Information and Computer Science

UNDERSTANDING ANALOGICAL REASONING: Computational Approaches[1]

Rogers P. Hall

Irvine Computational Intelligence Project
Department of Information and Computer Science
University of California, Irvine, 92717

June 4, 1986

TECHNICAL REPORT

UNIVERSITY OF CALIFORNIA
IRVINE

UNDERSTANDING ANALOGICAL REASONING:
Computational Approaches[1]

Rogers P. Hall
Irvine Computational Intelligence Project
Department of Information and Computer Science
University of California, Irvine, 92717

June 4, 1986

Technical Report 86–11

Abstract

Analogical reasoning has a seductive history in artificial intelligence (AI) because of its assumed promise for the acquisition and effective use of knowledge. This paper, the second of a two part series, reviews a variety of efforts at capturing some of this promise in the existing AI literature. Studies are reviewed chronologically within broadly defined task domains, and a simple framework for examining components of analogical reasoning is developed for comparing computational studies. This framework, and many of the ideas surrounding it, are related to important but often overlooked contributions from other academic disciplines reviewed in the first paper of this series. Particular emphasis is given here to the role of analogical reasoning in machine learning.

[1] This work was supported in part by ONR grant N00014-85-K-0373

Contents

1. Introduction

Beginning little more than a decade ago, metaphor and analogy have grown to occupy a position of some popularity in current artificial intelligence (AI) research interests, both for practical and theoretical reasons. In areas of problem solving and learning, analogical reasoning is offered as a promising contributor to overcoming the explosive complexity of searching for solutions to novel problems or inducing generalized knowledge from particular events. In the area of natural language understanding, metaphor and analogy provide both practical goals for flexible language interfaces and theoretical goals for understanding figurative thinking and expression. In a very basic way, analogy presents a challenging epistemological question: when are two representational descriptions, for some purpose, alike? While interest for these issues in AI has grown, few substantial contributions have been forthcoming.

This paper is the second in a two–part series reviewing studies of analogy in AI and a variety of related disciplines. In the first paper, approaches to analogy and metaphor from psychology, linguistics and related disciplines were examined with a specific interest to identify what these disciplines might contribute to computational studies. In this paper, consideration of these contributions leads to the construction of a simple framework for discussing components of analogical reasoning. Computational studies are presented in a roughly chronological fashion, and the framework is used both to evaluate individual studies and to draw comparisons between different approaches. The paper concludes with a prospectus for further research on the use of analogy in reasoning and learning from a computational perspective.

1.1. Contributions from other disciplines

As noted in the conclusion to the first paper in this series, psychology, linguistics, philosophy and related disciplines encourage a number of general conclusions about metaphor and analogy which may be useful for computational approaches. Four general contributions are discussed: the prevalence of analogy in human reasoning, likely cognitive structures supporting analogy, corresponding process models of analogical reasoning, and the role of analogy in learning. In brief, metaphor and analogy are seen to be ubiquitous constituents of our cognitive experience, although estimates of the importance of these constituents vary widely: from misleading ornamentation to an indispensable epistemological bridge for acquiring new concepts. There has been a clear shift towards representational media in which the "higher–level" structure of similarity between the source and target of an analogy can be made explicit. Hence "schematic" or "structured" forms of knowledge representation are increasingly preferred. Process models of analogical reasoning generally make a strong distinction between access to analogical sources and their use. There is a growing appreciation for the necessity of studying analogy within a context of use (e.g., analogical problem solving) and in a manner which is sensitive to characteristics of the reasoner (e.g., expertise). "Learning by analogy," while increasingly popular within the AI community, is a less well–defined interest in the other literatures reviewed. Psychological studies of analogy in problem solving, for example, treat analogy as a reasoning or problem modelling mechanism, for which learning may play a facilitative role in spontaneous access (e.g., Gick and Holyoak's schema abstraction, 1983) or effective use (e.g., base specificity partially enables structure mapping in Gentner's framework, 1982).

While the literatures surveyed in the previous paper do not provide clear–cut answers to the underlying structures or processes of analogical reasoning, they do provide conceptual frameworks within which interesting questions have been identified and partial answers have been proposed. Computational studies would do well to consider these frameworks, hopefully building upon and contributing to increasingly detailed theoretical and empirical treatment of analogy and metaphor. The following section outlines a simple process model for analogical reasoning which is used throughout the rest of the paper.

1.2. A descriptive framework for studies of analogical reasoning in AI

Taken in isolation, computational studies of analogy and metaphor are difficult to understand as an integrated whole. Areas of commonality are difficult to extract both because of the recency of much work in AI and the variety of task domains in which computational approaches have been attempted. There are few comprehensive reviews of computational studies of analogy, a problem which this paper addresses. Before considering the body of work done in AI on analogy, however, a simple process framework for analogical reasoning will be presented, drawing from conceptual frameworks explored in the preceding paper. This framework gives process components which a relatively complete picture of analogical reasoning, computational or otherwise, would include:

1. *recognition* of a candidate analogous source, given a target situation,
2. *elaboration* of the ground between an retrieved source and the target situation (the ground may be a match or mapping between domains),
3. *evaluation* of the elaborated analogy in the context of application (issues of justification, repair or extension of an analogical mapping would fall here),
4. and *consolidation* of what can be learned from an elaborated analogy so that such knowledge will be subsequently available.

Further descriptive properties have been included as a result of the special nature of computational models of intelligent behavior, including representation, system inputs and system outputs. Final properties for describing studies are somewhat more evaluative, including completeness of process specification (across components described above), specificity of these descriptions for implementational purposes, plausibility of described performance (either as a result of implementation or optimism), and the generality or extensibility of a proposed computational model to wider competence in similar or different domains. As always, criticism of others' work comes more easily than does proposing novel solutions to identifiable problems. Hopefully, critical evaluation in this case will further a process of identifying important research problems.

1.3. An itinerary for reviewing individual studies

Since study of analogical reasoning in AI is a recent phenomena by comparison with other disciplines, characteristic approaches (as in comparison versus interaction theories of metaphor in philosophy and psychology) and their attendant concerns have yet to solidify in the AI literature. Thus the studies which will be surveyed here are difficult to place in some coherent order of presentation. As an approximation to such an ordering, the studies will be roughly divided into the problem areas which they investigate: automatic deduction, problem solving and planning, natural language processing, and learning. Certainly these areas of interest are not mutually exclusive (e.g., work by Carbonell (1983a, 1986) examines problem solving and learning); however, they provide conceptual pegs on which to hang various efforts, allowing some basis for comparison between different studies. This situation is preferable to an exhaustive enumeration of studies ordered chronologically, for example.

As will be discussed in the concluding chapter of this paper, a number of general issues concerning analogical reasoning as an intelligent behavior and, more specifically, computational models of this behavior (whether motivated as cognitive or performance models), can be extracted. These will be discussed with an eye towards describing a prospectus for future research on analogical reasoning in AI with particular emphasis on machine learning. Now for a discussion of specific studies.

2. Early paradigms for analogical reasoning in AI

According to Ringle (1979), much early work in AI could be generally described as an attempt to get machines to do something, anything, intelligent as an existence proof for computational approaches to intelligence. To some extent, the three studies discussed in this section represent such an effort. However, these studies are of interest not only for their role as ancestors of subsequent work in this weak sense, but also because they provide an orientation towards analogical reasoning which has shaped much subsequent work in AI. Evans' (1968) system for solving pròportional analogies will be discussed in detail; Becker's (1969) model of analogical processes in induction will be examined in detail; and Kling's work on using analogies in automated deduction will be discussed in the abstract sense of providing a paradigm for subsequent studies. The details of Kling's work are discussed more fully in a following section.

2.1. Evans [1968] Solution of geometric-analogy questions

Evans (1968) describes an implemented system, ANALOGY, which solves geometric-analogy intelligence-test questions in which geometric figures are restricted to two dimensional line drawings in a single plane. In these problems (see Figure 1), the A/B pair of components and relations between them serve as a given source, while a correct C/D pair with corresponding relations serves as the target. In overview, ANALOGY solves problems of this sort in three steps:

1. Construct transformation rules which, on the basis of a specialized descriptive vocabulary, express how the A component could be altered to obtain the B component.
2. Construct similar rules which take the C component into each of the answer components.
3. Compare each of the transformation rules generated in step (2) with each rule generated at step (1). Select a single rule taking C into an answer which preserves the most information about an appropriate A to B transformation.

Transformation rules are not necessarily unique for pairs of components, since their constituent figures may be matched and transformed in multiple ways.

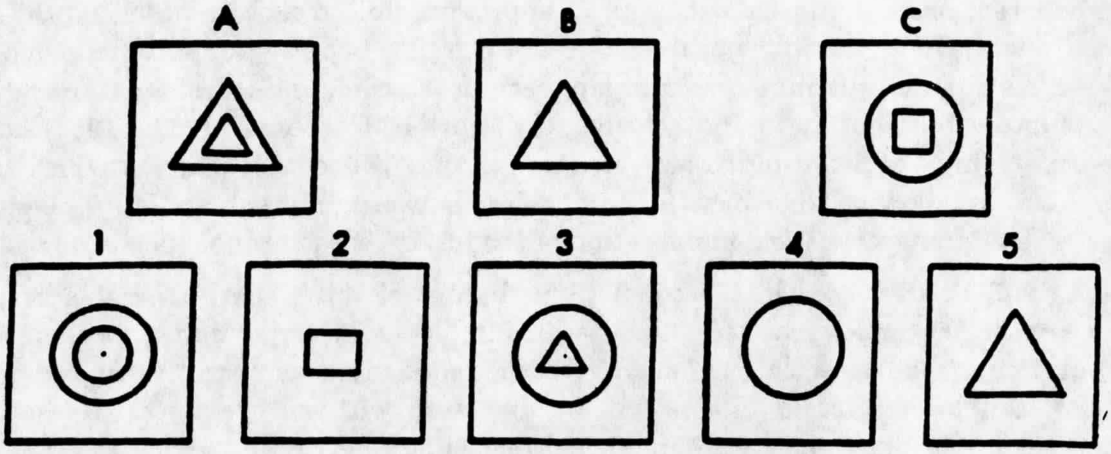

Figure 1
Figural analogy solved by ANALOGY (Evans, 1968).

Lists of attribute–value pairs are used as a representation and contain selected characteristics of geometric figures, relations among those figures and transformations of those figures. As input the system accepts a list describing each of eight components of the question (i.e., A:B::C:[five choices]) along with a set of parameters indicating which descriptive features of figures should be calculated (e.g., above, inside, left–of). Once components have been decomposed into constituent geometric figures (a process described at some length but not of particular interest here), relations among figures (e.g., inside) are computed using specific knowledge of analytic geometry. As output, the system returns a single best answer choice providing that one can be found.

Having generated a description of figures within each component, ANALOGY examines relations between figures of *different* components. Component matching for similarity calculation is rigidly constrained such that figures be of the same general type (e.g., a simple closed figure). Having found figure correspondence across components, the "similarity" of a pair of components (e.g., components A and B) amounts to a set of transformations involving scale, rotation or reflection such that one figure contained within the first component could be transformed into a corresponding figure in a second component.

Since figure matches required for similarity comparisons may not be unique (some figures may be absent or compatible with multiple figures of another component), a set of *candidate rules* taking component A into component B is constructed which accommodates additions, deletions and transformational matches between figures of both components. In a similar fashion, *candidate matches* between component C and each of the choice components are constructed. In each case of rule construction, constraints introduced by earlier computations are used to restrict the space of potential matches or transformation rules encountered in subsequent computations. For example, when evaluating rules taking the C component into an answer, transformations which do not preserve the same number of figures as the A to B rule(s) can be dropped immediately.

Selection of an answer candidate proceeds by matching transformation rules from the source (A/B) against possible target rules (C/answer component). This matching process can be described as constructing a "weakened" version of the rule taking A into B such that the version also takes C into one of the answer choices. Weakening is described by Evans as a form of generalization in which unnecessary conditions (for the hypothesized transformation rule) are simply dropped from the candidate rule. Obviously, a great many such weakened rules are possible even after excluding choice components which have differing subfigural constituents than those required by A and B. However, within the confines of analogy problems drawn from an American Council on Education examination, Evans reports considering a maximum of 36 such weakened rules for any single problem. A "best" candidate transformation is a minimally weakened rule taking C into an answer component. By requiring that the chosen rule (and thus answer) be as similar to an A to B transformation rule as possible, the system chooses an answer which best preserves the transformation relationship between A and B components. A similarity metric which is sensitive to differences in transformational matches (e.g., rotating a subfigure some number of degrees) is used to rank candidate answers. Should a tie among choice components occur, the system reports failure.

In terms of coverage, Evans work primarily addresses the issue of elaborating an existing analogy between two representations within the same problem domain. Issues of recognition, evaluation or consolidation are not considered. Some aspects of ANALOGY incorporate quite specific knowledge of geometry. However, mechanisms for constructing transformations between represented objects, generalizing those transformations and then choosing amongst a variety of candidate rules are described in a fashion quite independent of domain-specific knowledge. Hence many of the mechanisms of ANALOGY would seem applicable to other problem domains assuming that an adequately descriptive set of relational predicates could be assembled. As a final note, Evans' work could be criticized in that generalizations are not allowed over figure relations within components (e.g., inside, above), preventing the solution of analogies based on transformation of relations as well as objects (a criticism made by Sternberg, 1977). Evans is aware of this shortcoming, proposing that it would be easy to alter similarity calculations to reflect these transformations, but warns of the combinatorially explosive difficulties involved in using this approach indiscriminately.

In summary, the concept of constructing generalized transformation rules on the basis of similarity between constituent components of a representation is both clear and suggestive (a point recognized by Evans in the conclusion of his paper). This approach, strongly syntactic once geometric relations between figures have been calculated or given, makes the combinatorial difficulties of finding satisfactory matches between simple relational descriptions of a *given* source and target quite clear. Not surprisingly, Evans' work has contributed to many of the computational studies of analogical reasoning presented in latter portions of this review.

