concepts and numbers in the garden and band problems. We performed a series of simulations of the mappings produced by ACME when given representations of the two problems. Three hint conditions were simulated, corresponding to conditions tested with human subjects by Novick and Holyoak. In the no-mapping-hint condition, the analogs were mapped using only the information contained in the problem representations (see Appendix II). In the remaining conditions, mapping hints were provided by using the feature of the program that gives extra pragmatic support to mapping units that are "presumed" in advance. The presumed mappings were selected to correspond as closely as possible to those given to subjects in Novick and Holyoak's experiments. In the concept-mapping hint condition, the following mapping units were presumed: the predicate mappings *band-members = plants*, *grouping-of = kind-of*, *number-per-group = number-per-kind*, and *number-left-out = number-extra-spaces*; and the proposition mappings *B63 = G74* and *B64 = G75*, which represent the parallel goals of dividing the total number of objects by a number that leaves zero remainder. In the number-mapping hint condition, the crucial numerical correspondences were presumed, as follows: *num12 = num10*, *num8 = num4*, *num3 = num5* (the mappings of divisors that leave nonzero remainders), *num5 = num6* (divisors that leave zero remainder), and *num1 = num2* (the nonzero remainders).

ACME applied its general constraints to the representations to build a network of over 1,600 mapping units representing possible correspondences, connected by over 31,000 excitatory and inhibitory links. The parameters used were identical to those used in the previous simulations. In addition, the mapping between the similar verbs "orders" and "suggests" was given a weight of .0025, half the value of the weight for identical predicates.

Each of the three runs of ACME found a stable set of mapping units representing the optimal correspondences between the band and garden problems after approximately 500 cycles of updating. In all three cases the correct mapping emerged as the clear victor for each of the critical concepts and numbers. Table 2.10 reports the asymptotic activation level of each of several

Table 2.10.
Selected Results of Mapping the Band Problem to the Garden Problem

Source	Target	Presumed Mappings None	Concepts	Numbers
NUM10	NUM12	0.91	0.91	0.98*
NUM4	NUM8	0.91	0.91	0.98*
NUM5	NUM3	0.91	0.91	0.98*
NUM2	NUM1	0.94	0.94	0.98*
NUM6	NUM5	0.91	0.91	0.98*
NUMBER-PER-KIND	NUMBER-PER-GROUP	0.90	0.98*	0.90
KIND-OF	GROUPING-OF	0.95	0.98*	0.95
NUMBER-EXTRA-SPACES	NUMBER-LEFT-OUT	0.77	0.97*	0.77
PLANTS	BAND-MEMBERS	0.60	0.97*	0.60
G74	B63	0.85	0.98*	0.85
G75	B64	0.72	0.97*	0.72

Note: * indicates a "presumed" mapping.

winning mapping units. Without any mappings presumed, each correct mapping unit achieved an activation that is both substantial in absolute magnitude (ranging from .60 to .95) and higher by at least .20 than its nearest competitor. The results for the two hint conditions reveal that units that are "presumed" achieve somewhat higher activation levels (about 98%), but that the rest of the mapping is essentially unchanged from the no-mapping-hint condition.

These simulations are broadly consistent with the mapping data presented by Novick and Holyoak (1991), in that ACME's mapping performance for the important concepts and numbers is accurate in all three conditions. As noted earlier, the empirical results indicated that the number-mapping hint was more effective than the concept-mapping hint in promoting successful transfer of the LCM solution procedure to the band problem. The ACME model implies that this difference is not due to the mapping stage (since for the simulation, mapping is accurate regardless of which hint, if any, is given). The pattern-completion algorithm, however, suggests that the number-mapping hint is more closely related to the requirements for postmapping inference: None of the concepts provided by the concept-mapping hint appear in the representation of the solution to the garden problem (see Appendix II.A); rather, only the numbers that must be manipulated to derive the solution are mentioned. Thus, in attempting to solve the band problem by analogy to the garden problem, successful pattern completion (in particular, substitution) depends on the availability of the correct number mappings (also see Novick & Holyoak, 1991, on this point). The concept mappings presumably play an important role in deriving the number mappings (because the two types of mappings are mutually supportive), but for pattern completion (and perhaps also adaptation) the latter are crucial.

An important issue is whether ACME's predictions for these materials are highly dependent on the particular parameter values selected. Accordingly, we simulated several variations of the no-mapping-hint condition to determine whether the correct mappings would still emerge. First, we sharply changed the relative balance of excitation and inhibition by reducing the inhibition parameter by half to −.08. In another run, we collapsed the parameters for semantic weights linking similar or identical predicates to a single value, .0025, which was half of the value for identical predicates used in the basic version. In both cases, all the crucial correspondences still emerged. In fact, the activations in the various runs rarely differed by more than .05, and the most extreme difference in activation was about .20.

4.3.3. Pattern Completion. Clearly, ACME, like people, is capable of producing a correct set of mappings between the crucial elements of the nonisomorphic band and garden analogs with relative ease. However, in the Novick and Holyoak studies, a majority of the subjects who obtained the correct mappings nevertheless failed to correctly solve the target problems, indicating that mapping alone is not sufficient for successful solution transfer. Subjects' failure to adequately transfer the source solution can be attributed to one or more of the following factors: (a) difficulty of performing pattern completion, (b) inaccuracy of pattern completion, or (c) difficulty and/or inaccuracy of adaptation. As we have previously discussed, we believe that inferences based on pattern completion ought to be relatively easy to compute, but that pattern completion will tend to produce inappropriate inferences when applied to nonisomorphic domains. To test the effectiveness of ACME on postmapping inference generation for the nonisomorphic band and garden analogs, we applied our pattern completion mechanisms in two ways. First, we performed an experiment analogous to the family-tree deletion experiments: we randomly deleted propositions from the source and the target, and observed the extent to which mapping followed by CWS could regenerate the deleted propositions. Second, we applied CWSG to the solution field of the target, as in our simulation of Reed's experiments. Both of these simulations point up limitations of the pattern completion mechanism, and demonstrate that additional (and presumably more complex) mechanisms may be necessary to effectively extend a mapping between nonisomorphic analogs.

To test the CWS mechanism, we randomly deleted propositions from the two analogs, in two separate sets of runs which replicated the first two deletion experiments reported for the family-tree problem. In the first set of runs, we randomly deleted propositions from both analogs. These simulations modeled a situation in which both the source and target are imperfectly understood. In the second set of runs, we restricted deletion to a single analog (in this case, the band target problem). These simulations modeled the situation in which the source problem is understood well, but the target problem is understood relatively poorly. It is likely that students usually attempt to solve a target problem by analogy to an earlier example problem when the earlier problem is better understood than the later problem. In Novick and Holyoak's (1991) research, it is likely that many subjects had an imperfect understanding of the target problem, and possibly also the source problem. The problem representations provided to ACME were written by the authors, who presumably have a better understanding of the problems than did the undergraduate subjects.