2.2. Becker [1969] Analogical processes in induction

Becker (1969) describes an implemented model of analogical reasoning and induction of simple rules. Analogy serves two distinct roles in this model: (1) organizing related information prior to inductive generalization and (2) using predictive information (existing rules) in novel situations. As a representation of knowledge, descriptive facts in a simple world of animate characters are described as being successively composed of ***nodes, kernels,*** and ***situations***. Nodes represent atomic conceptual elements in the world (e.g., "fireman" as a class of objects); kernels represent predicates (e.g., [member Wilfred fireman]); and situations represent a conjunction of predicates giving an assertion about the world:

{[wears Wilfred AA]
[member AA suspenders]
[property AA red]
[time @ Thursday] }

The symbol, @, in the last kernel of the situation above designates that the enclosing situation occured on Thursday.

Criteriality of nodes within kernels or kernels within situations can be represented by integer–valued weights attached to the appropriate components – e.g., {[member^{4} Wilfred^{2} fireman^{4}]:4}. A higher weight encodes higher salience for the node (or kernel) within the enclosing kernel (or situation). Alteration of criteriality values as a result of experience allows the system to generalize by turning constants into variables (node criteriality goes to zero) or dropping conditions (kernel criteriality goes to zero). Situations can either represent isolated assertions of fact about the world, or they can be combined to give left and right–hand sides of rules. Rules may have estimates of ***subjective probability*** associated with them, which record their relative frequency of successful. When a rule's subjective probability drops below threshold, that rule is expunged from memory.

Becker defines analogy as a one–to–one mapping between kernels of two situations, providing that node–node pairings are consistent for the mapping. Thus, analogy is a symmetric and invertable relation between two situations. Becker uses this symmetry to propose a *prediction paradigm* in which a novel situation can be mapped into the left–hand side of an existing rule, and this mapping can then be used to inverse–map the rule's implication (right–hand side) into a predicted novel situation. Hence an existing rule serves as a source for the analogy, the novel situation a target, and the predicted situation results from elaborating the ground between source and target.

The mapping between source and target situations is to be a *motivated correspondence* in that non–identical node pairings are justified by assertions within the situations or retrieved from memory. For example, assume a second person, Cyrus, is also known to be a fireman – i.e., [member Cyrus fireman]. Now assume the system observes Cyrus wearing red suspenders in the city of Peoria.

{[wears Cyrus BB]
[member BB suspenders]
[property BB red]
[location @ Peoria] }

According to Becker, this latest observation can be understood (or assimilated in the Piagetian sense) by analogy to the prior observation of Wilfred. However, to place non-identical nodes in direct correspondence – e.g., Wilfred and Cyrus, justifying facts must be collected which give evidence for the correspondence of these nodes at a relational level – e.g., both are members of the class of firemen. Similarity of class membership for these two nodes serves as a motivating justification for the analogy between observed situations. Similar justifications must be assembled for associating different instances of suspenders in each situation – e.g., both are members of the class of suspenders and both are red in color.

Collected justifications for situations participating in an analogy provide a conjunctive record of the facts which have been important in elaborating that analogy. Becker refers to these collections (one each for source and target) as the "*Path* of the Analogy" (p. 661), and performs a *path compression* operation over each of them which records this simple association. Path compression yields a new situation which is the union of the original situation and all collected justifications. Becker claims that subsequent analogies with these augmented situations will be easier to construct since past justifications will be predictive of future justifications.

In addition to facilitating justification of future analogies, augmented situations eventually provide the materials out of which more general rules may be constructed. For example, Becker claims that after seeing a sufficient number of situational analogies as described above, the system would be able to summarize their compressed records in the form of a rule. Selecting one situation arbitrarily as being prototypical, a rule is constructed by lowering the criteriality of nodes which take on multiple values (e.g., Wilfred, Cyrus, ...) or kernels which occur infrequently. The resulting rule might appear as follows:

$$\{[\text{member}^4\ \text{Cyrus}^2\ \text{fireman}^4]{:}4\} \quad \Rightarrow \quad \begin{array}{l} \{[\text{wears}^4\ \text{Cyrus}^2\ \text{BB}^2]{:}4 \\ [\text{member}^4\ \text{BB}^2\ \text{suspenders}^4]{:}3 \\ [\text{property}^4\ \text{BB}^2\ \text{red}^4]{:}3 \\ [\text{location}^4\ @^4\ \text{Peoria}^4]{:}1\ \} \end{array}$$

Node criterialities, lowest for Cyrus and the suspenders token (BB), allow these components to be treated as variables in future uses of the rule. Essentially the generalized rule states: "if a man is a fireman, then he is likely to wear red suspenders." According to Becker, the converse of this rule would also be formed. At inception, a rule is given a subjective probability of 1/4, a value which will vary between a lower threshold and 1 (perfect predictive performance) over the lifetime of the rule.

Becker's work covers aspects of all four process components mentioned in the beginning of this survey, although without complete specificity. Given a new situation to be "understood," analogies are recognized during a search of memory apparently containing each fact (or situation) which the system has yet seen or composed. Candidate sources are eventually ranked by the degree to which they match the target situation, the criteriality of matched and unmatched components, and the depth to which the matching effort must proceed. Given an initial analogical mapping consisting of node pairings, justifications are collected which support these pairings. The fully elaborated (or justified) mapping is then compressed within target and source situations, giving new factual representations which Becker claims facilitate further analogical reasoning. When a compressed situation succeeds in predicting justifications for a new analogy, the system attempts to form rules which express the predictive aspects of this justification. Hence, some portion of the compressed analogical record being formed is asserted to be predictive of the remainder of the record. Generalization over similar cases of analogical reasoning is accomplished by altering the criterial weights associated with various nodes and kernels in both sides of a rule. Rule formation and storage of compressed situations constitute a process of consolidation for this model.

As mentioned above, Becker's process model leaves something to be desired at the level of detailed specification. It is far from clear how memory is searched for candidate analogs, either isolated situations or rules. In fact, some examples are given for which closer analogical matches can be found among exemplary materials than those used. Also, although a scoring function for analogical similarity is mentioned, it is not clear how such a scoring mechanism would be applied to help constrain search. Becker acknowledges that the process model which he proposes is syntactically–based and "cannot ensure that the analogies produced will be semantically meaningful – i.e. that the Justifying Information will in fact be relevant" (p. 660). Thus, although Becker acknowledges the crucial importance of context in constructing relevant analogies, his process model is practically context–blind. In addition, he notes that there is no provision for termination of analogy construction or justification so that the system would not infinitely regress while assimilating some arbitrary fact.

In summary, Becker's work provides an early example of work on analogical reasoning and learning in AI. He rather remarkably anticipates many of the issues which continue to play a central role in AI research on analogy and learning: the difficulty of recognizing and retrieving a source, the importance of contextual relevance, assembling justifications for an elaborated mapping between source and target, the predictive role which analogies play in reasoning, and the contribution of analogy to inductive processes.

2.3. Kling [1971a] Paradigms for analogical reasoning

Although the details of Kling's work on automatic deduction will be considered in the next section, his claim to advancing a "paradigm" for computational approaches to analogical reasoning is worth examination here. In fact, Kling presents two distinct paradigms. The first involves the application of a transformational mapping to a retrieved plan for solving an analogically related (source) problem. This results in a usable plan for the target problem. The transformed plan is then applied in the target problem domain to obtain a solution, circumventing traditional problem solving activities (search) in arriving at a solution for the target problem. The second paradigm, which Kling actually adopts in the ZORBA system described shortly, involves limiting the space of candidate axioms which might participate in the proof of a current theorem on the basis of axioms used in the proof of an analogous theorem. Although Kling's work is firmly based in the second paradigm, the former paradigm can be found to reverberate through much subsequent work in the field. In fact, the plan re–use notion probably corresponds to what most AI researchers would produce if asked to characterize analogical reasoning.

As mentioned in the beginning of this section, the three studies discussed above not only represent early computational approaches to analogy, but also appear to have established an orientation towards analogical reasoning and learning by analogy which has

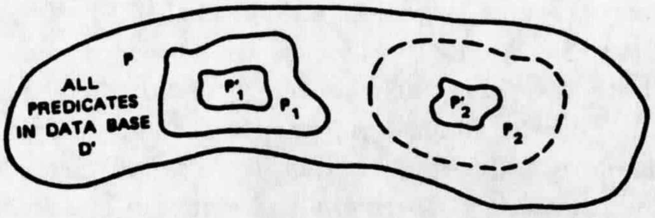

Figure 2
Using an analogous proof to constrain a starting set of predicates (Kling, 1971b)

shaped subsequent work in the field. Evans' work on proportional analogies highlights issues of elaborating an analogical mapping between representational descriptions. This amounts to a search for transformations which take one description into another and then finding a "best" match between candidate transformations. If transformations are viewed as solution paths which result from a problem solving process, finding a match between these solution paths stands as a clear precursor to subsequent work. Likewise, Becker anticipates many of the issues which have continued to be of importance in evaluating an analogical mapping (e.g., collecting justifications) and in consolidating the results of analogical reasoning (e.g., forming generalized predictive rules). Finally, Kling describes alternate approaches to analogy which stand as exemplary cases for subsequent work in the field. For example, Carbonell's work on transformational and derivational analogy (Carbonell, 1983a, 1983b, 1986, discussed in a later section) owes a conceptual debt to each of these early projects.

3. Analogical Reasoning in Automated deduction

In this section, three approaches to automated deduction will be discussed. Kling (1971a, 1971b, 1971c) proposes a mechanism for constraining irrelevant inferences using a uniform theorem proving approach, Munyer (1981) exploits a conceptual parallelism between analogy and unification to reuse portions of previously accomplished proofs at both levels of transforming individual formulas and transforming entire derivations; and Greiner (1985b) develops the notion of abstraction–based useful analogical inference. These approaches are probably among the most detailed of any work done on analogical reasoning in AI to date.

3.1. Kling [1971a. 1971b. 1971c] Automating deduction with ZORBA

Kling describes an implemented system, ZORBA, which assists resolution-based theorem proving by constraining potential axioms participating in a new proof (the target) on the basis of analogy to an existing proof (the source) supplied by the user. As shown in Figure 2, the set of predicates in statements of the source and target problems, P'_1 and P'_2, can be placed in an initial correspondence mapping which is incrementally extended to give sets of predicates used in proofs of these statements, P_1 and P_2. Thus a known proof of the source theorem is used to restrict the set of facts considered by a uniform proof procedure working on the target theorem.

As a representation, clauses in conjunctive normal form are augmented by clause descriptions which keep track of negation and implication. Semantic templates which specify argument type and order for predicates are also represented. Inputs to ZORBA include formulas for both the target and source theorem, a clausal representation of the negation of the target theorem, clausal representations of axioms and resolvants in the proof of the source theorem, and semantic templates for all predicates in the system. In overview, a constrained database of theorems is assembled by incrementally extending an analogical mapping over variables, predicates and axioms. This extension process starts with a partial map over variables and predicates between the target and source theorems, and eventually includes axioms corresponding (via the extended mapping) to those used in the proof of the source theorem. The resulting analogy map is restricted to being 1–1 for predicates, 1–many for axioms and many–many for variables. The final, constrained database of axioms and target theorem are presented to a resolution-based theorem proving system, QA3 (Green, 1969).

Kling's description of the ZORBA system descends rather quickly into a detailed explication of subprograms which cleverly elaborate the growing analogy map. From a somewhat more abstract viewpoint, however, much of this process of incremental extension can be explained in terms of a search of a set of potential axioms guided by rather strict reliance on syntactic correspondence among predicates. This arises out of the fortuitous combination of a well-formed representational formalism with explicit semantic knowledge provided in the form of semantic templates and clause descriptions. Without criticizing the suggestiveness of this early work, the sense of analogy conveyed is one of strong similarity at the level of clausal correspondence, with indeterminacy in higher levels (i.e., the process of proving a theorem) left as an exercise for the theorem prover. That is, the analogy is not constructed at the level of a proof "approach," to the extent that such an approach would be identifiable using uniform proof techniques. Nonetheless, application of ZORBA to nontrivial problems in algebraic group theory results in starting axiom sets only slightly larger than optimal as identified by human experts. These constraints allow automated proof by resolution for difficult theorems within computational resource limitations.

Of the various components of analogical reasoning used in this survey, ZORBA focuses squarely on elaboration of an existing analogy with little evaluation and no consolidation. The mechanisms within the system for selecting candidate axioms seem generally applicable to problems amenable to resolution-based theorem proving. While Kling's use of analogy as a semantically and syntactically constrained mapping between domain statements proves effective, we might ask if choosing a candidate set of axioms on the basis of syntactic correspondence between theorems judged "analogous" by the user of the system actually provides the "paradigm for reasoning by analogy" as advertised. As noted keenly by Miller (1983), Kling's use of analogy is,

> Rather as though, possessing a recipe for lamb casserole, and wishing to cook a beef stew, we noted that we were likely to need beef, onions, potatoes, carrots, stock, salt, an oven, a knife, a dish and a work–surface, and then threw away the recipe book without reading the method of preparation. (p. 33)

A fuller sense of analogy in automated deduction is addressed in the next two studies of this section.

3.2. Munyer [1981] Analogy as relaxed unification

Central to Munyer's approach is an appreciation for the commonality between analogy, generalization, and unification in terms of describing similarities between objects (derivations, formulas or terms). According to Munyer, "viewing an analogy match as a generalization of unification allows it to be used in place of unification in applying inference rules to formulas" (p. 34). He describes a partially implemented system which proves target theorems by re-using retrieved components of analogically related derivations. Essentially, unification serves as a transformation operator on source derivations.

Specifically, unification is relaxed to allow bindings between unlike constant or predicate symbols, many–to–one bindings, commutative or associative reordering of terms, and bindings which delete either individual symbols or entire terms. Transformed derivations serve as tentative plans which "chunk" the combinatorially explosive space of potential proof approaches. This effectively reduces an exponential search of depth n into k exponential searches of depth n/k (where k is the number of chunks). Since plans giving rise to chunks may lack logical validity, successful proofs resulting from their use will often involve further analogy matchings or straightforward problem solving for missing plan steps. In the wake of a successful proof attempt, generalization both over the derivation of analogically related proofs and constituent formulas provides a learning mechanism. Generalized derivations or formulas are stored and can later be retrieved for use in new proofs.