The results of these simulation experiments are shown in Table 2.11. Although there are fewer data points here than for the family-tree experiments, the pattern of results is obvious: virtually none of the deleted propositions were correctly regenerated. In the most successful simulation—40% deletion from the band problem only—only 11% of the deleted propositions were inferred. In addition, a small number of commission errors were made, as well as occasional "redundant regeneration" of propositions that had not, in fact, been deleted. (Note that the sum of these two types of errors plus correct inferences can exceed the number of propositions deleted.) This result sharply contrasts with the case where the domains are isomorphic, as in the family-tree experiment. In that case, ACME is highly effective at regenerating information which has been removed from the representation of the target domain. However, when the domains are not isomorphic, ACME is virtually incapable of regenerating any information which has been deleted from the target domain. This behavior leads to the untested prediction that humans ought to be much less capable of making appropriate analogical inferences when presented with impoverished representations of nonisomorphic domains than when they are presented with impoverished representations of isomorphic domains.

The failure to obtain useful pattern completion for this example can primarily be attributed to proposition mismappings. Because the analogs are not isomorphic, each has a number of extraneous propositions which are not mapped to anything before deletion. When a mapped proposition from one analog is deleted, its "mate" will often remap to a similar but previously unmapped proposition, and will thus be over the threshold required for transfer candidacy. To illustrate a simple case where mismappings occur, consider the following example. In the band problem, we have three "orders" propositions:

B14: The director orders division by 12,
B15: The director orders division by 8, and
B16: The director orders division by 3.

The corresponding set of propositions in the garden problem is:

G13: Mr. Renshaw suggests division by 10,
G14: Mrs. Renshaw suggests division by 4,
G15: Mr. Renshaw suggests division by 5, and
G16: Mrs. Renshaw suggests division by 5.

Table 2.11.
CWS Pattern-Completion Results after Deletions from the Band and Garden Problem

Transfer: Deletion from Both Garden and Band

Propositions deleted		Correctly recreated	Generated with error	Redundant proposition generation	Total commission errors (erroneous + redundant generation)
%	#	#	#	#	#
2.5	4	0	1	3	4
5	8	0	0	4	4
10	17	1	1	2	3
20	35	2	0	0	0
40	70	0	2	0	2
60	106	0	1	1	0
80	140	0	0	0	0
90	156	0	0	0	0

Transfer: Deletion from Band Only

Propositions deleted		Correctly recreated	Generated with error	Redundant proposition generation	Total commission errors (erroneous + redundant generation)
%	#	#	#	#	#
5	4	0	3	4	7
10	8	1	1	4	5
20	17	1	3	4	7
40	35	4	3	4	7
60	53	5	2	4	6
80	70	2	1	0	1
90	78	3	1	0	1

Before deletion, ACME obtains the mappings *B14* = *G13*, *B15* = *G14*, and *B16* = *G15*. Here *G16* is an extraneous, unmapped proposition, but which could replace *G15* as the mapping from *B16*. When we delete *G13* and rerun the simulation, the mappings shift a bit. The mapping *B15* = *G14* remains the same, but now *B16* maps to *G16* (an accurate slippage), and *B14* maps to *G15*, since the two propositions are somewhat similar, and since nothing else maps to *G15*. Hence, ACME fails to generate a transfer inference from *B14*, even though *B14* has no truly analogous proposition in the other analog. Because of such mismappings of propositions, CWS, in general, tends to fail to regenerate deleted propositions. It should be noted, however, that mapping of the basic concepts and numbers remains robust even when a significant number of propositions are deleted. For instance, ACME was able to correctly produce about 80% of the crucial mappings when 40% of the propositions were randomly deleted from the source and the target. This approximates the level of mapping performance observed by Novick and Holyoak (1991) for their human subjects.

To see how ACME fares when transferring a solution from a source problem to a non-isomorphic target, we applied the CWSG procedure to the solution field of the garden problem to generate an analogous solution for the band problem. The results, shown in Table 2.12, may be interpreted in English as follows: First, find the lowest common multiple (lcm) of 12, 8 and 3; the result will be a number (obj32328). Then generate a set of multiples of the lcm; the result of this operation will be a list (obj32333). Next, add 1 to each number in the list, generating a new list (obj32339). Finally, find the smallest number in the list obtained that is evenly divisible by 5 (obj32345).

This solution is entirely correct, except for the last operation. ACME incorrectly transports the source's solution constraint (i.e., find the lowest multiple divisible by the ultimate divisor in the problem), whereas the solution constraint actually indicated in the target problem (find a multiple greater than 45 and less than 200) is completely ignored. Additional mechanisms capable of making the connection between the target problem statement and the target solution are needed in order to mold the freshly imported source solution to fit the stated requirements of the target problem. This gap illustrates where pattern completion ends and adaptation begins. Although some adaptations are very difficult for solvers (as indicated earlier in the summaries of the empirical results of Reed, 1987, and Novick & Holyoak, 1991), it seems that this particular adaptation is fairly easy. Failure to perform this adaptation would lead subjects to indicate 25 as the answer to the band problem (rather than the correct answer of 145), because that is the smallest number in the corrected list that is evenly divisible by 5. However, only 3 subjects out of a total of 207 made this error.

The relative success of the CWSG transfer mechanism in generating an approximate solution to the target problem, in contrast to the relative failure of the CWS mechanism in restoring randomly deleted propositions from either the target problem alone or both the source and target problems, illustrates the need for pragmatic guidance of pattern completion when attempting to relate

Table 2.12.
CWSG Pattern Completion Results for Transferring the Garden Problem Solution to the Band Problem

Proposition	Target (Band) Proposition
G83	(FIND-LCMI (NUM12 NUM8 NUM3 obj32328))
G84	(NUMBER (obj32328))
G85	(FIND-MULTIPLESI (obj32328 obj32333))
G86	(LIST (obj32333))
G87	(LIST-PLUSI (NUM1 obj32333 obj32339))
G88	(LIST (obj32339))
G89	(FIND-LEAST-MULTIPLEI (NUM5 obj32339 obj32345))

nonisomorphic problems. In the CWS experiments, ACME did not receive any information about what propositions should be treated as most important in generating inferences, and hence was easily misled by inappropriate proposition mappings. In contrast, in the CWSG experiment ACME was told to restrict pattern completion to the crucial solution field, thus greatly reducing the adverse impact of nonisomorphic propositions in other parts of the source and target representations.

5. GENERAL DISCUSSION

5.1. Summary

In this chapter, we reviewed evidence that analogical transfer can be usefully decomposed into three substages, corresponding to mapping, inference by pattern completion, and evaluation/adaptation. We described an extension of the ACME model that can perform mapping by constraint satisfaction followed by pattern completion using "copy with substitution" (CWS) and "copy with substitution and generation" (CWSG) mechanisms. Our basic pattern of simulation results may be summarized as follows. When transferring knowledge from a source analog to a fundamentally isomorphic but incompletely understood target analog, pattern completion mechanisms are highly effective at generating an accurate and comprehensive set of inferences. However, when the source and target are not isomorphic, pattern completion is less effective in its capacity to generate appropriate inferences, although mapping tends to be fairly robust. Given guidance that focuses attention directly on transfer of the source solution (rather than allowing global pattern completion), ACME generates target solutions that are accurate up to the point at which adaptation based on direct knowledge of the target is required. Beyond that point, ACME necessarily fails to produce the correct solution to the target, because the model lacks any capacity for adaptation. In general, the model is consistent with the hypothesis that human analogical transfer involves a rather sharp break in performance between mapping and pattern completion on the one hand, which can be executed relatively easily, and adaptation on the other, which often is quite difficult.

5.2. Comparison with Other Connectionist-Style Models of Analogical Transfer

It is useful to compare the extended ACME model described here to other connectionist and symbolic-connectionist models of analogy. The most similar model is CARE (Nelson, Thagard, & Hardy, this volume). CARE is also based on the ACME model of analogical mapping, and uses a similar pattern-completion mechanism to generate analogical inferences. CARE's pattern-completion mechanism resembles a hybrid of the CWS and CWSG algorithms we described in this paper. In CARE, propositions are eligible for transfer only if they contain some element that is also present in the goal field of the source analog's representation; it thus uses pragmatic knowledge to guide pattern completion. CARE does not use a fixed threshold based on argument-mapping activations to prevent profligate transfer; however, for each inference it uses the activations to generate a confidence value. CARE goes beyond ACME in its ability to handle the adaptation phase of transfer. In CARE, rules of a localist-connectionist sort are triggered in parallel with the analogy process. Hence, domain-specific inferences about the initial representations, and about the analogical inferences generated by pattern completion, can effectively enhance the basic solution generated by the mapping and transfer processes. Unlike the extended version of ACME described here, the more complex transfer mechanisms embodied in CARE have yet to be tested against detailed experimental data concerning human problem solving.

Hofstadter and Mitchell's Copycat model (this volume) processes letter-string analogies, such as "if *abc* is changed to *abd*, how would *ijk* be changed 'in the same way?'" Copycat implements analogical pattern completion: based on the relationship between *abc* and *abd*, and between *abc* and *ijk*, it is capable of "filling in" the missing component of the target. Copycat is notable for its capacity to flexibly construct elaborated representations of the letter strings (cf. Chalmers, French, & Hofstadter, 1991). In this respect it resembles CARE, which also uses the products of analogical transfer to trigger rule-based inferences that flesh out the original input, thus influencing the process of forming an analogy. The generalizability of Copycat to domains other than letter strings, particularly problem domains that require complex propositional representations, is yet to be established.

Halford, Wilson, Guo, Gayler, Wiles, and Stewart (this volume) describe a tensor-product network that processes analogy problems. For example, when the network is presented with the problem "mother:baby :: mare: ?" it can correctly produce "foal" as the missing element in the analogy. Like the Copycat model, we would characterize this behavior as analogical pattern completion: once the network has established the appropriate relationships between *mother* and *baby*, and between *mother* and *mare*, the network completes the analogy based on the obtained correspondences. The main appeal of Halford et al.'s model is that it is implemented in an extremely simple connectionist network. However, it is unclear how this model could represent or process more complex analogies, such as those involving problem representations.

5.3. Back-Propagation and Analogical Transfer

It is also of interest to compare the ACME model with a major class of connectionist models, based on back-propagation learning, which hold promise for dealing with the problem of analogical transfer. A question of central importance is: Can the automatic generalization capabilities of back-propagation alleviate the need for complex symbolic implementations of analogy? Harris's work (1994) represents a direct attack on this question. Harris demonstrates that when mapping sentences to meanings, a distributed connectionist network is able to abstract invariants both within a particular domain and also between domains. Generalization within a domain might be construed as induction of rules, and generalization between domains might be construed as analogy. Since it is difficult to distinguish between the two types of generalization, Harris characterizes the range of generalization capabilities as a rule-analogy continuum, rather than a strict dichotomy. This viewpoint is broadly consistent with the philosophy of CARE (i.e., both rules and analogies can be processed by constraint satisfaction)—although in CARE, rules and analogical processes are explicitly programmed rather than learned.

The rule-analogy continuum can be illustrated by considering the network and domain of Hinton's (1986) project (described previously). If the proposition *Arthur has_father Christopher* were removed from the entire set of propositions, and then correctly regenerated during a transfer phase, we could invoke two alternative explanations to account for the transfer success. First, we could claim that the inference that Arthur's father is Christopher is an analogical inference. To support our claim we might demonstrate that the distributed representation of Arthur (i.e., the pattern of activation in the second layer of the network) is similar to the distributed representation of Emilio, that Christopher's distributed representation is similar to Roberto's, and that this similarity is instrumental in producing the appropriate output. This claim is tantamount to claiming that the network mapped Arthur to Emilio, Christopher to Roberto, and applied some form of copy-with-substitution to complete the cross-domain transfer.

Alternatively, we could claim that the inference was of the rule-based variety, rather than an analogical inference. To support this claim, we could demonstrate that the missing proposition could be regenerated correctly when training was performed only on the English family tree. If the network has capitalized on within-domain relational regularities (e.g., the husband of the mother of X is ubiquitously the father of X), the network may be characterized as inducing rules, rather than making analogies. We suspect that Hinton's network used a combination of these two strategies to produce the generalization behavior that it did.

The issue of characterizing the nature and quality of generalization that back-propagation networks are capable of performing remains somewhat murky. If backprop-style generalization is equivalent to or even more powerful than the requisite mechanisms of analogical transfer (i.e., mapping, pattern completion, and adaptation),then the problem of implementing these analogical mechanisms becomes trivial: we need only train a network on a set of propositions, and query the network for the missing propositions. For example, we could train a network on the problem statement, goal and solution of the garden problem, and the problem statement and goal of the band problem. If the network has learned to do analogical transfer from this training set, the network ought to be able to produce the solution to the band problem when it is appropriately queried.

The actual outcome of such a hypothetical experiment may not be immediately apparent. However, it is useful to examine the results of tests of a recent back-propagation model that we believe suggest limitations stemming from competing pressures to generalize within a domain versus between domains. The model we consider is St. John's (1990) model of story comprehension. The model is based on a recurrent architecture (Rumelhart, et al., 1986), in which propositional representations of the story are fed to the network in a sequential fashion. A hidden layer called the *story gestalt* contains the current representation of the entire story, and is fed back to the input layer at each time step. The current representation of the story may be extracted by probing the network with a particular predicate, which in turn produces the representation of an entire proposition associated with the predicate.

In one experiment, the network was trained on three stories about going places in a car. These stories involved driving to a beach, a restaurant, and an airport, respectively. Both the beach and the restaurant stories contained a set of propositions involving the details of driving to the destination (e.g., the distance to the beach was far, Andrew got in the car, etc.), whereas the airport story did not contain any similar set of propositions. When the airport story was fed into the network and then probed with the predicates of the nonexistent driving propositions, an interesting phenomenon occurred. Inferred propositions were created on the output layer; however, the roles of the propositions were incorrectly instantiated. For example, the network tended to activate either the beach or the restaurant items for the filler of the location/destination role, rather than the correct airport filler. In terms of our problem-solving materials, this behavior would be equivalent to the network producing the (source) garden solution when probed for the (target) band solution. In short, the network clearly did not learn to perform general analogical transfer.