To recognize analogous source derivations, Munyer proposes an associative lookup algorithm which, given a target formula, returns a candidate set of source formulas on the basis of similarity of functional containment for common symbols (variables are not included). Candidate sources are then filtered on the basis of their actual overlap with the new formula. Elaboration occurs in a bottom up fashion, piecing together a match on the basis of mutually constraining local mappings between individual symbols. Exactly how analogically related plans or derivations are retrieved and used is difficult to extract from Munyer's thesis. An "implicit planning method" attempts to "recognize a plan among the pattern of analogy matches... already found in the search" (p. 79). An "explicit planning method" applies analogical transformations to intermediate steps in a stored derivation and then evaluates (attempts to justify) the resulting plan steps. Alternate plans are explored in case of discrepancies. Hence, implicit planning occurs at the level of individual formulae while explicit planning attempts to make use of entire analogous derivations.

Consolidation occurs by recording in the associative database specific instances of successful target derivations, generalizations over source and target successes, and the individual components of those derivations. Wisely, Munyer makes some effort to eliminate redundancy in stored components, resulting in a database of formulas and derivations which are densely interconnected. The utility of each stored component is recorded as a ratio of successful to attempted uses, allowing deletion of stored components with a poor performance record. Using generalization over both derivational structure and component formulas, Munyer claims that the proposed system might learn everything from specific plans to general heuristics (e.g., isolation of an unknown on one side of an equation).

Since many system components have not been implemented and are described in a rather *ad hoc* fashion, it is difficult to tell if the entire system would behave as Munyer suggests. In fact, it is quite difficult to get a coherent picture of how the varied system components (i.e., generalization over terms, formulas and derivations) would be integrated into some global control structure. Munyer proposes a global task agenda mechanism, prioritized by logical certainty. However, as described, analogy matches are allowed from virtually any point in a derivation on the basis of a database which grows rapidly. In sum, it is not clear that Munyer's proposal would not trade one form of intractable search (general deduction) for another (unconstrained analogical inference).

3.3. Greiner [1985a, 1985b] Abstraction–based useful analogical inference

Greiner (1985a, 1985b) presents a recent attempt to formalize the notion of analogical reasoning and learning by analogy within the framework of automated deduction. The gist of this work is that unconstrained analogical inference, defined as conjecturing facts about a target domain from the existence of similar facts in a known source domain, is intractable. There are simply too many legal (if not plausible) conjectures available, even after being given hint that a target concept (A) *is like* a source concept (B). To remedy this situation, Greiner advocates a set of heuristic strategies for constraining the space of generated conjectures: prefer "useful" analogical inferences within some problem solving context, draw inferences only from existing **abstractions** that give " 'coherent' clusters of facts ... which encode solution methods to past problems" (1985b, p. 1), and prefer abstractions which require a minimal number of additional conjectures about the target domain in order to be useful. Each of these heuristics, within the larger framework of analogical inference, are given painstaking formal definitions. The result is a deductive framework for identifying and using abstraction–based useful analogical inferences (defined shortly). In addition, Greiner presents empirical results of an implemented simulation, NLAG, which is able to deduce answers to questions in analogically–paired domains of electricity ↦ hydraulics, algebraic operators ↦ other operators, and programming language control structures ↦ other control structures.

Greiner's thesis is dense with formal definitions which, in total, are beyond the scope of this review (and reviewer). However, a specific definition of "abstraction–based useful analogical inference" (1985b, p. 57) is indispensible for further discussion (see Figure 3). Given *Th*, a deductively closed and consistent collection of axioms which serves as a theory (knowledge base) for target and source domains, a statement of the target query (*PT*), and a hint that the target concept *is like* the source concept (A $\sim$ B), Greiner defines an ***abstraction–based useful analogical inference operator*** which yields a conjecture about the target concept, A. "Concept," as used here, actually refers to a function in the predicate calculus (e.g., **FlowRate** in a fluid system). This conjecture (φ) is equivalent to Greiner's notion of an abstraction, and will in practice amount to a collection of n–ary predicates which involve objects in the target domain. The action of this operator is subject to a number of constraints:

$$Th,\ \mathtt{A}\sim\mathtt{B}\ \mathrel{|\!\!\sim}_{PT} \varphi(\mathtt{a}_1,\ldots\mathtt{A},\ldots\mathtt{a}_n)$$

$$\begin{array}{lll}
\text{where} & \textbf{Common:} & Th \models \varphi(?\mathtt{b}_1,\ldots\mathtt{B},\ldots?\mathtt{b}_n) \\
& \textbf{Unknown:} & Th \not\models \varphi(\mathtt{a}_1,\ldots\mathtt{A},\ldots\ \mathtt{a}_n) \\
& \textbf{Consistent:} & Th \not\models \neg\varphi(\mathtt{a}_1,\ldots\mathtt{A},\ldots\ \mathtt{a}_n) \\
& \textbf{Useful:} & Th + \varphi(\mathtt{a}_1,\ldots\mathtt{A},\ldots\mathtt{a}_n) \models PT \\
& \textbf{Abstraction:} & Th \models AbstForm(\varphi)
\end{array}$$

Figure 3

Abstraction–based useful analogical inference (Greiner, 1985b, p. 57)

1. The conjecture must be **common** to both concepts, requiring that the conjecture as it applies to the source concept (B) is known of, or can be logically inferred from, the starting theory.

2. The conjecture could not have been logically inferred for the target concept (A) from the starting theory – i.e., it is **unknown**. This gives Greiner's sense of analogy as plausible, non–deductive inference.

3. The conjecture, as instantiated for the target (A), is **consistent** with the starting theory – i.e., the negation of any fact in the conjecture as instantiated for A could not be logically inferred from the theory.

4. When added to the starting theory, the instantiation of the conjecture for the target (A) is **useful** – i.e., it allows logical inference of the query, *PT*.

5. The conjecture is a relation existing within the starting theory that has been explicitly "tagged" as being an **abstraction**. In effect, only relations which have been *a priori* designated as abstractions are considered while pursuing an analogy.

This formal definition of analogical inferences, requiring they be useful in context and based on existing abstractions, gives a formal account of the successive constraints Greiner places on the process of analogical reasoning. Now for a discussion of the processes which achieve this formal notion and an example of their use.

Fortunately, in addition to an extensive formal presentation of NLAG's focus on abstraction–based useful analogical inference, Greiner does a careful job of explicating the actual processes through which the simulation achieves his formal sense of analogical reasoning (see Figure 4). To make discussion more concrete, we will examine the behavior of these processes within the confines of a particular problem solved by NLAG: for a fluid system in which an initial flowrate is temporarily diverted through two pipes, find the flowrate of the fluid through the second pipe, given information about the initial flowrate and characteristics of the two pipes. The desired flowrate (target query) is to be found by analogy to better understood relations about electrical circuits (the source).

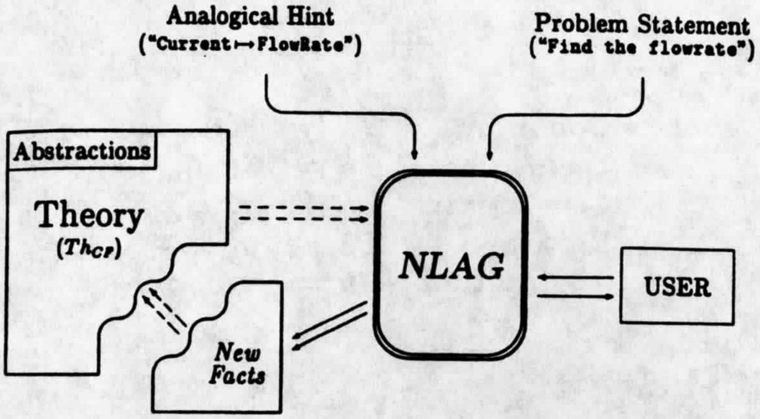

Figure 4
Overall structure of the NLAG system (Greiner, 1985b, p. 109)

For useful analogical inferences to be found, the system requires a variety of inputs. First, an *analogical hint* specifies a point of correspondence between target and source domains. In this case, the concept of **Current** is declared analogous to the concept of **Flowrate**. Second, the system is given a specific *problem* to solve, in this case to find the desired flowrate. Third, a *theory* (Th_{CF}) is given which includes many facts about the source domain (electrical circuits), few facts about the target domain (fluid systems), and a set of *abstractions* which organize general clusters of facts. For the flowrate problem, four different linear systems abstractions are included in the starting theory. The following is an example of the abstraction which eventually enables solution of the flowrate problem:

Abstraction(**RKK**)
RKK(t, c, r, l) ⇔ Kirchoff1(t) & Kirchoff2(c)
& ConservedThru(t, l) & Ohms(t, c, r, l)

Abstractions are identified by the "Abstraction(**X**)" predicate and are stated so that they apply to no particular domain. However, the starting theory is sufficient to instantiate abstractions within the source domain. In addition, the starting theory contains instantiations for some of the abstraction's predicates within the target domain (e.g., Kirchoff1(FlowRate)). As a final input, tests of user acceptance for useful and consistent conjectures are used to filter the system's output. This last input is less invasive that it may sound, since conjectures presented for user evaluation have already been shown to solve the target problem in a manner consistent with the starting theory. As output, the system shows a general abstraction, its instantiation in the target domain, and the set of specific predicates which support instantiation of the abstraction in the target domain.

At a process level, Greiner's NLAG model proceeds in an iterative generate and test fashion: the generator (ComAbs) finds a common abstraction for source and target, and a test process (Verify) determines both the effectiveness of this abstraction for solving the target problem and the acceptability of this abstraction to a user of the system. Finding a common abstraction proceeds in three steps:

1. *Find-Kernal.* The source concept is lexically substituted into the problem statement, beginning what appears to be a limited backward chaining search for rules in the starting theory which would contribute to proving the source problem statement. The result is a set of "kernal facts" necessary for deducing the source problem statement.
2. *Instantiate-Source.* Using forward chaining breadth–first search from the kernal facts, deduce a ground instance of an abstraction relation which mentions the source concept (e.g., **RKK**(Current, VoltageDrop, Resistance, Resistors)). This abstraction satisfies the common criterion mentioned above and requires least effort for instantiation vis-a-vis competing abstractions.
3. *Instantiate-Target.* Attempt to find a corresponding ground instance of the abstraction in which the source concept has been replaced by the target concept. This will typically produce "residual" conjectures which are required for the ground abstraction with the target concept to hold.

The second step essentially solves the source problem by finding an applicable abstraction. The result of the third step is a proposed solution to the target problem, based on re–using an abstraction found in the first two steps. Residual supporting conjectures for the proposed target solution which cannot be deduced may simply be asserted.

The proposed solution to the target problem is verified as follows:

1. Add the target conjectures from step three, above, to the starting theory and attempt to prove the target query. Proceed if the proof is successful, otherwise return for another abstraction.
2. Verify the consistency of each conjecture in the effective abstraction by attempting to prove its negation from the starting theory. Each such consistency test includes all other target conjectures in the theory. Proceed if all conjectures are consistent, otherwise seek another abstraction.
3. Ask the user (Greiner) to approve each new conjecture. If any conjecture fails, a new abstraction is sought.

The model described briefly above supports Greiner's notion of abstraction–based useful analogical inference: given a target problem and an analogical hint, the model selects among existing (known) abstractions which appear relevant, and attempts to deduce a target solution which Greiner finds reasonable. A variety of heuristics are used to constrain an intractable space of possible analogical inferences: abstractions, which encode "minimal" sets of information of demonstrable usefulness in the past, are preferred *a priori* as candidates for satisfying the *a posteriori* usefulness constraint. The following shows the instantiated abstraction and supporting conjectures which lead to a verifiable solution:

Abstraction(**RKK**)
RKK(FlowRate, PressureDrop, PipeCharacter, Pipes)

⇔

{Kirchoff1(FlowRate), Kirchoff2(PressureDrop),
ConservedThru(Flowrate, Pipes),
Ohms(FlowRate, PressureDrop, PipeCharacter, Pipes)}

Greiner presents the results of multiple empirical simulations of the model, within the electricity ↦ hydraulics analogy, in terms of the number of deductions required for each variant. Variant simulations are devised by weakening the general model successively. In the extreme, he finds that attempts to deduce the target query with no analogy mechanism typically exceed memory limits before reaching solution. Successive additions of heuristics for selecting abstractions and pruning possible instantiations of a given abstraction give increasingly efficient performance. For example, inclusion of a heuristic to prefer a maximally general abstraction results in a 15–fold reduction in the number of necessary deductions. Other results are less clear–cut. For example, when abstraction labels are removed, the system suffers no noticable degradation. Hence, the available relations in the starting theory are sufficient as sources of analogies, without benefit of segregating them as explicit abstractions. Greiner interprets this result positively, characterizing these relations as "*pre–defined* relations — which the ancient scholars, and others, have defined and named earlier." (p. 161) He has little to say about situations in which the starting theory is either poor with respect to such reifications or there are a very large number of them. Both seem to have many real–world counterparts (e.g., a novice solver of geometry problems).

In overview, Greiner's work addresses each of the four process components used in this survey. Recognition is accomplished by the synergistic combination of existing abstractions, an analogical hint, and a target problem. The former two ingredients simplify the recognition problem considerably: a source analog of the target problem enables retrieval of an abstraction which can be shown to be effective for solving the source problem. The problem of finding an applicable source domain is eliminated by the hint, while the subsequent issue of determining *what* to transfer is focused on existing, tagged abstractions. Given a candidate abstraction, elaboration of the common ground between source and target proceeds purely within the deductive requirements of the target problem. The abstraction serves as the invariant structure to be transferred between domains, and its instantiation after identification is independent of the source domain.

Evaluation is essentially a proof process which attempts to find an effective instantiation of the abstraction within the target domain. The user acceptance criterion appears to function as a filter on conjectures which must be asserted without deductive certainty within the existing theory. Hence, at least part of the evaluative process is given directly by the author. Finally, consolidation occurs as target domain conjectures required for instantiation of the transferred abstraction are added to the starting theory. The model could be said to learn in the sense that a set of facts have been assembled which were not in the starting theory. Some of these facts have been deduced while others have been asserted subject to user acceptance. Greiner mentions an interest in learning new abstractions as an area of future work, citing a variety of existing techniques in the machine learning literature as possible mechanisms.

In summary, the contents of the starting theory (or knowledge base) in Greiner's model exert a powerful influence on reasoning prowess. By assuming that the reasoner starts with abstractions or reified relations, explicit focus is given to processes of recognition, elaboration and evaluation. This focus is a form of model–driven or top–down reasoning which Greiner uses aptly throughout his model of analogical reasoning.

With respect to detail, the three studies described above are among the most exhaustive of those available in the computational literature on analogical reasoning. Munyer extends the mapping extension notions of Kling to include analogies between terms which ZORBA would probably not consider on the basis of discrepancies at the level of predicate correspondence. Further, Munyer's proposal for implicit and explicit planning attempts to utilize entire segments of the proof process (chunks), moving closer to Kling's idealized paradigm of plan re–use. Finally, Munyer devotes some attention to issues of recognition and consolidation, proposing an associative store for the former and liberal use of generalization for the latter. Kling's work constitutes a demonstration that analogical reasoning can facilitate automated deduction, while Munyer's work is suggestive of a larger role for analogy, including learning. Finally, Greiner presents a somewhat sobering perspective on unconstrained analogical inference within a formal deductive framework. To circumvent the combinatorially explosive number of legal (or even plausible) analogical inferences, he introduces a series of heuristic constraints, most notably the preference for re–using abstractions.

In addition to providing a frank view of the complexities of attempting analogical reasoning from a general computational framework, these studies also underscore the importance of being explicit about *what* is retrieved and used in pursuing an analogy. Kling demonstrates the utility of a rather indirect use of entire proof derivations; Munyer advocates using analogies both at the level of individual formula and successively larger proof plans; and Greiner argues strongly for the exclusive use of "coherent clusters of facts" – abstractions. As argued in the first paper of this series, it would be desirable to have a relatively complete and principled view of what can stand as the source of a useful analogy. For the studies in this section, given Greiner's arguments, the idea of allowing analogies at arbitrary levels of granularity as in Munyer's thesis promises to be unworkable. Again, it is important to avoid trading the heavy search costs incurred by "weak methods" against a comparable burden of determining the applicability of everything known for a current problem. This highlights the importance of recognition and consolidation as processes contributing to analogical reasoning: unfortunately, these have received the least attention to date.

4. Analogical reasoning in problem solving and planning

Although most of the studies in this survey involve some form of problem solving or could be described as such, the projects assembled in this section have a primary focus either on problem solving or planning. Analogy allows using solutions to previously encountered problems to help solve some current problem. Hence these systems are, on the whole, strict adherents to Kling's sense of analogy as plan re–use. In this section, three research projects will be discussed in some detail. Brown's (1977) work focuses on transferring problem solving expertise expressed at multiple levels of representation from one problem domain to another; McDermott (1979) describes a simple planning system which essentially memorizes every problem solution encountered and re–uses these subject to analogical transformations; and Carbonell (1983a, 1983b, 1986) proposes a variety of mechanisms for transforming retrieved solution derivations into suitable solutions for a new problem. All three projects also explicitly address issues of learning.

4.1. Brown [1977] Use of analogy to achieve new expertise

Brown proposes the use analogies between related problem domains (plane and solid geometry) with the goal of transferring problem solving expertise. Expertise in Brown's view consists of a domain *description* composed of axioms in first order logic, problem solving *plans* consisting of goals and subgoals with attendant constraints, and *programs* written in LISP which provide a computational mechanism for carrying out portions of plans. In addition, descriptive *justifications* for the consistency of plans play a central role in verifying the appropriateness of transferred knowledge. Justifications show that goals will indeed be achieved given subgoal satisfaction. Each of these sources of knowledge provides a language for computation, with successively less control information as one moves from programs to descriptions. Hence there are multiple levels of representation for domain knowledge, and analogical reasoning consists of "lifting" aspects of these knowledge sources from an "image world" of existing expertise (the source domain) into a "domain world" about which relatively little is known (the target domain).

Brown assumes the system starts with extensive expertise in the source domain (i.e., plane geometry), is given descriptive knowledge of the target domain (i.e., axioms of solid geometry), and then receives a carefully selected sequence of problems which can make use of a cumulatively growing analogy map between source and target domains. Chosen problems are presented as assertions (theorems) to be verified in the target domain. As output, the system is not only to have solved the posed problems, but to have acquired a properly transformed representation of expertise for the target domain including both plans and code.

Unfortunately, transformations between levels of representation within a particular problem domain dominate Brown's efforts. Since there is no evidence that the system has been implemented, specific discussion of the rather convoluted components would be of little value here. Instead, some aspects of this work are interesting at a more abstract level. Brown's analogy process is composed of three stages: *map, solve*, and *lift*. In brief, these stages map a target problem into a source problem, find a justifiable solution to the source problem, and them inverse map the solution and justifications back into the target domain.

In constructing a map, objects and predicates in the target problem description are placed in correspondence with objects and predicates in the description of the source domain. This mapping is constrained by "semantic templates" giving type information for objects in both domains. The resulting map (which goes from target to source domain) is applied to selected aspects of the target problem to obtain an analogous source problem. If the source problem can be solved, progress is straightforward in that an inverse analogy map can be applied to the source solution to suggest a target solution. This process is essentially identical to Kling's (1971a) notion of analogy as re–using solution procedures from a previous problem.

For a candidate target solution, the justification which supports the correctness of that solution in the source domain must be verified in the target domain before plans and code can be transferred. If a justifiable solution to the source problem cannot be found or violates expectations based on target domain knowledge, the mapping process can be continued to include more information regarding the original target problem. Inconsistencies between justifications and world descriptions, if severe, may necessitate use of a different analogy. Inconsistencies caused by missing or unnecessary components amount to "bugs" which may be "patched." Patches are then propagated through plans and code in the target domain.

The major contributions of Brown's work are difficult to evaluate. Recognition of an analogy between problem domains is given directly and followed by a sequence of carefully selected problems. Elaboration and evaluation of the developing analogy occur as a justified source solution (plans and code) is lifted into the target domain. Failure to achieve an analogous justification in the target domain results in a complicated process of debugging the lifted solution. Consolidation occurs to the extent that plans and code can be effectively lifted from source to target domains. The idea of incremental transfer of expertise on the basis of a sequence of problem solving tasks is interesting, but complicated by the need to repair transferred solutions at multiple levels of representation. Issues of organizing the store of source information that would accumulate as a result of success on a sequence of problems are not addressed. In summary, much of Brown's description of the proposed system seems *ad hoc*, with a variety of complicated mechanisms assembled around the necessity of vertical transformations between differently represented sources of knowledge. The result, unexpectedly, is described as a "theory" of analogical reasoning.

4.2. McDermott [1979] Learning to use analogies

McDermott describes "a production system that is capable of both assimilation and accommodation" by the use of analogy. By assimilation, McDermott means that the system is to view a new problem in terms of known problems; by accommodation he means the system is to be able to modify its knowledge so that new problems are familiar when seen again. An implemented system, ANA, solves problems by allowing an analogical match between a current goal (the target) and the goal statement of a method which has been successful in the past (the source). Analogy in this context refers to substitutions of object or action types for a method under consideration. Successes or failures with this approach are recorded by storing productions which will guide the system under similar conditions in the future. By McDermott's own estimate, ANA is an archetypical "hacker" in that everything from successful method application to exception conditions involving a single over-constrained precondition is "remembered" by the system in the form of an added production. Nonetheless, ANA does provide evidence of what types of performance can be achieved using relatively simple learning methods (rote memorization, essentially) in a highly constrained experimental environment.

ANA consists of a simulated paint shop environment in which a fixed set of recognizable objects can be found, moved about and sprayed with paint or water (see Figure 5). Implemented as a production system, ANA takes inputs describing the location of specific objects in the environment, and a starting set of six methods for accomplishing

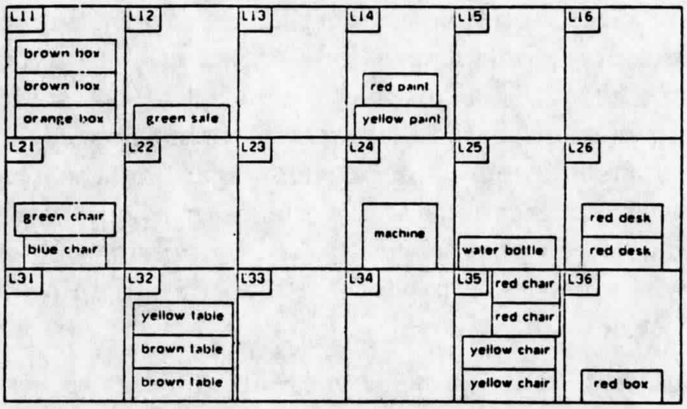

Figure 5
The paint shop in which ANA works (McDermott, 1979)

specific goals in the environment (e.g., paint a table at L32 red, carry tables in L32 to L23). Among other things, methods contain a set of subgoals which must be achieved for successful application, giving an implicit hierarchy of method types available to the system. Should a specific method not be available, the method hierarchy is consulted to give, in the context of analogy used here, recognition of an analogous method for use in achieving a current goal.

Given a task to perform (e.g., wash the thing in L12) for which no specific method yet exists, the system "searches" for a best analogous method based on the "mutability" of violated preconditions for using that method. The search for recognition of an analogous method is implicit: action and object type elements for the target goal statement are placed in working memory in the hopes of triggering distinguished enablement conditions of existing methods. These distinguished conditions, stored in a "method description production," give a form of partial matching between target goals and source methods. The partial match for a candidate source method must be further elaborated by considering additional method preconditions and constraints, some of which may be violated by the target problem statement. Eventually a "best" matching method is identified, although there may still be problems with applicability of the method.

Violated preconditions must be overcome by finding suitable attribute substitutions between the goal description of the target problem and the analogous source method. This elaborative process appears to use a variety of knowledge sources, including: background knowledge of object and action types, previously learned exceptional conditions (e.g., that an object in the middle of a stack cannot be carried until clear), and suggestions from the "task master" (McDermott himself). In the worst case, elaboration and evaluation of candidate source methods will simply fail since immutable differences cannot be overcome. Given an apparently applicable source, methods considered by ANA can still go wrong in several ways: by generating an inappropriate subgoal (e.g., loading paint into a spraying machine when the top level goal is washing), being underspecified (e.g., trying to carry a heavy safe), or being overspecified (e.g., moving an object to a specific location unnecessarily). In each of these cases, the analogous method is augmented with a production suggesting a "patch" should similar conditions be encountered in the future. Consolidation,

as mentioned before, amounts to remembering everything (successes and failures) in the form of added productions for methods, subgoals and patches. None of these productions are generalized in any fashion.

In summary, McDermott's ANA proposes mechanisms for each of the components expected in a complete account of analogical reasoning. Recognition is accomplished implicitly as partially matched candidate methods are activated. Elaboration and evaluation, less clearly described, rely on an interaction between task demands, retrieved methods, and a variety of background knowledge sources. At several points a "task master" is consulted for information regarding immutable difficulties or unnecessary constraints. It is not clear how much interaction between the task master and ANA is necessary to give the level of performance described. McDermott notes that a serious weakness of the system is its limited approach to selecting appropriate methods for a target problem. As the knowledge base of specific instantiations of methods grows (in fact, it grows without bound as described), this shortcoming could become crucial. Although ANA has been implemented and is able to accomplish tasks for which it initially has no specific methods, it is not clear that this approach would be sufficient for tasks requiring more complex planning in which subgoals interact.

ANA highlights two interesting issues in learning by analogy. First, recording specific successes and failures without generalization represents a resolution to the problem of *what to learn* and, hence, what is later available as a source for analogical reasoning. One view, that adopted by McDermott in ANA, is to acquire a body of specialized cases and provide a mechanism for adapting those cases to new target problem. In the limit, of course, this approach has drawbacks since there is effectively no compression of experience through learning. An alternative view, which we will see evidence of later in this survey, is to acquire generalized methods and provide a mechanisms for instantiating them for new target problems. This becomes a continuum along which various projects can be placed – e.g., Carbonell's work (1986) discussed in the next section covers both ends of this continuum. The second issue highlighted by ANA has to do with the granularity of information acquired during consolidation. Not only does ANA learn specific plan instantiations to achieve goals at the level of tasks presented directly to the system (e.g., paint an object at some location), but specific productions are also acquired for subgoals which contribute to solving the encompassing task (e.g., unstacking objects to clear a surface). Since recognition of methods for achieving subgoals can occur outside their original context of use through partial matching, analogical re–use of problem solving experience need not only occur at the level of goals given directly with the input task. While McDermott's work does not examine this possibility in any detail, subsequent studies of across task transfer have progressed in this direction (e.g., Laird, Rosenbloom and Newell, 1983).

4.3. Carbonell [1986] Reconstructive problem solving

Carbonell describes an approach which embeds analogical reasoning within more traditional approaches to problem solving and learning. Traditional problem solving techniques employ "weak methods" of searching for solution paths within a problem space defined by a set of primitive operators and state descriptions, or attempt to apply general

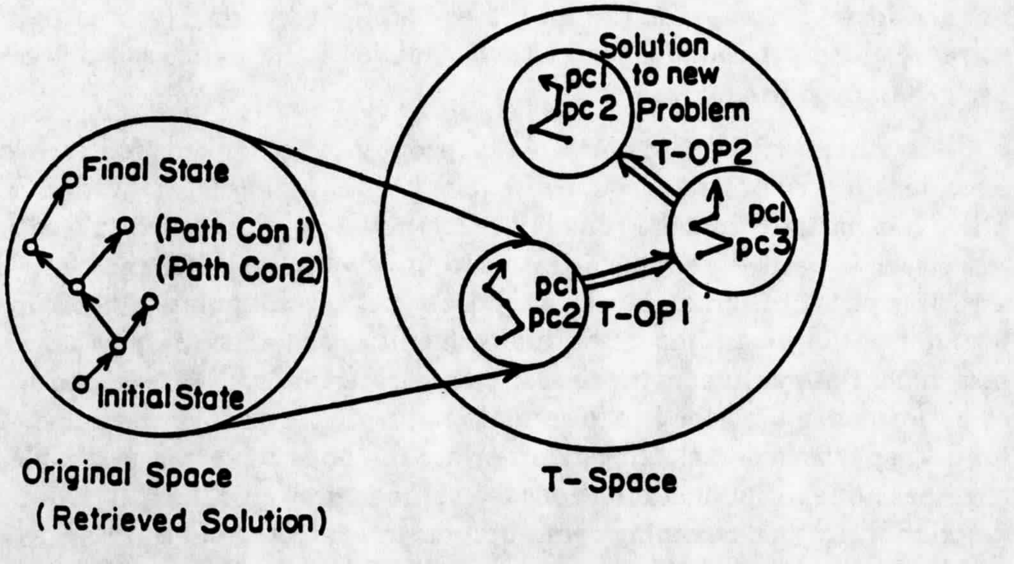

Figure 6
Analogy as search in a transformation space (Carbonell, 1983a).

plans which reduce the original task into (hopefully) easier subproblems. As with Kling's earlier notion of re–using existing solution procedures, *reconstructive* problem solving attempts to make use of previously solved problems or specific plans which have been induced from these problems. In addition to increasing problem solving power, reconstructive methods are to provide frequent opportunities for learning both specific and more general plans. Carbonell proposes two primary methods for achieving reconstructive problems solving: transformation of past solutions or re–playing derivational histories. Although derivational analogy partially subsumes the transformational work, both approaches will be reviewed here.