This seemingly anomalous behavior can be understood in terms of competing pressures to generalize within a domain versus between domains. While the ideal behavior form the perspective of the experimenter is for the network to generalize between domains, or to learn a variablized, domain-general rule (making it possible to instantiate the driving proposition by retrieving or inferring the particular destination involved in the current story), the network may instead "generalize" simply by assimilating a new domain to one or more known domains. For example, the story gestalt for the new airport story may be sufficiently similar to that of the beach story as to elicit "beach" as the destination when the network is probed with a predicate in the driving subschema. That is, the network may behave as if an airport were just some sort of atypical beach. Although St. John suggests that additional specialized training concerning which items on the input layer are paired with identical items on the output layer could remedy this mis-instantiation behavior, the effectiveness of such a remedy remains to be demonstrated.

A different possible approach to using back-propagation to perform analogical transfer might exploit a recent branch of research concerned with constructing and processing *reduced descriptions* of complex symbolic structures using back-propagation (e.g., Pollack, 1988, 1990; Chalmers, 1990; Blank, Meeden, & Marshall, 1992; Lee, 1991). This work is mainly centered around Pollack's (1988) RAAM (Recursive Auto-Associate Memory) model (or variants of the basic model), which is capable of encoding binary trees of arbitrary depth into a fixed-width vector of features. A tree is constructed in RAAM by placing representations of two subtrees on the input and output layers of the network; after auto-associative learning by back-propagation, the pattern of activation established on the hidden layer (the "reduced description") is interpreted as representing the entire tree. This pattern of hidden-unit activation can be used in a recursive manner as an input component for other trees in which the original subtrees are constituents. Chalmers (1990) and Blank et al. (1992) have shown that back-propagation networks can be trained to perform syntactic manipulations on reduced descriptions of simple subject-verb-object sentences. This work is extremely provocative, since it raises the possibility that complex processing mechanisms can be learned rather than programmed; and further, that these processing mechanisms can operate in parallel on an entire data structure, represented by a vector-string reduced description, rather than requiring serial procedures that take apart structures and operate on the components.

How far can we take the implications of such work, and what are its implications for general processing mechanisms such as means-ends-analysis, analogy, and rule-based inference? Perhaps we might, for example, be able to train reduced descriptions of problems to produce reduced descriptions of the correct solution to the problems. While this idea is certainly appealing, we are skeptical that things could be as simple as this. Barnden (1992) offers detailed arguments as to why processing mechanisms based on distributed representations remain inadequate for important types of inferencing. While the preliminary results of experiments using such mechanisms are certainly intriguing, we believe that significant research remains to be done before anything resembling the reduced-description approach can solve difficult computational problems such as analogical transfer.

5.4. The Prospects for Connectionist Models of Analogy

ACME, like many of the other models described in this volume (e.g., CARE and Copycat), represents a hybridization of symbolic representations with connectionist-style processing mechanisms. In ACME, input representations are provided in a "classical" symbolic form that resembles predicate calculus. The mapping component dynamically forms a localist connectionist network, in which each possible mapping hypothesis is represented by a unit. Pattern completion is accomplished by an explicit symbolic algorithm that operates on the output of the relaxation algorithm that finds the optimal mapping between the source and target.

What are the prospects for "cashing out" the symbolic components of hybrid models such as ACME in neural-style machinery? We view this as an open question, which is closely related to the above discussion of possibilities for developing distributed connectionist models capable of general analogical transfer. Because complex analogical thinking clearly depends on structured representations, a major hurdle is to find suitable techniques for representing relational structure in connectionist terms, without losing the flexibility of human analogical processing. An adequate model of analogy requires a solution to the difficult problem for connectionism of keeping bindings straight within a complex structured representation; in addition, mapping crucially depends on being able to find a novel binding between elements of two complex representations, without hopelessly muddling the two representations together. Whatever eventual form a successful model of analogy takes, work in this area is likely to have general implications for future developments in connectionist theory.

REFERENCES

Barnden, J.A. (1992). Connectionism, generalization, and propositional attitudes: A catalogue of challenging issues. In J. Dinsmore (Ed.), *The symbolic and connectionist paradigms: Closing the gap* (pp. 149–178). Hillsdale, NJ: Erlbaum.

Blank, D.S., Meeden, L.A., & Marshall, J.B. (1992). Exploring the symbolic/subsymbolic continuum: A case study of RAAM. In J. Dinsmore (Ed.), *The symbolic and connectionist paradigms: Closing the gap* (pp. 113–148). Hillsdale, NJ: Erlbaum.

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, pp. 351–370). Los Altos, CA: Morgan Kaufmann.

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

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: Morgan Kaufmann.

Chalmers, D.J. (1990). Syntactic transformations on distributed representations. *Connection Science, 2*, 53–62.

Chalmers, D.J., French, R.M., & Hofstadter, D.R. (1991). *High-level perception, representation, and analogy: A critique of artificial intelligence methodology.* (CRCC Tech. Rep. 49). Bloomington: Center for Research on Concepts and Cognition, Indiana University.

Chi, M.T.H., Feltovich, P.J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science, 5*, 121–152.

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

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

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

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

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.

Hall, R. (1989). Computational approaches to analogical reasoning: A comparative analysis. *Artificial Intelligence, 39*, 39–120.

Harris, C.L. (1994). Back propagation representations for the rule-analogy continuum. In J.A. Barnden & K.J. Holyoak (eds.), *Advances in connectionist and neural computation theory, Vol. 3: Analogy, metaphor, and reminding* (282–326). Norwood, NJ: Ablex.

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

Hofstadter, D.R., & Mitchell, M. (1988). Conceptual slippage and mapping: A report on the Copycat project. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

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

Holyoak, K.J., & Koh, K. (1987). Surface and structural similarity in analogical transfer. *Memory & Cognition, 15*, 332–340.

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

Keane, M.T. (1988). *Analogical problem solving*. Chichester, England: Ellis Horwood.

Lange, T.E., & Wharton, C.M. (1994). REMIND: Retrieval from episodic memory by inferencing and disambiguation. In J.A. Barnden & K.J. Holyoak (eds.), *Advances* *in connectionist and neural computation theory, Vol. 3: Analogy, metaphor, and reminding* (pp. 29–94). Norwood, NJ: Ablex.

Lee, G. (1991). *Distributed semantic representations for goal/plan analysis of narratives in a connectionist architecture.* Unpublished doctoral dissertation, Department of Computer Science, University of California, Los Angeles.

Melz, E.R., & Holyoak, K.J. (1991). Analogical transfer by constraint satisfaction. *Proceedings of the 13th Annual Conference of the Cognitive Society.* Hillsdale, NJ: Erlbaum.

Novick, L.R. (1988). Analogical transfer, problem similarity, and expertise. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14*, 510–520.

Novick, L.R. (1992). The role of expertise in solving arithmetic and algebra word problems by analogy. In J.I.D. Campbell (Ed.), *The nature and origins of mathematical skills* (pp. 155–188). Amsterdam: Elsevier Science.

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

Pollack, J.B. (1988). Recursive auto-associative memory: Devising compositional representations. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

Pollack, J.B. (1990). Recursive distributed representations. *Artificial Intelligence, 46*, 77–105.

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

Reed, S.K., Dempster, A., & Ettinger, M. (1985). Usefulness of analogous solutions for solving algebra word problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 11*, 106–125.

Reed, S.K., Ernst, G.W., & Banerji, R. (1974). The role of analogy in transfer between similar problem states. *Cognitive Psychology, 6*, 436–450.

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

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

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

Schoenfeld, A.H., & Hermann, D.J. (1982). Problem perception and knowledge structure in expert and novice mathematical problem solvers. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 8*, 484–494.

Silver, E.A. (1981). Recall of mathematical problem information: Solving related problems. *Journal for Research in Mathematics Education, 12*, 54–64.

St. John, M.F. (1990). *The story gestalt: Text comprehension by cue-based constraint satisfaction.* Unpublished doctoral dissertation, Department of Psychology, Carnegie Mellon University, Pittsburgh, PA.

Winston, P.H. (1980). Learning and reasoning by analogy. *Communications of the ACM, 23*, 689–703.

APPENDIX I

A. ACME Representations of Work Problems Used by Reed (1987)

Pipe Problem (Source)

Start State

```
;the important objects are a small pipe, a large pipe, and an oil tank
   (pipe (obj_pipe_sm)P1)
   (pipe(obj_pipe_lg)P2)
   (tank(obj_oil-tank)P3)
;the two pipes combine to form obj_two-pipes
   (combine (obj_pipe_sm obj_pipe_lg obj_two-pipes)P4)
;number of hours for a pipe to fill tank is val_hrs*
   (fill-time (obj_pipe* obj_oil-tank val_hrs*)P5)
   (time(val_hrs*)P6)
;rate at which a pipe fills tank is val_1/hrs*
   (fill-rate (obj_pipe* obj_oil-tank val_1/hrs*)P7)
   (rate(val_1/hrs*)P8)
   (quotient-of(num1 val_hrs* val_1/hrs*)P9)
   (number (num1)P10)
   (one(num1)P11)
;val_hrs* to fill tank takes on different values in different situations
;for the small pipe working by itself
   (fill-time(obj_pipe_sm obj_oil-tank num12)P12)
   (instantiates (obj_pipe_sm obj_pipe*)P13)
   (instantiates (num12 val_hrs*)P14)
   (number(num12)P15)
   (twelve(num12)P16)
;for the large pipe working by itself
   (fill-time (obj_pipe_lg obj_oil-tank num8)P17)
   (instantiates (obj_pipe_lg obj_pipe*)P18)
   (instantiates (num8 val_hrs*)P19)
   (number (num8)P20)
   (eight (num8)P21)
;for the small and large pipes working together
   (fill-time (obj_two-pipes obj_oil-tank num_h)P22)
   (instantiates (obj_two-pipes obj_pipe*)P23)
   (instantiates (num_h val_hrs*)P24)
   (number (num_h)P25)
```

Goals

```
;the goal of the problem is to determine the combined working time
   (known (num_h)P26)
```

Solution

;the two pipes take different amounts of time to fill the tank
;we find the rate for each working alone
(divide! (num1 num12 val_1/hrs_sm)P27)
(fill-rate (obj_pipe_sm obj_oil-tank val_1/hrs_sm)P28)
(instantiates (val_1/hrs_sm val_1/hrs*)P29)
(divide! (num1 num8 val_1/hrs_lg)P30)
(fill-rate (obj_pipe_lg obj_oil-tank val_1/hrs_lg)P31)
(instantiates(val_1/hrs_lg val_1/hrs*)P32)
;the two pipes each do a portion of the complete task of filling the tank
;the portion completed by obj_pipe_sm is product_pipe_sm
(portion-completed (obj_pipe_sm product_pipe_sm)P33)
(multiply! (val_1/hrs_sm num_h product_pipe_sm)P34)
(proportion (product_pipe_sm)P35)
;the portion completed by obj_pipe_lg is product_pipe_lg
(portion-completed (obj_pipe_lg product_pipe_lg)P36)
(multiply! (val_1/hrs_lg num_h product_pipe_lg)P37)
(proportion (product_pipe_lg)P38)
;to write equation to solve for num_h
(add! (product_pipe_sm product_pipe_lg sum_whole_task)P39)
(equal(sum_whole_task num1)P40)
(number (val_1/hrs_sm)P41)
(number (val_1/hrs_lg)P42)
(number (product_pipe_sm)P43)
(number (product_pipe_lg)P44)
(number (sum_whole_task)P45)
(proportion (sum_whole_task)P46)

Typing Problem (Isomorphic Target)

Start State

;the important objects are two people and a manuscript
(person (ann) T1)
(person (florence) T2)
(manuscript (obj_ms)T3)
;the two people combine to form obj_two-people
(combine (ann florence obj_two-people)T4)
;number of hours for a person to type manuscript is val_hrs*
(type-time (obj_person* obj_ms val_hrs*)T5)
(time (val_hrs*)T6)
;rate at which a person types manuscript is val_1/hrs*
(type-rate (obj_person* obj_ms val_1/hrs*)T7)
(rate (val_1/hrs*)T8)
(quotient-of (num1 val_hrs* val_1/hrs*)T9)

(number (num1)T10)
(one (num1)T11)
;val_hours to type manuscript takes on different values in different situations
;for Ann working by herself
(type-time (ann obj_ms num10)T12)
(instantiates (ann obj_person*)T13)
(instantiates (num10 val_hrs*)T14)
(number (num10)T15)
(ten (num10)T16)
;for florence working by herself
(type-time (florence obj_ms num5)T17)
(instantiates (florence obj_person*)T18)
(instantiates (num5 val_hrs*)T19)
(number (num5)T20)
(five (num5)T21)
;for Ann and Florence working together
(type-time (obj_two-people obj_ms num_h)T22)
(instantiates (obj_two-people obj_person*)T23)
(instantiates (num_h val_hrs*)T24)
(number (num_h)T25)

Goals

;the goal of the problem is to determine the combined working time
(known (num_h)T26)

Solution: *unknown*

Tank Problem (Similar Target)

Start State

;the important objects are a small pipe, a large pipe, and a water tank
(pipe (obj_sm_pipe)K1)
(pipe (obj_lg_pipe)K2)
(tank (obj_water-tank)K3)
;the two pipes combine to form obj_two-pipes
(combine (obj_pipe_sm obj_pipe_lg obj_two-pipes)K4)
;number of hours for a pipe to fill tank is val_hrs*
(fill-time (obj_pipe* obj_water-tank val_hrs*)K5)
(time (val_hrs*)K6)
;rate at which a pipe fills tank is val_1/hrs*
(fill-rate (obj_pipe* obj_water-tank val_1/hrs*)K7)