Transformational analogy is explored in Carbonell's ARIES system (1983a), where a solution path to a previously solved problems (the source) is incrementally transformed into a solution to a similar target problem. Traditional problem solving by means-ends-analysis (MEA) is extended into an "analogy transform problem space" (T-space). In T-space (see Figure 6), each state corresponds to an entire sequence of operator invocations in the original space. Each T-space operator (T-OP) gives a mechanism for modifying the encapsulated sequence to give a new T-space state which is also a sequence. Thus problem solving should occur in T-space in much the same way than problem solving occurs in a traditional problem space: enabling conditions for T-OP's are satisfied, their application gives a new sequence, and eventually a resulting sequence satisfies some goal specification. The central idea in transformational analogy is that analogical reasoning can be accomplished by invoking traditional problem solving techniques in T-space.

Given a target problem described in the original state space (initial state, goal state and constraints), ARIES retrieves a previously experienced problem solving episode with similar state descriptions and constraints. The retrieved problem solving sequence serves as the source of the developing analogy as well as an initial state in T-space, while a sequence which satisfies the original goal specification for the target problem will serve as a goal state in T-space. Analogical reasoning occurs in T-space as the previous solution sequence (states, operators and path constraints) is gradually transformed using T-OP's into a solution sequence which satisfies the target goal specification. A set of 11 T-OP's is described, including operators like *deleting* subsequences or *splicing* new subsequences into a current sequence. To control search in T-space ARIES uses difference–driven operator invocation for sequence transformations quite similar to GPS (Newell and Simon, 1972).

Inputs to the ARIES system are varied, but for present discussion the most important are a *similarity metric* used for ranking candidate matches between target and source problems, and a *dynamic memory* of previous problem solving cases. The similarity metric is sensitive to comparability of initial states, goal states, path constraints (e.g., cost), and the likely applicability of a candidate source. Unfortunately, memory organization is not discussed beyond use of a MOPs–like structures (Schank, 1982) which are not described. As output, the system is described as being capable of learning a variety of things: generalized plans are learned for solving clusters of problems related by having a common transformational source; successes and failures in problem solving can be used as instances to modify the similarity metric as well as difference table entries for original and T–space operators; and new T–space operators could be discovered by using conceptual clustering methods over problem solving instances where T–space MEA fails.

In terms of the process components used in this survey, Carbonell's work on transformational analogy is difficult to evaluate. Recognition occurs on the basis of partial matching of a target problem statement with source episodes in an organized memory. Unfortunately, neither memory structures, their organization, nor their access are described in any detail. Elaboration of the source solution path is more clearly described as the selection and application of transformation operators. Of 11 such operators described, only a subset are described as having been implemented. Although ARIES is described as "effective when tested in various domains, including algebra problems and route–planning tasks" (Carbonell, 1986, p. 376), results of these tests are not described. Furthermore, there is no description of mechanisms or experimental results for the proposed learning. Summarizing with respect to specificity, few of even the most basic of proposed system capabilities have been demonstrated, while many of the more ambitious possibilities like tuning the similarity metric or learning generalized plans are described abstractly.

In his work on **derivational analogy**, Carbonell (1983b, 1986) extends previous efforts at problem solving in transformation space to include complete derivational traces for previously solved problems (see Figure 7). This information includes subgoals, alternative operator choices at each step along with reasons for decisions among them, beginning and terminal (failed) nodes for unsuccessful paths with reasons for failure, interdependency linkages between successive decisions, pointers to peripheral knowledge structures used during problem solving, and the reasons and/or assumptions associated with successful or unsuccessful problem solving episodes. According to Carbonell, this information is necessary for successful transformation of retrieved solutions into novel solutions, particularly in complex task domains. An example of coding quicksort in LISP having previously coded quicksort in PASCAL is given as evidence for the idea that useful analogical transfer must

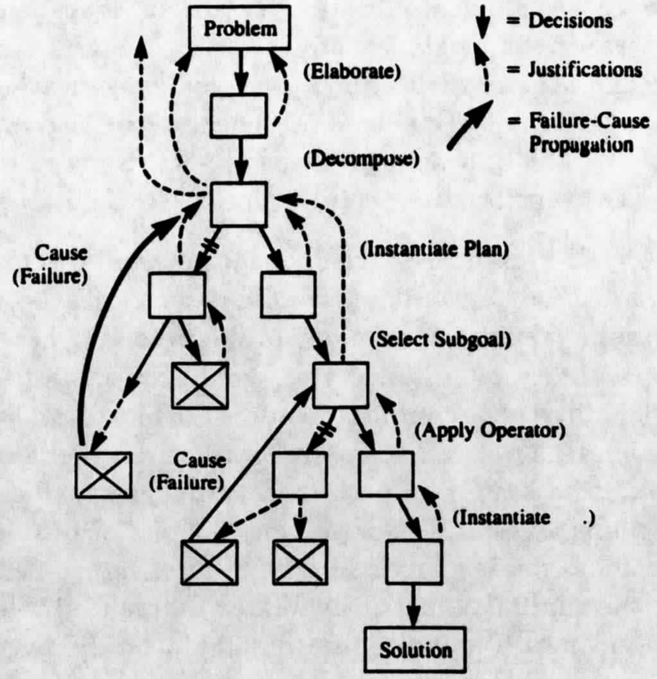

Figure 7
Derivational trace for a solved problem (Carbonell, 1986).

occur at a more abstract level and include a reconsideration of decisions made in the source problem (e.g., design of the PASCAL program).

Given a target problem, problem solving by re–playing derivations is somewhat complex. If direct plan instantiation is not possible, the system applies traditional weak methods (i.e., search). If early reasoning resembles initial segments of a derivational trace in dynamic memory, the system attempts to apply the retrieved derivational trace. Replaying a derivation proceeds as follows:

1. For each step in the trace, if reasons/assumptions for that step stored in the derivation hold in the target problem's state, that step is applied directly.
2. If some reason is violated but an alternative justification for the derivational step can be found in the new problem description, the step is applied and a record of this deviation from the retrieved derivation is stored for later inductive activities.
3. If attempts to justify a derivation step fail, a variety of activities are suggested:
    a. attempt to justify available alternatives stored in the derivation,
    b. post the unsatisfied justification as a subgoal for further problem solving,
    c. try paths marked as failures in the derivation if the reasons for those failures do not hold in the target problem,

d. give up the current analogical derivation in favor of another source derivation or traditional weak methods.

Carbonell points out that minor deviations from derivational traces are not sufficient cause for abandoning the derivation, since subsequent derivational decisions and steps may be independent and still apply to the target problem. Finally, a "perseverance threshold" is mentioned for interrupting fruitless derivational efforts. Carbonell argues that the processes sketched above can be implemented efficiently because space requirements are proportional to the depth of the trace rather than to the number of nodes visited while solving the source. In addition, few dependency links point outside the local structure of the derivational trace.

Opportunities for learning are again legion. First, more problem solving experience results in a richer database of case derivations. Second, stripped-down solution traces for successful and unsuccessful problem solving efforts can be used as input to a general purpose "induction engine" to form generalized plans. This is essentially similar to the primary learning mechanism proposed in the transformational approach. Third, various decision points encountered when re-playing a derivation can be used in a similar fashion as positive and negative instances for induction. According to Carbonell, justifications for choice points serve to focus the learning process on "functionally relevant aspects" of the derivational trace. The result would be domain-specific search heuristics. Finally, a "decomposition" process is offered as a mechanism for extracting meaningful subsequences out of derivational traces. Frequently occuring subgoal sequences are extracted, justifications for these subgoals are assembled (e.g., satisfying preconditions of a "supergoal"), and rules are assembled. The assembly phase is complicated by finding computable predicates for preconditions of the rule which test for assembled justifications.

Given Carbonell's report that "ARIES ... turned out to be far more complex than was originally envisioned" (p. 375), it is not surprising that most aspects of derivational analogy remain open research areas. The movement from transforming solution traces to re-playing entire derivations does little to make this approach to analogical reasoning or learning by analogy more specific. Recognizing a source derivation amounts to finding parallels between "initial decisions made and the information taken into account" (p. 380) for source and target problems. Elaboration and evaluation of these parallels is more encouraging, with explicit justifications (dependency linkages) stored in the source derivation serving as problem solving goals within the target problem. Consolidation depends upon a set of mechanisms which span the bulk of current research in machine learning. In summary, Carbonell's theory of reconstructive problem solving probably functions most effectively as a long range research proposal. Accomplishing any component of the proposed system would amount to a major contribution.

5. Metaphor and analogy in natural language processing

Given the bulk of related research in linguistics, philosophy and psychology, it is not surprising that some attention has been given to metaphor and analogy in computational approaches to language understanding. In this section, a variety of studies will be described, starting with approaches which attempt to derive meaning representations for isolated metaphorical utterances in and out of context, and ending with approaches which give metaphor a wider role in language and cognition.

5.1. Winston [1978] Concept learning by creatifying transfer frames

As somewhat of a departure from earlier work on learning from examples, in this paper Winston is interested in learning through "absorb(ing) simile–like instruction" (p. 151). Although the intent of this work appears to be that of examining a highly constrained form of learning, the proposed mechanisms are perhaps more interesting as a computational model of making appropriate inferences when presented with simile–like statements (e.g., Robbie is like a fox). In overview, Winston develops the notion of understanding similes as the construction of "transfer frames" which allow the projection of particular slot/value pairs from a source frame (e.g., fox) into a less well–understood target frame (e.g., Robbie). Of crucial importance in this transfer process are constraints which arise through *salience* of particular aspects of the source frame and its conceptual siblings, *prototypicality* of particular aspects of the target frame and its siblings, and the *instructional context* imposed by a teacher in presenting similes or statements of fact to the learning system. As a representation, a relatively simple frame language is described with a finite set of objects and properties for two microworlds consisting of blocks and animals, respectively. The "computational problem" proposed by Winston is that of filling slots in representational descriptions of new entities on the basis of analogical similarity with other descriptions available to the learner.

Winston describes four stages of processing, each of which has apparently been implemented:

1. *hypothesize* which of the possibly many source characteristics might transfer to the destination based on a set of progressively weaker strategies capturing *salient* aspects of the source frame;
2. *filter* the resultant candidate "transfer frames" using prototypic knowledge of the target frame or the ongoing instructional context (i.e., the sorts of properties involved in recent similes);
3. *justify* the selected (filtered) transfer frame by appealing to external assistance or noting important similarities between source and target frames (e.g., aspects supporting the purpose of objects);
4. and *conjecture* further possible transfer either on the basis of "curiosity" of the learner or encouragement from the instructor.

In comparison with processing components used in this paper, hypothesize and filter roughly correspond to elaboration, justification to evaluation, and conjecturing to peripheral aspects of consolidation. A collection of strategies is described in some detail for each of these process stages.

For hypothesizing transferable properties, Winston advocates using existing transfer frames previously constructed when reasoning from a source object, providing that the justification (stored as a set of distinguished properties in a frame) for previous transfer can be shown true of the current target frame (i.e., similar distinguished properties exist). If unable to identify existing transfer and justification frames, the system selects source properties (slots and/or values) which are "salient" by virtue of having extreme values, being globally important (i.e., a particular property is marked as important in an absolute sense), being atypical in comparison with conceptual siblings of the source frame, or having atypical values in comparison with siblings. Should these strategies for hypothesizing transferable properties fail, all properties of the source are considered. Candidate properties for transfer are grouped by similar property category (e.g., size, weight, height, width and depth are all instances of the size–property category) and considered independently by the filtering process.

Filtering strategies attempt to make use of prototypical aspects of the target frame in selecting which properties to transfer (i.e., which transfer frame to select). Winston describes the statistical construction of "typical–instance frames" which record common slot and/or value properties for a class of related frames (e.g., a typical cylinder has a color slot which takes on varied values in over 65% of cylinder instances). Transfer frames are preferred which promote slot/value pairs that typically occur in frame descriptions which are conceptually similar to the current target frame. In the absence of prototypical knowledge, transfer frames are preferred which promote properties contained by any conceptual siblings of the existing target frame. Finally, in the absence of any knowledge of what is typical of the target frame, transfer frames which continue the instructional context (i.e., the type of properties recently transferred on the basis of instruction) are preferred.

Justification strategies are somewhat less ambitious. Either the learning system asks the instructor directly about the appropriateness of a candidate transfer frame, or the system checks for transferable values which violate known restrictions on transfer frame slots. As a third and somewhat more challenging strategy, the learner notes what shared aspects of frames involved in a previously successful transfer were particularly important for the suitability of that transfer. Winston gives an example in which the instructor presents the simile, "CUBE–1 is like TABLE." The property of "purpose" being a platform for eating or writing is transferred from table to cube, and at the suggestion of the instructor a "justification frame" is formed which captures common characteristics of the cube and table which support their purpose. In this case, being of medium size and having a flat, level top are justifying properties for the purpose of being a platform for eating or writing. Subsequent instructor–initiated transfers involving TABLE can use this justification frame in confirming that a new target object would have properties supporting the transfer of information regarding purpose.

Strategies allowing independent conjectural learning by the system are described by Winston in the least detail. As described, the system is allowed by the instructor to occasionally amplify on recently acquired information in a particular target frame. Three mechanisms are proposed for this sort of exploration. Since the system has an explicit frame representation for slot types (e.g., TOP–FLATNESS) which includes a record of extreme values encountered while absorbing instructor–generated similes (e.g., VERY–HIGH for flatness when TABLE is used as a source), the re-occurrence of extreme values can serve as a trigger for system generated similes (e.g., a new object with VERY–HIGH TOP–FLATNESS is like a TABLE). Alternately, when adding a new slot/value pair to a current target frame, the system can look for conceptual siblings which have a similar property, generating a new simile with a selected sibling. Finally, Winston describes the learner as being able to fill in (actually, to inherit) properties for a target frame for which a typical instance frame is available.

In terms of completeness, Winston's system addresses each of the four components of analogical reasoning. Treatment of recognition is probably least satisfying in that similes are generally presented directly to the learner. However, some aspects of conjectural processing as discussed in the preceding chapter are suggestive of how similes might be recognized and use in arbitrary target frames without intervention of the teacher. Elaboration and evaluation processes are clearly addressed with proposals for hypothesizing and filtering potential transfer frames. Consolidation is accomplished both by a straightforward recording of successfully transferred information between frames and by the construction and later use of transfer and justification frames.

As described, computational mechanisms in this work seem of general applicability providing that a domain "world" is describable within the confines of a limited vocabulary of objects and their properties. Although Winston claims that domains expressed as "simple physical world(s)" are manageable within the proposed framework, one might wonder if the depth of knowledge in such domains is accurately reflected in the relatively simple object and property hierarchies used as examples in this paper. In fact, it seems likely that the presence of a particularly benevolent instructor would become increasingly important as one increased the complexity of the task domain and the scope of existing knowledge available to the learner. However, taken as a study of metaphor comprehension, Winston's model is interesting in its explicit use of salient characteristics of the source domain and prototypical qualities of the target domain. These ideas parallel psychological approaches to metaphor comprehension discussed in earlier sections.

5.2. Kilpatrick [1982] An A-frame model for metaphor

Kilpatrick presents a frame–based model of metaphor comprehension for novel metaphors in everyday conversation. As a representation, frames with terminals taking typed values are augmented with prototypic and stereotypic bundles containing, respectively, criterial exemplary characteristics of the referent and culturally shared information regarding the referent. According to Kilpatrick, stereotypic knowledge is of particular importance in metaphor comprehension since the hearer may know this information is empirically incorrect but still use it in constructing an interpretation. A short description of metaphor production involving search among candidate stereotypic bundles is mentioned, followed by a fuller account of metaphor comprehension.

Given a metaphorical utterance, comprehension proceeds by a recognition that prototypical knowledge of the target is incompatible with knowledge relating to the stated source. An example statement, "The hearings are a blunderbuss," is presented in which a blunderbuss (the source) cannot be understood as an instance of hearings (the target). In an attempt to resolve this "metaphor parasite," the blunderbuss frame is activated and placed in correspondence with the hearings frame. This produces an "A–frame" (pictorially denoted by frames leaning against each other) consisting of the union of target and source stereotypic bundles, with the remaining frame information (apparently including prototypic knowledge) held in readiness for extension of the metaphor. It is on the basis of juxtaposed stereotypical knowledge, according to Kilpatrick, that metaphors are comprehended quickly and easily by hearers. Unfortunately, the details of comprehension are given in a series of questions (of unknown source) which, as the comprehension process attempts to find answers, guide the elaboration of a metaphorical ground between target and source. For example, "Are the hearings accurate?" prompts a search for an accuracy value, contributed by the blunderbuss stereotypic bundle. Ambiguity arises when knowledge of target and source suggest differing answers, but Kilpatrick argues that contextual information can be used to resolve such ambiguities.

Although Kilpatrick's work seems generalizable beyond the confines of comprehending conversational metaphors, several of the proposed processing mechanisms lack specificity as described. Kilpatrick mentions recognition of metaphors in the context of intentionally producing them, but the search for stereotypic bundles is not elaborated beyond suggesting that frame terminals in the target could guide such a search. In addition, Kilpatrick's example of knowledge stored for hearings and blunderbusses is reassuringly sparse, concealing the likely fact that a tremendous amount of knowledge would result from the juxtaposition of frames. How particular correspondences between components of this knowledge could be uncovered or how the hearer's attention might be focused on particular aspects of this knowledge (i.e., attention to salient elements) is not addressed. Finally, issues of consolidating the results of metaphorical reasoning are not addressed.

5.3. Hobbs [1983a, 1983b] Metaphor as selective inferencing

Hobbs offers a natural language understanding system, DIANA, as a computational model of how textual (in context) metaphors can be understood without resorting to processes apart from ordinary discourse analysis. The system takes as input predicate calculus formulas (generated by syntactic analysis of natural language text) representing textual material. Hobbs assumes a general knowledge base of axioms (also in predicate calculus) describing commonsense and linguistic knowledge. He argues that in the process of resolving typical discourse analysis problems like pronoun reference, noun relations, and textual coherence, appropriate inferences necessary for the comprehension of textual metaphors will also be collected. The following is given as an example of a textual metaphor: "We insist on serving up these veto pitches that come over the plate the size of a pumpkin." Hobbs finds special significance in metaphors involving domains for which the hearer possesses extensive knowledge (i.e., a large set of axioms), such as spatial metaphors. A domain is defined as a richly interconnected set of predicates and axioms.

Apparently, source domain knowledge is triggered by the appearance of predicates in the text (e.g., the word "pitch" invokes knowledge of baseball), and correspondence between domains is established on the basis of contextual constraints imposed by ongoing discourse. Correspondence for relatively basic concepts between domains (e.g., Congress/pitcher) allows the instantiation of more complex concepts or relationships (e.g, easily hit slow pitches). Such inferences are "selective" in the sense that they are guided by context to favor intended (by the speaker) inferences while discouraging inappropriate inferences (e.g., that bills/balls are round with stitching). Hobbs alludes to the particularly powerful role which more neutral inferences (neither contextually intended nor inappropriate) may play in increasing knowledge of the target domain. The consolidation of metaphorically imported knowledge is obliquely described in several stages intended to capture the ontogeny of a metaphor as a linguistic phenomena. Originally, a metaphor must be explicitly decomposed into simple and complex concepts which are placed in correspondence through considerable computation. At a subsequent appearance, the more familiar metaphor can be comprehended quickly since appropriate inferences are expected. Eventually the metaphor may become "tired" in the sense that direct links have been acquired between simple and complex concepts in the target domain, resulting in comprehension without explicit transfer between domains. Finally, imported knowledge has been so well integrated into the knowledge base that metaphoricity can no longer be recognized: the metaphor has become frozen or dead.

There are a number of problems with Hobbs' description of metaphor comprehension despite the overall acceptability of his specifications which are quite in keeping with views of metaphor outside of AI. In terms of completeness, Hobbs discusses issues of recognition rather briefly, suggesting that on the basis of distinguished sentential components (e.g., pitch) appearing in the textual input stream, appropriate domains of knowledge are brought into play for comprehension. However, it is not clear how such a mechanism would choose among the multiplicity of word senses available for a single reference (e.g, a sales pitch or a tar–like substance) or how knowledge sources activated by non–salient words (e.g., "serving up" or "plate" with the supporting evidence of a consumable "pumpkin") could be suppressed. Furthermore, given that a salient source domain has been recognized and retrieved, Hobbs gives little suggestion of how "appropriate" inferences are to be selected within that domain. For example, "inappropriate" inferences in the baseball domain involving ticket sales, umpires or hotdogs are not distinguished from the "suggestive" inference involving adversarial games, an inference which Hobbs uses as an example of "metaphor's greatest powers." In short, if selective inference is to be guided by textual context, a somewhat more powerful use of context should be forthcoming. As a final note on completeness, Hobbs treatment of consolidation is only suggestive, without description of how the knowledge base of axioms is to be updated such that metaphors eventually become "frozen." In terms of performance and generality, it is not clear what aspects of the proposed competence in metaphor comprehension have been demonstrated with DIANA, and hence it is difficult to tell to what extent the proposed mechanisms (with attendant shortcomings) would be applicable to a broad range of metaphorical utterances.

5.4. Dyer [1983a. 1983b] Adages and planning failures

Dyer describes an implemented system, BORIS (1983a), and its successor, MORRIS (1983b), which are proposed as cognitive models of in–depth understanding of short but fairly intricate narrative stories. Noticing that many such stories involve failures in expectations arising from faulty planning by the characters involved, Dyer introduces the concept of "thematic abstraction units" (TAU's) which serve to organize related narratives involving similar planning failures for subsequent retrieval. This form of memory organization, according to Dyer, accounts for an important class of cross–contextual "remindings" in which people often spontaneously produce an adage summarizing the planning failure involved in the narrative (e.g., being caught "red–handed" or remarking that "every cloud has a silver lining"). Dyer (1983a) describes experimental evidence that subjects given narratives exemplifying a particular TAU are later able to generate literally unrelated stories which exemplify the same sort of planning failure. Such stories are then sorted by a different group of subjects in a manner which preserves abstract planning similarities despite wide variations in story content. Although it is not Dyer's primary intent to examine aspects of metaphor comprehension, storage and retrieval of narratives based on abstract planning information could provide a plausible account for some forms of metaphorical reasoning. Such planning information (unexpected goal failure or satisfaction is also discussed), shared in common between a current narrative and some previously encountered story, clearly resembles the sorts of knowledge implicated in Gentner's mapping of structural relations (1983a) or Carbonell's transfer of invariant information (1981). August and Dyer (1985) describe preliminary work on understanding analogies in editorial arguments. Analogies are recognized on the basis of textual clues (e.g., "is similar to") or similarity of textually contiguous conceptual structures. Corresponding causal structures for target and source are placed in correspondence using "comparison links," which later guide inferences during a question answering task.

Dyer's work is probably most pertinent for computational approaches to metaphor comprehension in its focus on memory organization and retrieval of analogous narratives. This aspect of Dyer's work appears to be an extension of Schank's interest in reminding and dynamic memory organization (1982), although the precise relation between TAU's and previously proposed memory structures (e.g., MOP's, meta–MOP's, and TOP's) remains rather vague. Dyer claims that TAU's are somewhat less abstract than Schank's TOP's (thematic organization packets) which are advanced as domain–independent organizers of memory on the basis of general goals and the conditions which attend those goals. As an example of a TOP, remembrance of choosing a compromise restaurant when two parties want to eat different kinds of food might be associated with a "Competing Goal; Compromise Solution" TOP. Later when rearranging an appointment with someone who can only meet at an inconvenient time, one of these parties might be reminded of the restaurant compromise. Thus while the primary organizational dimension for TOP's appears to be that of goal interaction, TAU's provide organization at the somewhat less abstract level of planning difficulties. To use an example from Dyer (1983b), when watching a friend grow increasingly upset about the possibility of some automobile repairs not being completed, Dyer claims to have told the friend to "quit bleeding before he'd even been cut" and to have been spontaneously reminded of a story in which a distressed and apprehensive motorist angrily refuses help from a farmer before even explaining his situation. Both of these events are instances, in Dyer's terms, of a "TAU–EMOT–ANTICIPATE" in which anticipation of a negative situation disrupts appropriate planning about what to do next.

Viewed from outside the context of understanding complex narratives, this work could easily be described in terms of the recognition, elaboration, evaluation and consolidation of analogical relationships between experienced events. Recognition, given a memory organized in terms of planning failures, occurs as planning mistakes in a new narrative are used as indices into some appropriate TAU which organizes memories involving similar planning errors. Planning information stored in this fashion can be instantiated within the new narrative and serve as a means of predicting what may come next in the narrative. To the extent that differences occur between new narratives and existing accounts of previous experience, memory structures may be reorganized so that new narratives are consolidated in memory for subsequent retrieval. Issues of recognition and consolidation, without concentration on adages reflecting planning errors, are discussed at length in Lebowitz's (1982) account of constructing and evaluating generalizations based on limited information and Kolodner's (1983a, 1983b) description of maintaining a long–term episodic memory.

6. Analogical reasoning in machine learning

Although many of the studies reviewed thus far touch on "learning by analogy" in some fashion, each can be seen as an instance of a computational approach to analogical reasoning within a larger context. The decision about where to place individual studies is somewhat arbitrary. For example, studies by Carbonell (1983a, 1983b, 1986) certainly address learning issues, but they provide an exemplary view of analogical reasoning as a problem solving process. In this section, four projects which explicitly focus on learning by analogy are discussed. Winston's work on learning principles from precedents and exercises (1980, 1982, 1983) is discussed first. Second, Burstein (1981, 1983, 1986) presents a cognitive model of integrating multiple analogies drawn from different levels of abstraction for learning simple concepts in a programming language. In the third project, Anderson, Farrell and Sauers (1984) describe a cognitive model of skill acquisition for learning to write list manipulation programs. Finally, Kedar–Cabelli (1984, 1985) proposes a model of purpose–directed analogy, in which an explanation for how a source artifact satisfies some reasoning goal can be used to focus the attention of a learning process. As will become clear, the role for analogy in learning is open to dispute.

6.1. Winston [1980. 1982. 1983] Learning and reasoning from analogous cases

In these papers, Winston describes an implemented model of analogical reasoning in which constraints from a source situation (termed a precedent) are transferred into a target situation (termed an exercise). Shared constraints enable reasoning in the target, and this reasoning provides basic materials out of which general rules can be learned. Simulation results are presented for several domains including plot narratives (e.g., Shakespeare's plays), medical cases, and physical systems (e.g., electrical circuits). For example, having given plot summaries for a number of plays, a teacher might give limited information about a target case and ask the system to answer a question about the target (e.g., show that a man is weak). A source plot summary is recognized as being applicable and retrieved (e.g., *Macbeth*). A constraint-directed matching process uses a relatively complex, summative similarity metric to select the best of a set of candidate matches between target and source. Questions asked by the system to confirm expected facts may or may not be answered by the teacher. Constraints which would allow an answer to an analogous question in the source are transferred to the target, supporting an answer to the target question (e.g., a weak man with a greedy wife is likely to be evil) The system delivers this solution, annotated by sources of evidence contributing to the solution. These may include facts transferred from the source case. Finally, the system induces a general rule over these two cases, and that rule is stored for later use in similar circumstances.