(rate (val_l/hrs*)K8)
(quotient-of (num1 val_hrs* val_l/hrs*)K9)
(number (num1)K10)
(one (num1)K11)
;number of hours for tank to empty by use is num40
(empty-time (obj_use obj_water-tank num40)K12)
(time (num40)K13)
(number (num40)K14)
(forty (num40)K15)
;the rate at which it is emptied is num_empty-rate
(empty-rate (obj_use obj_water-tank num_empty-rate)K16)
(quotient-of (num1 num40 num_empty-rate)K17)
(rate (num_empty-rate)K18)
(number (num_empty-rate)K19)
;val_hrs* takes on different values in different situations
;for the small pipe working by itself
(fill-time (obj_pipe_sm obj_water-tank num20)K20)
(instantiates (obj_pipe_sm obj_pipe*)K21)
(instantiates (num20 val_hrs*)K22)
(number (num20)K23)
(twenty (num20)K24)
;for the large pipe working by itself
(fill-time (obj_pipe_lg obj_water-tank num15)K25)
(instantiates (obj_pipe_lg obj_pipe*)K26)
(instantiates (num15 val_hrs*)K27)
(number (num15)K28)
(fifteen (num15)K29)
;for the small and large pipes working together
(fill-time (obj_two-pipes obj_water-tank num_h)K30)
(instantiates (obj_two-pipes obj_pipe*)K31)
(instantiates (num_h val_hrs*)K32)
(number (num_h)K33)
;the small pipe and large pipe combined act at same time as water use
(use-time (obj_use obj_water-tank num_h)K34)

Goals

;the goal of the problem is to determine the combined working time
(known (num_h)K35)
;corrected for the simultaneous water use
(corrected-for (num_h K34)K36)

Solution: ***unknown***

B. ACME Representations of Mixture Problems Used by Reed (1987)

Nurse Problem (Source)

Start State

;the important objects are a weak acid solution, a strong acid
;solution, and a combined acid solution (a mixture).
(solution (obj_soln_weak)N1)
(solution (obj_soln_strong)N2)
(combine (obj_soln_weak obj_soln_strong obj_soln_mix)N3)
(solution (obj_soln_mix)N4)
;acid-concentration, acid-content, and amount of
;solution are variables
;val_conc*, and similar expressions, are variable numerical values
(acid-concentration (obj_soln* val_conc*)N5)
(percentage (val_conc*)N6)
;amount of acid in solution
(acid-content (obj_soln* val_acid*)N7)
(volume (val_acid*)N8)
;amount of solution.
(amount (obj_soln* val_pints*)N9)
(volume (val_pints*)N10)
;concentration times amount of solution = amount of acid
(product-of (val_conc* val_pints* val_acid*)N11)
;acid concentration takes on different values for the different solutions
;for the weak acid solution
(acid-concentration (obj_soln_weak num6%)N12)
(instantiates (obj_soln_weak obj_soln*)N13)
(instantiates (num6% val_conc*)N14)
(number (num6%)N15)
(six-one-hundredths (num6%)N16)
;for the strong acid solution
(acid-concentration (obj_soln_strong num12%)N17)
(instantiates (obj_soln_strong obj_soln*)N18)
(instantiates (num12% val_conc*)N19)
(number (num12%)N20)
(twelve-one-hundredths (num12%)N21)
;for the mixture of weak and strong acids
(acid-concentration (obj_soln_mix num8%)N22)
(instantiates (obj_soln_mix obj_soln*)N23)
(instantiates (num8% val_conc*)N24)
(number (num8%)N25)
(eight-one-hundredths (num8%)N26)
;the acid concentrations are ordered from strong to mixture to weak
(greater-than (num12% num6%)N27)
(greater-than (num12% num8%)N28)

```
    (greater-than (num8% num6%)N29)
;the goal of the problem is to determine how much of the weak and
;strong acid solutions are needed for the specified mixture
;for the mixture of weak and strong acids
    (amount (obj_soln_mix num_4.5)N30)
    (number (num_4.5)N31)
    (instantiates (num_4.5 val_pints*)N32)
    (four-point-five (num_4.5)N33)
;for the weak acid solution
    (amount (obj_soln_weak num_w)N34)
    (number (num_w)N35)
    (instantiates (num_w val_pints*)N36)
    (amount (obj_soln_strong num_s)N37)
    (number (num_s)N38)
```

Goals

```
    (known (num_w)N40)
    (known (num_s)N41)
```

Solution

```
;amount of strong solution = amount of mixture - amount of weak solution
    (subtract! (num_4.5 num_w num_s)N42)
;amount of acid takes on different values for the different solutions
;for the weak acid solution
    (acid-content (obj_soln_weak val_acid_w)N43)
    (multiply! (num6% num_w val_acid_w)N44)
;for the strong acid solution
    (acid-content (obj_soln_strong val_acid_s)N45)
    (multiply! (num12% num_s val_acid_s)N46)
;for the mixture of weak and strong acids
    (acid-content (obj_soln_mix val_acid_m)N47)
    (multiply! (num8% num_4.5 val_acid_m)N48)
;to generate equation to solve for num_w and num_s
    (add! (val_acid_w val_acid_s val_acid_m)N49)
    (number (val_acid_w)N50)
    (number (val_acid_s)N51)
    (number (val_acid_m)N52)
    (instantiates (val_acid_w val_acid*)N53)
    (instantiates (val_acid_s val_acid*)N54)
    (instantiates (val_acid_m val_acid*)N55)
```


Grocer Problem (Isomorphic Target)

Start State

```
;the important objects are peanuts, almonds, and a mixture of the two
    (nuts (obj_peanuts)G1)
    (nuts (obj_almonds)G2)
    (combine (obj_peanuts obj_almonds obj_mix)G3)
    (nuts (obj_mix)G4)
;unit-price, total-price, and lbs of nuts are variables
    (unit-price (obj_nuts* val_$/lb*)G5)
    (price (val_$/lb*)G6)
    (total-price (obj_nuts* val_$*)G7)
    (price (val_$*)G8)
    (lbs (obj_nuts* val_lbs*)G9)
    (weight (val_lbs*)G10)
    (product-of (val_$/lb* val_lbs* val_$*)G11)
;price per pound takes on different values for the different nuts
;for the peanuts
    (unit-price (obj_peanuts num_1.65)G12)
    (instantiates (obj_peanuts obj_nuts*)G13)
    (instantiates (num_1.65 val_$/lb*)G14)
    (number (num_1.65)G15)
    (one-point-six-five (num_1.65)G16)
;for the almonds
    (unit-price (obj_almonds num_2.10)G17)
    (instantiates (obj_almonds obj_nuts*)G18)
    (instantiates (num_2.10 val_$/lb*)G19)
    (number (num_2.10)G20)
    (two-point-one-zero (num_2.10)G21)
;for the mixture of peanuts and almonds
    (unit-price (obj_mix num_1.83)G22)
    (instantiates (obj_mix obj_nuts*)G23)
    (instantiates (num_1.83 val_$/lb*)G24)
    (number (num_1.83)G25)
    (one-point-eight-three (num_1.83)G26)
;the unit prices are ordered from almonds to mixture to peanuts
    (greater-than (num_2.10 num_1.65)G27)
    (greater-than (num_2.10 num_1.83)G28)
    (greater-than (num_1.83 num_1.65)G29)
;the goal of the problem is to determine how many pounds of
;peanuts and of almonds are needed for the specified mixture
;for the mixture of peanuts and almonds
    (lbs (obj_mix num30)G30)
    (number (num30)G31)
```