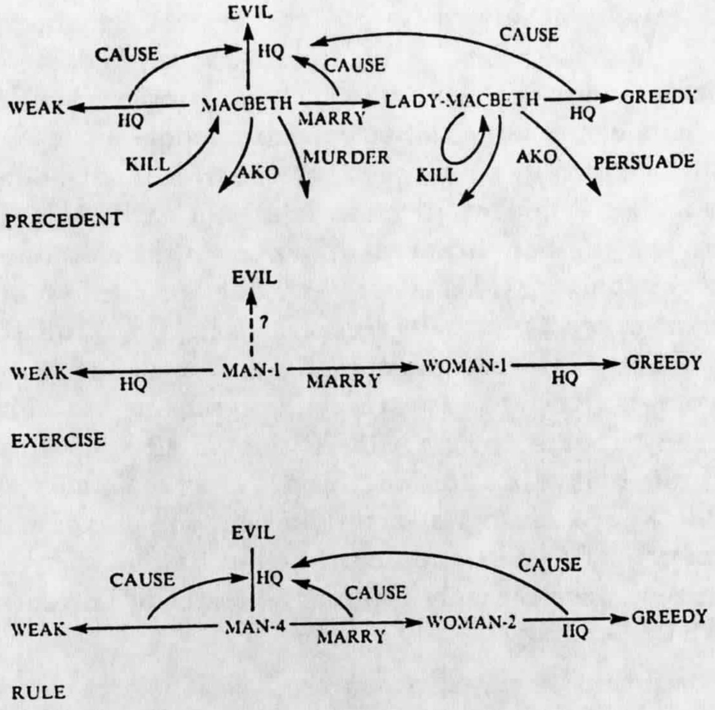

Figure 8
A precedent, exercise and resulting rule (Winston, 1982, p. 331)

Knowledge of situations for a given domain is represented in object–oriented (rather than event–oriented) propositional networks. These are augmented with supplementary descriptions of constraining information (e.g., causality). Figure 8 shows a graphical depiction of the source (precedent), target (exercise) and rule as described in the preceding paragraph. The target problem (? MAN–1 EVIL) can be answered on the basis of constraining relations imported from the source (e.g., the three causal relations supported by MAN–1 being weak and being married to a greedy spouse). Inputs (from a teacher) to the system include structured natural language descriptions of situations, questions concerning relations among objects (these are target problems), answers to system generated questions concerning values or relations in the situation, and suggestions that the system calculate values or form laws. Outputs include answers to a variety of questions posed by the teacher along with justifications for those answers, more richly detailed network representations for new situations as a result of analogical transfer, and newly formed general rules.

The work described by Winston in these papers explicitly proposes mechanisms for recognizing analogous situations, elaborating an appropriate match between source and target situations, transferring knowledge from source to target, and consolidating the results of that transfer in general rules. No mechanism for evaluating the usefulness of generated analogical mappings is proposed, other than allowing a solution to the given problem. Recognition, elaboration and consolidation will be described here in some detail.

Recognition of source situations is accomplished by "classification-exploiting hypothesizing." This is essentially a bottom–up retrieval scheme which exploits an exhaustive indexing of candidate sources through a type hierarchy of objects known to the system. "Votes" for a particular source are collected by traversing a–kind–of (AKO) links originating with components of the target situation. Each node in the AKO hierarchy has an APPEARS–IN slot which records all of the situations which that node and its descendants participate in. For each AKO node visited, candidates in APPEARS–IN slots cast votes in inverse proportion to the number of candidate sources in the slot and in direct proportion to the size of the target concept. This latter weighting is a crude reflection of the relative importance of different concepts in the target situation (e.g., "Juliet" is a concept in the story of Romeo and Juliet). After a complete AKO traversal, the source situation with the highest number of votes is retrieved. This recognition scheme has some possible problems, however. Including fully updated APPEARS-IN slots in all nodes of various AKO hierarchies could become quite bulky. In addition, it is not clear that this strategy would support recognition of genuinely novel analogies across different domains. For example, an analogy between current in electrical circuits and depression in Hamlet's psychological state would not be easily recognized, since the only common ancestor of current and depression may be THING in the AKO hierarchy.

Having recognized and retrieved a source situation, a match must be elaborated between this source and whatever is known about the target. Using exhaustive search through the space of possible network matches, Winston claims the matching process is confined to problems of 100 or fewer possible pairings. In the worst case, this restricts the system to matches over situations having five or fewer objects in each situation. In practice, it appears that Winston carefully chooses initial points of correspondence, preferring maximally constraining relations (e.g., cause) that can be compatibly aligned. Thus the space of possible matches between source and target is enumerated using some heuristic constraints (see Winston, 1984). For a candidate match, Winston (1980) proposes an elaborate scoring process which rewards correspondence of object properties, relations among objects, and constraint relations. A variety of similarity metrics are considered, and empirical results with plot summaries suggest that scoring only constraint relations known to be important *a priori* can be as effective as more complicated scoring schemes.

Eventually, a best match between situations is selected, and the system proceeds to the original task of answering a question (i.e., an unknown but desired relation) about the target. Question answering proceeds in three steps:

1. If the desired relation is available in the target, simply use it to answer the question.
2. If the desired relation is missing in the target but caused by other relations in the source, try to establish those causal antecedents in the target situation. This essentially produces another question to be answered (i.e., a subgoal).
3. If the desired relation is missing in the target and has no known cause in the source, look for another source. Multiple sources (precedents) may be strung together in searching for an answer.

This process is somewhat complicated since Winston allows abductive inferences by analogy: a desired relation in the target may be inferred if its known consequences (as shown by the source) are all known to have occured. Thus the desired relation is "covered" by its consequences. In some cases when step (3) above is reached, the system directly asks the teacher (Winston) to confirm or disconfirm desired relations. In general, questions concerning unknown relations in a target situation can be answered by analogically inferring relatively complex causal chains available in the source situation. These causal chains can be expressed as AND trees, with the desired relation at the root and a sufficient set of facts at the leaves.

AND trees provide the basic materials for consolidation in Winston's system: teacher initiated reasoning is summarized and remembered by learning rules. Target and source AND trees serve as instances for an inductive process which extracts common implicational structure from the trees and finds general terms for the selected nodes. The result is also an AND tree in which leaves make up the conditions of the rule, and the root node provides the action or consequence. Intermediate causal links are dropped. The resulting rule in the simple case described above might be expressed more conventionally as follows:

```
Rule    RULE–1
if      [MAN–4 HAS–QUALITY WEAK]
        [MAN–4 MARRY WOMAN–2]
        [WOMAN–2 HAS–QUALITY GREEDY]
then    [MAN–4 HAS–QUALITY EVIL]
case    MA
```

Note that the source of this rule, MA, is retained so that some analysis is possible should the rule later fail. Winston suggests that falling back on the appropriate precedent may allow case–based reasoning when learned abstractions (rules) are insufficient. Just as a new rule may be learned from an analogy between situations, new rules may also result from applying an old rule in a new situation.

Since general rules acquired in the above fashion may be overly general, these rules are further augmented (Winston, 1983) with "unless" conditions which can block the rule should exceptional conditions arise. Augmented rules are of the form:

IF [preconditions ...] THEN [conclusion] UNLESS [exceptions ...]

Exceptions include negations of relations between conditions and conclusion in the AND tree out of which a rule has been synthesized as described above. Thus, exceptions are limited to causal relations of known causal relevance. When an augmented rule is considered for use, a limited amount of effort is put into showing that no unless conditions hold. If an exception holds, the supporting causal structure of the rule is violated, and the rule will be blocked. Exceptions may be concluded by other rules, termed "censors." Censor rules have the same form as other augmented rules, and may be learned and used accordingly.

Thus augmented rules serve as abstractions of knowledge gained by transferring information from precedent to exercise situations, and may subsequently provide needed information without resorting to full analogical reasoning between precedent and exercise. To facilitate retrieval, rules are indexed through the classes of actors, relations and objects found in their consequences (in this order). Indices are stored in multi-level association lists residing in frame structures which describe these classes. Given a new target question, components of the question are used as retrieval cues to identify sets of candidate rules. These candidates must further match the target situation before being used to find an answer.

In terms of the process components used in this survey, Winston model of reasoning and learning by analogy fares well. Acquired rules both summarize analogous cases and provide a mechanism for recognition and retrieval. Candidate source situations are recognized in a similar, though possibly less precise, fashion. Both approaches would encounter difficulties in recognizing analogies across widely disparate domains. The matching approach described in earlier reports appears applicable to rules as well as precedents. In fact, Winston argues that censors might be used to interrupt attempts at fruitless analogical mappings by recognizing when necessary relations in a candidate source are implausible. Elaboration occurs as a completed match is used for placing aspects of source and target in correspondence, within the confines of a given problem. Since the match is calculated before attempting to solve the problem, evaluation plays a static role, primarily in the form of a scoring function over competing matches. Through consolidation, rules become preferred sources of knowledge which take precedence over computationally more expensive precedents. When a known rule can be applied, the need for analogical reasoning is circumvented.

6.2. Burstein [1981, 1983] Learning assignment statements in BASIC

Burstein describes a computational model of learning by analogy for relatively simple assignment statements in the BASIC programming language. This model is based on extensive interviews with students learning to program, and is intended as a cognitively plausible account of learning in this domain. As a knowledge representation, interconnected networks of frame-like structures similar to Schank's notions for dynamic memory (1982) are utilized. Central to Burstein's discussion is the concept of a "top-down analogical mapping" of causal abstractions from a well understood base (or source) domain into the target domain of assignment statements. This is offered in contrast with "bottom-up partial matching" which Burstein claims is typical of Gentner's (1983) or Winston's (1980) approaches to analogical reasoning (e.g., start with node correspondence which maintains constraint relations). Specifically, Burstein argues that a bottom-up elaboration of an analogical mapping between two domains cannot be sufficient when little is known of the target domain or when the causal structure offered as a means of constraint can be decomposed into complex lower-level relational structure.

In Burstein's view, causal structure representing a common abstraction between domains (analogies are directly presented by a very helpful teacher) is incrementally extended into the target domain, facilitated by exemplary uses of these structures in the domain. As an example, Burstein might tell the system, CARL, that "a variable is like a box ... To put the number 5 in the variable named X, type 'X = 5' " (1983, p. 21). As a result, CARL attempts to map causal knowledge of preconditions, actions and results in the domain of boxes into the target domain of assignments to variables. As Burstein is careful to point out, such a mapping need not remain at a literal level with respect to relational information being transferred. Hence, since numbers are not objects in the sense of items usually placed within boxes, a new sense of the "INSIDE" relation must be found. CARL finds such an abstracted relation (e.g., INSIDE–VAR) by selecting an ancestor or sibling in a relation type hierarchy. In addition, relational information salient within the box domain (e.g., a precondition that the size of the transferred object fit within the dimensions of the box) but inappropriate because of role incongruity in the target domain is not transferred into the target domain. Hence, although analogical mappings are argued to be top–down, their elaboration appear to require relatively low level constraints (e.g., expected roles) originating within the target domain.

On the basis of analysis of protocols collected while tutoring novice BASIC programmers, Burstein claims that multiple analogies are regularly used in acquiring programming concepts. For example, in learning how to use the assignment operator, the following analogies often occur: a variable is like a box, a computer "remembers" numbers after an assignment statement, and assignment is like mathematical equality. In keeping with observed cognitive phenomena, the CARL system uses multiple analogies to incrementally extend and repair knowledge in the target domain. For example, while the box analogy may be acceptable for simple assignment statements with a single entity to the right of the equals sign, more complex arithmetic assignments (e.g., $X = B + 1$) often uncover errors in student's conception of assignment. As a concrete error, the expression "B + 1" is said to be placed inside the variable, X. Interestingly, precisely the same sorts of errors have been reported by Bayeman and Mayer (1983). In that study, 76% of novice subjects describe the effects of such a complicated assignment as placing an equation in the memory space for A, rather than transferring the appropriate value.

According to Burstein, these complicated assignments are better understood by a combination of algebraic knowledge (i.e., equality of value and algebraic operations like addition) with some notion of placing objects (a value in this case) within containers (a variable). In fact, Burstein argues that such an understanding would also include the incorporation of a third analogy, that of the computer as a "human information processor" capable of interpreting algebraic expressions and placing values within variables. It is in this fashion, as the teacher explicitly and implicitly presents the learner with a variety of analogies, that the learner incrementally acquires an effective conceptualization of the assignment operator. As a result of such a process, Burstein claims that the CARL system eventually develops semantic representations for common assignment statements, rules for how to parse them, knowledge of their effects, and knowledge of how to use assignment statements as "components of simple plans."

6.3. Anderson, Farrell and Sauers [1984] Learning to program in LISP

Although Anderson devotes some attention to processes of analogical reasoning in his earlier work (1981, 1983), he probably gives closest scrutiny to this facet of learning in the GRAPES simulation of learning to program in the language LISP (Anderson, Farrell and Sauers, 1984). GRAPES is intended as a cognitive model of learning, and is based on analysis of detailed protocols with three subjects. It is interesting that analogy is given a limited role in learning, occuring early in the acquisition process before a sufficient store of procedural knowledge has been compiled. Early problem solving (e.g., writing a LISP function to extract the first or second element of a list) proceeds by the use of "structural analogy to concrete cases." At this point, subjects tend to construct "mappings" between abstract templates given in a text or example functions (also given) and a new problem. Of general interest here is the finding that subjects appear to make very little use of verbal instructional materials. Instead they reason about similarities between concrete exemplars and the current task requirements.

To continue the example above, the subject presented with the task of writing a FIRST function engages in a process of attempting to instantiate (or "map" in Anderson's terminology) components of an abstract function definition template within the current problem specification. In this case, the template would be (DEFUN [function name][parameter list][process description]), where each bracketed element represents such a component. Errors committed during instantiation (e.g., inadvertently enclosing an argument to CAR in parenthesis) are corrected in a similar fashion by using "structural analogies" with other concrete exemplars. An exemplar might be a previous use of the CAR function for direct interpretation. Unfortunately, although analogical reasoning is given a role early in the acquisition process, it is far from clear how structural analogies are recognized, "mapped" or evaluated. Instead, Anderson simply states that a correct instantiation for some functional component is "solved by analogy" (p. 95) with a chosen exemplar. As various components of the abstract function template are considered, other concrete analogs are retrieved and used without explanation.