(instantiates (num30 val_lbs*)G32)
(thirty (num30)G33)
;for the peanuts
(lbs (obj_peanuts num_p)G34)
(number (num_p)G35)
(instantiates (num_p val_lbs*)G36
;for the almonds
(lbs (obj_almonds num_a)G37)
(number (num_a)G38)
(instantiates (num_a val_lbs*)G39)

Goals

(known (num_p)G40)
(known (num_a)G41)

Solution: ***unknown***

Alloy Problem (Similar Target)

Start State

;the important objects are a more pure copper alloy, a less pure copper alloy,
;and an alloy consisting of a mixture of the two
(alloy (obj_alloy_good)A1)
(alloy (obj_alloy_bad)A2)
(combine (obj_alloy_good obj_alloy_bad obj_alloy_mix)A3)
(alloy (obj_alloy_mix)A4)
;alloy-purity, copper-content, and lbs of alloy are variables
;purity is a relation between amount of copper and amount of alloy.
(alloy-purity (obj_alloy* val_%*)A5)
(percentage (val_%*)A6)
(copper-content (obj_alloy* val_cu*)A7)
(weight (val_cu*)A8)
(lbs (obj_alloy* val_lbs*)A9)
(weight (val_lbs*)A10)
;alloy-purity times lbs of alloy = copper-content
(product-of (val_%* val_lbs* val_cu*)A11)
;purity of alloy takes on different values for the different alloys
;for the more pure alloy
(alloy-purity (obj_alloy_good num20%)A12)
(instantiates (obj_alloy_good obj_alloy*)A13)
(instantiates (num20% val_%*)A14)

(number (num20%)A15)
(twenty-one-hundredths (num20%)A16)
;for the less pure alloy
(alloy-purity (obj_alloy_bad num_12%)A17)
(instantiates (obj_alloy_bad obj_alloy*)A18)
(instantiates (num12% val_%*)A19)
(number (num12%)A20)
(twelve-one-hundredths (num_12%)A21)
;the alloy purities are ordered from low to high
(less-than (num12% num 20%)A22)
;the purity of the mixture of the more and less pure alloys is not stated
;the goal of the problem is to determine how much of each component alloy
;must be melted together to get the specified mixture
;for the mixture of more and less pure alloys
(instantiates (obj_alloy_mix obj_alloy*)A23)
(lbs (obj_alloy_mix num60)A24)
(instantiates (num60 val_lbs*)A25)
(number (num60)A26)
(sixty (num60)A27)
;for the more pure alloy
(lbs (obj_alloy_good num_g)A28)
(instantiates (num_g val_lbs*)A29)
(number (num_g)A30)
;for the less pure alloy
(lbs (obj_alloy_bad num_b)A31)
(instantiates (num_b val_lbs*)A32)
(number (num_b)A33)
(copper-content (obj_alloy_mix num_10.4)A34)
(instantiates (num_10.4 val_cu*)A35)
(number (num_10.4)A36)
(ten-point-four (num_10.4)A37)

Goals

(known (num_g)A38)
(known (num_b)A39)

Solution: ***unknown***

APPENDIX II
A. ACME Representation of Garden Problem (Source) Used by Novick and Holyoak (1991)

Start State

```
;plants grow in a garden
   (plants (obj_plants)G1)
   (garden (obj_garden)G2)
   (grow-in (obj_plants obj_garden)G3)
;the number of plants first considered is num_total_G0
   (number-of (obj_plants num_total_G0)G4)
;dividing num_total_G0 by 10, 4, or 5 leaves 0 remainder
   (divide (num_total_G0 num10 quotient_G1)G5)
   (remainder-of (G5 num0)G6)
   (divide (num_total_G0 num4 quotient_G2)G7)
   (remainder-of (G7 num0)G8)
   (divide (num_total_G0 num5 quotient_G3)G9)
   (remainder-of (G9 num0)G10)
;Mr. Renshaw suggests dividing by 10, Mrs. Renshaw suggests
;dividing by 4, and both suggest dividing by 5
   (person (mr_renshaw)G11)
   (person (mrs_renshaw)G12)
   (suggests (mr_renshaw G5)G13)
   (suggests (mrs_renshaw G7)G14)
   (suggests (mr_renshaw G9)G15)
   (suggests (mrs_renshaw G9)G16)
;the Renshaws' daughter points out that the total number of plants,
;num_total_G0, can be increased by 2 to num_total_G1
   (person (renshaw_daughter)G17)
   (plus (num_total_G0 num2 num_total_G1)G18)
   (points-out (renshaw_daughter G18)G19)
;she points out that dividing by 10, 4, or 5 into num_total_G1
;leaves a non-zero remainder of 2
   (divide (num_total_G1 num10 quotient_G1)G20)
   (remainder-of (G20 num2)G21)
   (divide (num_total_G1 num4 quotient_G2)G22)
   (remainder-of (G22 num2)G23)
   (divide (num_total_G1 num5 quotient_G3)G24)
   (remainder-of (G24 num2)G25)
   (not-equal (num2 num0)G26)
   (number-extra-spaces (obj_plants obj_garden num2)G27)
   (points-out (renshaw_daughter G21)G28)
   (points-out (renshaw_daughter G23)G29)
   (points-out (renshaw_daughter G25)G30)
```



```
;she suggests dividing num_total_G1 by 6, leaving 0 remainder
   (suggests (renshaw_daughter G32)G31)
   (divide (num_total_G1 num6 quotient_G4)G32)
   (remainder-of (G32 num0)G33)
;division by 6 is a success because it leaves 0 remainder
   (success (G32)G34)
   (cause (G33 G34)G35)
;all the numbers mentioned are numbers
   (number (num_total_G0)G36)
   (number (num_total_G1)G37)
   (number (num0)G38)
   (number (num2)G39)
   (number (num10)G40)
   (number (num4)G41)
   (number (num5)G42)
   (number (num6)G43)
   (number (quotient_G1)G44)
   (number (quotient_G2)G45)
   (number (quotient_G3)G46)
   (number (quotient_G4)G47)
;various numbers have specific values
   (zero (num0)G48)
   (two (num2)G49)
   (ten (num10)G50)
   (four (num4)G51)
   (five (num5)G52)
   (six (num6)G53)
;divisions by 10, 4, 5, and 6 constitute possible selections of
;kinds of plants
   (kind-of (obj_kind10 obj_plants)G54)
   (kind-of (obj_kind4 obj_plants)G55)
   (kind-of (obj_kind5 obj_plants)G56)
   (kind-of (obj_kind6 obj_plants)G57)
;10, 4, 5, and 6 are possible numbers per kind
   (number-per-kind (obj_kind10 num10)G58)
   (number-per-kind (obj_kind4 num4)G59)
   (number-per-kind (obj_kind5 num5)G60)
   (number-per-kind (obj_kind6 num6)G61)
;dividing num_total_G0 by 10, 4, and 5, and num_total_G1 by 10, 4, 5
;and 6 are the first to seventh solution attempts, respectively
   (first-try (G5)G62)
   (second-try (G7)G63)
   (third-try (G9)G64)
   (fourth-try (G20)G65)
   (fifth-try (G22)G66)
   (sixth-try (G24)G67)
   (seventh-try (G32)G68)
```

;10, 4, 5, and 6 are the first to fourth divisors considered, respectively
(first-divisor (num10)G69)
(second-divisor (num4)G70)
(third-divisor (num5)G71)
(fourth-divisor (num6)G72)

Goals

;the goal is to determine num_total_G1
(known (num_total_G1)G73)
;which can be divided by num_Gx to give quotient_Gx with 0 remainder
(divide (num_total_G1 num_Gx quotient_Gx)G74)
(remainder-of (G74 num0)G75)
;and is the smallest value possible
(subtract (num_total_G1 num0 num_difference_G)G76)
(minimal (num_difference_G)G77)
;where num_Gx = 6 and quotient_Gx = quotient_G4
(equal (num_Gx num6)G78)
(equal (quotient_Gx quotient_G4)G79)
(number (num_Gx)G80)
(number (quotient_Gx)G81)
(number (num_difference_G)G82)

Solution

;find the least common multiple of 10, 4, and 5
(find-lcm! (num10 num4 num5 lcm_G)G83)
(number (lcm_G)G84)
;find multiples of lcm_G
(find-multiples! (lcm_G list-of-multiples_G)G85)
(list (list-of-multiples_G)G86)
;add 2 to each number in the resulting list of multiples
(list-plus! (num2 list-of-multiples_G corrected-list_G)G87)
(list (corrected-list_G)G88)
;num_total_G1 is the lowest multiple of 6 in the corrected list
(find-least-multiple! (num6 corrected-list_G num_total_G1)G89)

B. ACME Representation of Band Problem (Target) Used by Novick and Holyoak (1991)

Start State

;band members march in the band
(band-members (obj_members)B1)
(band (obj_band)B2)

(march-in (obj_members obj_band)B3)
;the number of band members is num_total_B
(number-of (obj_members num_total_B)B4)
;dividing members by 12, 8, or 3 leaves a non-zero remainder of 1
(divide (num_total_B num12 quotient_B1)B5)
(remainder-of (B5 num1)B6)
(not-equal (num1 num0)B7)
(number-left-out (obj_members obj_band num1)B8)
(divide (num_total_B num8 quotient_B2)B9)
(remainder-of (B9 num1)B10)
(divide (num_total_B num3 quotient_B3)B11)
(remainder-of (B11 num1) B12)
;the director orders the above attempts
(person (obj_director)B13)
(orders (obj_director B5)B14)
(orders (obj_director B9)B15)
(orders (obj_director B11)B16)
;Andrew orders division by 5, which results in 0 remainder
(member-of (andrew obj_members)B17)
(person (andrew)B18)
(left-out-of (andrew obj_band)B19)
(orders (andrew B21)B20)
(divide (num_total_B num5 quotient_B4)B21)
(remainder-of (B21 num0)B22)
;division by 5 is a success because the remainder is 0
(success (B21)B23)
(cause (B22 B23)B24)
;all the numbers mentioned are numbers
(number (num_total_B)B25)
(number (num1)B26)
(number (num0)B27)
(number (num12)B28)
(number (num8)B29)
(number (num3)B30)
(number (num5)B31)
(number (quotient_B1)B32)
(number (quotient_B2)B33)
(number (quotient_B3)B34)
(number (quotient_B4)B35)
;various numbers have specific values
(zero (num0)B36)
(one (num1)B37)
(twelve (num12)B38)
(eight (num8)B39)
(three (num3)B40)
(five (num5)B41)
;divisions by 12, 8, 3, and 5 constitute possible selections of groupings
;of the members

(grouping-of (obj_row12 obj_members)B42)
(grouping-of (obj_column8 obj_members)B43)
(grouping-of (obj_row3 obj_members)B44)
(grouping-of (obj_row5 obj_members)B45)
(row-groups (obj_row12)B46)
(column-groups (obj_column8)B47)
(row-groups (obj_row3)B48)
(row-groups (obj_row5)B49)
;12, 8, 3, and 5 are possible numbers per group
(number-per-group (obj_row12 num12)B50)
(number-per-group (obj_column8 num8)B51)
(number-per-group (obj_row3 num3)B52)
(number-per-group (obj_row5 num5) B53)
;dividing by 12, 8, 3, and 5 are the first to fourth solution
;attempts respectively
(first-try (B5)B54)
(second-try (B9)B55)
(third-try (B11)B56)
(fourth-try (B21)B57)
;12, 8, 3, and 5 are the first to fourth divisors considered, respectively
(first-divisor (num12)B58)
(second-divisor (num8)B59)
(third-divisor (num3)B60)
(fourth-divisor (num5)B61)

Goals

;the goal is to determine num_total_B
(known (num_total_B)B62)
;which can be divided by num_Bx to give quotient_Bx with 0 remainder
(divide (num_total_B num_Bx quotient_Bx)B63)
(remainder-of (B63 num0)B64)
;where num_Bx = 5 and quotient_Bx = quotient_B4
(equal (num_Bx num5)B65)
(equal (quotient_Bx quotient_B4)B66)
;and num_total_B is greater than 45 and less than 200
(greater-than (num_total_B num44)B67)
(less-than (num_total_B num200)B68)
(number (num_Bx)B69)
(number (quotient_Bx)B70)
(number (num44)B71)
(number (num200)B72)
(forty-four (num44)B73)
(two-hundred (num200)B74)

Solution: *unknown*

3
Integrating Analogy With Rules and Explanations*

Greg Nelson, Paul Thagard, and Susan Hardy

1. INTRODUCTION

In the past decade, analogy has been one of the most progressive research areas in cognitive science. Previously, there had been isolated investigations in philosophy, psychology, and artificial intelligence, but the 1980s brought substantial work on many aspects of analogy, particularly on how two analogs can be mapped to each other and on how analogs can be retrieved from memory. Case-based reasoning, which is analogy in workaday clothes with a restriction to single domains, became an active research area in artificial intelligence.

There are, however, important unresolved issues concerning the role of analogy in human cognition. One of the most pressing concerns the relation of analogy to other central cognitive processes. How, for example, is analogical problem solving related to rule-based problem solving in which chains of rules are used in quasideductive fashion to accomplish goals? One extreme view, implied by some of the advocates of case-based reasoning, is that there is no such thing as rule-based reasoning. At the other extreme, there is the view that analogy is of peripheral interest, at most a minor module to be added onto a rule-based system which handles basic cognitive operations. In between, there is the view that analogy and rule-based reasoning should be viewed as integrated aspects of a general cognitive system.

How rule-based reasoning can be integrated with analogical reasoning depends in large part on what computational mechanisms are seen as crucial to

* This research was supported by contract MDA903-89-K-0179 from the Basic Research Office of the Army Research Institute for the Behavioral and Social Sciences, and conducted at the Princeton University Cognitive Science Laboratory. We thank David Gochfeld for discussions and programming that helped guide us toward the current model, and Dmitry Gorenburgov for developing the Gorbachev example. For helpful comments on a previous draft, we are grateful to John Barnden, Keith Holyoak, and Heather Pfeiffer.