In terms of consolidation, GRAPES forms productions which capture central aspects of the instantiation process. In this exemplary case, two productions are stored: the first identifying the abstract template components and pertinent subgoals. In this case, the acquired rule requires writing DEFUN and setting a subgoal to "code the relation calculated by this function." (p. 96) The second rule specifies how to write an argument which is a variable local to the enclosing function; this production avoids difficulties with parenthesis encountered with the argument to CAR (described above). These productions and others like them are used in future problem solving episodes in a manner which obviates the need for continued use of structural analogies. For example, an analogy would not occur when writing a function to extract the second element of a list. Use of such compiled knowledge is reflected in subjects' rapid creation of a SECOND function. Proceduralization (i.e., constructing productions which record the manner in which declarative knowledge is used) and composition are described in a fashion consistent with earlier work (Anderson, 1981).

In summary, Anderson describes a limited and weakly specified use for analogical reasoning in the acquisition of programming skills. Part of the limited role for analogy in this task domain may be attributable to the fact that novice LISP programmers have little previous knowledge of programming techniques which might provide intra-domain analogs. Hence, they may not be able to generate useful analogies with knowledge domains outside of LISP programming. However, such a constrained role for analogical reasoning is not peculiar to Anderson's approach in this particular task domain. In earlier studies of geometry proof generation (Anderson, 1981) and language acquisition (Anderson, 1983), Anderson characterizes the role of analogical reasoning in knowledge acquisition in essentially the same manner. When attempting to prove geometry theorems, for example, subjects consistently refer back to previously solved problems in an effort to "map" portions of a previous proof into the current problem. Anderson argues that this sort of analogical reasoning is of limited effectiveness since memory for specific exemplary problems is short–lived. In addition, partial matching which controls recognition and use of analogies is overly superficial and sensitive to irrelevant aspects of problem similarity. Hence, analogies used under these circumstances may simply be misleading. In short, Anderson's approach to analogical reasoning in learning seems to refute the stronger sense of usefulness for analogy characteristic of many others in the field.

6.4. Kedar–Cabelli [1984, 1985] Purpose–directed analogical reasoning

Kedar–Cabelli argues that existing approaches to analogical reasoning provide insufficient constraints on *which* aspects of the source should be extended to the target. As noted by others in the psychological and AI literatures (e.g., Holyoak, 1985 or Greiner, 1985), contextually–relevant analogical inferences may be a small subset of the possible inferences which could be produced while considering how a source and target situation are related. Some studies of analogical reasoning either represent *only* relevant causal structure in advance of making analogical inferences, or give explicit hints concerning which structures in the source should be considered for elaboration. Kedar–Cabelli proposes a model of analogical reasoning which uses explicit knowledge of the *purpose* for which an analogy is being constructed to constrain the process of elaboration. In addition, the model of analogical reasoning is related to ideas about using explanations to guide learning (Mitchell, Keller and Kedar–Cabelli, 1986).

In overview, Kedar–Cabelli proposes a method for learning concepts about common physical objects (e.g., a cup or a vehicle) by using an analogy between specific instances of the concept: a given target instance and a retrieved source instance. For example, Kedar–Cabelli (1985) re–uses a situation originally described by Winston, Binford, Katz and Lowry (1983) in which the concept of a HOT–CUP used for drinking hot liquids is to be refined after seeing a new target instance, a styrofoam cup. Knowledge of a ceramic cup is used as the source instance. What is transferred in elaborating the analogy is an explanatory structure which justifies why the ceramic cup can be used to drink hot liquids. In an earlier paper (1984), Kedar–Cabelli discusses the problem of acquiring legal concepts by re–using justifications in a similar fashion. Kedar–Cabelli's model of purpose–directed analogy requires that several important pieces of knowledge be given as input. First, the concept being learned must be available. This concept would be sufficient for instances seen thus far, but may need to be changed to cover the target instance. Second, the purpose for which the desired concept will be used is given. In the example case, the purpose of a HOT–CUP is to enable drinking hot liquids. Third, a *domain theory* is given containing axioms which specify how physical attributes of objects relate to their functional roles. In the example case, an object's having a handle enables grasping that object. As final input, attributes of the target instance are given.

Learning concepts by analogy proceeds in five basic steps:

1. **Retrieve** a prototypical source which is a known instance of the goal concept. No sense of the term, prototypical, is described.

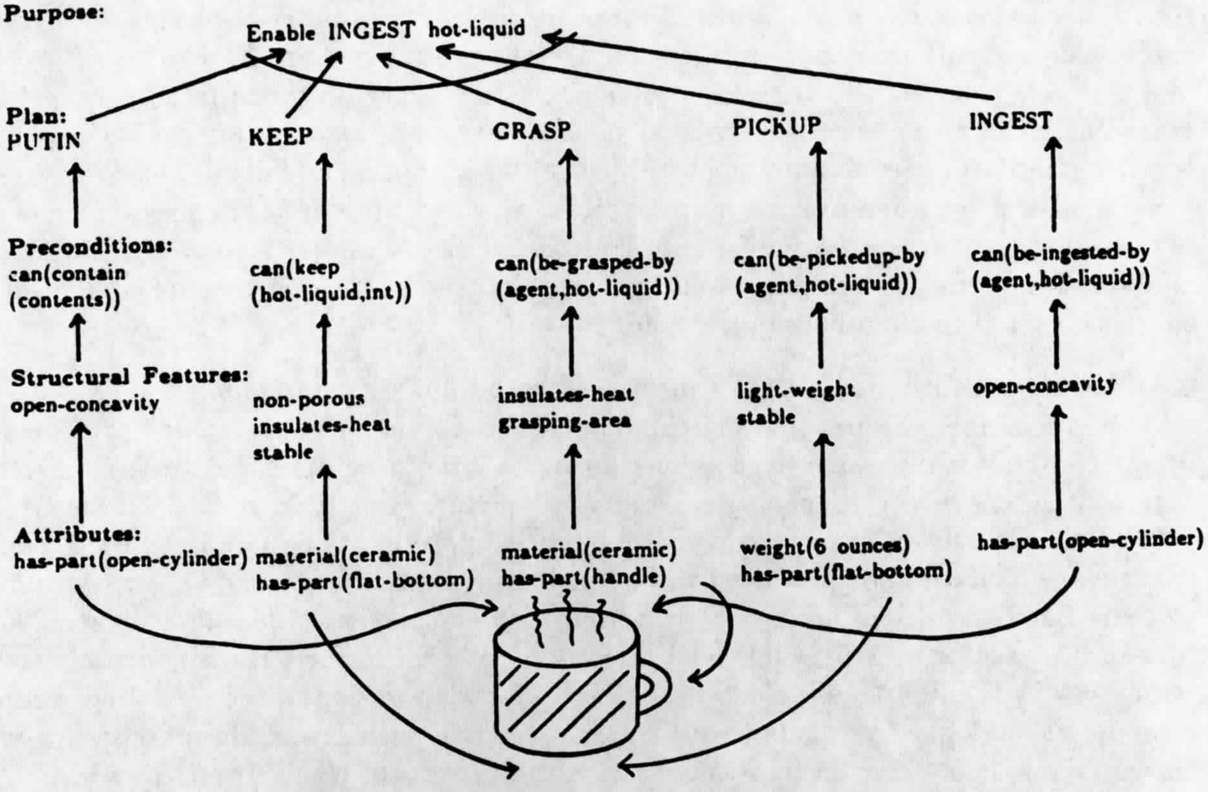

Figure 9

Explanatory inferences satisfy plan preconditions (Kedar–Cabelli, 1985)

2. **Explain** why the source instance is considered a member of the goal concept for the given purpose. Derivation of such an explanation requires several steps:

   2a. Find a plan which achieves the given purpose. For drinking hot liquids, a plan might be: to put the liquid in the container, to keep the liquid in the container and hot long enough to drink it, to grasp the container, to pick it up, and to drink the liquid.

   2b. Extract object related enabling conditions from the plan. These are called *object preconditions* – e.g., that the container object can be grasped. These are functional requirements which objects must satisfy if the plan from step (2a) is to be used.

   2c. Construct a network of "explanatory" inferences which show how structural characteristics of the source object enable functional requirements of the plan (see Figure 9). This network is deduced from domain theory axioms – e.g., having a handle and being constructed of ceramic material provides an insulated grasping area which can be used by an agent to grasp the cup. An adequate explanation enables the plan provided in step (2a) above.

3. **Map** (actually just copy) the explanation found in step (2) into the target.

4. **Justify** the explanatory inferences of step (2c) for attributes of the target instance. Justification starts with structural attributes of the target, but may eventually require

finding alternative structural attributes or even plan steps which are effective for satisfying the stated purpose for the target object. For example, the styrofoam cup is graspable by virtue of having a conical shape, rather than a handle.

5. **Learn** or refine the goal concept. This can be accomplished by simply recording justifications for source and target objects, or by inducing a characterization over both objects treated as positive instances.

In the ceramic/styrofoam cup example, the resulting concept specifies that a HOT–CUP "can have an open concavity, can be made of nonporous, insulating material, can be stable, lightweight, and can be graspable." (1985, p. 154) This is a sufficient characterization of HOT–CUPs formed by finding a common sub–network of explanatory inferences within each of the instance explanations. Note that the "graspable" attribute is a functional rather than structural term, resulting from the absence of a handle in the target instance (styrofoam cup).

In terms of process components discussed in this survey, Kedar–Cabelli's model makes a primary contribution to processes of elaboration and consolidation. Rather than considering any plausible analogical correspondence between source and target, attention is given only to information which provides an explanation for why the source instance satisfies the learning purpose. Thus, in the example shown above, only structural features which are salient for using the ceramic cup to drink hot liquids are considered, rather than allowing inferences about where the cup was purchased, whether the cup was given as a gift, etc. However, in order to use purpose–relevant features of the source, the retrieval and plan finding processes (steps 1 and 2a, above) must first provide the source instance and plan. Neither of these capabilities are described.

Consolidation provides the surrounding reasoning context in this model: analogies are used to learn new concepts or to refine old concepts. In either case, a new concept description is to be induced by finding a common inference network between explanations for source and target instances. Unfortunately, it is not clear in the model how learned concepts are to be stored or subsequently used. Neither is it clear how learning from pairs of specific instances could be accommodated within more traditional incremental techniques (e.g., Mitchell, 1982), unless the retrieved "prototypical" source carries an inductive summary of observed instances.

For recognition and evaluation, the purpose–directed analogy model contributes less. Recognition is essentially given by providing the system with a goal concept from which a descendant will serve as the source instance. Recognizing an appropriate plan through which this source object can be used to achieve the given purpose (e.g., drinking hot liquids) is not described. Evaluation of an explanatory network mapped from source to target is described briefly and depends strongly upon the contents of the given domain theory. Axioms within the theory which yield structural or functional consequences (e.g., a flat–bottom attribute provides structural stability) must be *a priori* sufficient not only for deducing a source explanation but also for justifying that explanation with respect to the target. Justification is an evaluative process which, as described, must either be able to deduce an appropriate target explanation (as mapped or newly created) or find alternative plan steps. Thus it appears that the results of analogical transfer, as justified in the target, should be inferrable from existing knowledge sources without assistance of analogical reasoning. Compared with traditional techniques, then, the utility of analogy in reasoning or concept learning becomes an empirical question. At a more theoretical level, one might argue that no learning occurs at all since all inferences (whether analogical or otherwise) must be derivable from existing knowledge.

In summary, purpose–directed analogy provides a relatively specific solution to a very general problem: constraining the set of plausible analogical inferences so that only *relevant* inferences are considered. Constraints are imposed by insisting that information considered for transfer come entirely from an explanation of how a source object satisfies an explicitly given reasoning purpose. Purpose, expressed as a goal associated with the desired concept, leads to selection of a plan which focuses the explanation process. A given domain theory supports derivation of the source explanation and later justification of that explanation for the target. Re–using the source explanation for the target may increase the efficiency of explanation–based inductive mechanisms which must construct new explanations for each instance observed. Finally, the notion of *purpose* as a constraint remains vague. Much of the constraint placed on the elaborative process of building an explanation for the source appears to come from the plan selected as a means of achieving the purpose. In fact, knowing what a target object will be used for seems in principle little different from knowing what the solution to a problem would look like (e.g., a state description and path constraints in Carbonell's work) or what question is being asked in an exercise situation (e.g., Winston's MACBETH system described previously). In Kedar–Cabelli's work as well as these others, obtaining plausible analogical inferences may still depend most heavily upon recognizing and retrieving a useful source.

7. Analogical reasoning in review

In this section, discussion shifts from specific contributions of various approaches to analogy to a more general examination of the areas of commonality across studies with respect to general process components of analogical reasoning. A patient and motivated reader of both papers in this series will have waded through fairly specific discussion of numerous studies of metaphor and analogy. Now an attempt will be made to extract and discuss principles of general importance from these studies with respect to each of the processing components used throughout this review to organize description of individual studies. These include:

1. *recognition* of an analogous source,
2. *elaboration* of the ground relating source and target domains,
3. *evaluation* of the elaborated analogy,
4. and *consolidation* of information generated in the process of using an analogy.

Each of these processes, in turn, presents a number of problems from a computational or psychological perspective to which various solutions have been proposed.

7.1. Recognition of a candidate analogy

Given a representational description of a new problem situation (a *target*) of which relatively little is known, how is a reasoner to "connect" this new situation with one or more related situations (*sources*) contained in a "store" of previously experienced situations? From a computational perspective, some form of search is implicated. Hopefully, some sort of organization is imposed on the store of previous experiences which would help constrain search for a candidate source. From a cognitive perspective, similar issues arise with respect to recognition of familiar aspects of the target and retrieval of appropriate experiences from memory. Allowing for partial similarity between target and candidate sources, much of the problem of recognition can be seen as imposing constraints on the retrieval process while still allowing recognition of "figuratively" related source candidates. For example, a strict matching criterion might not allow recognition of relatively abstract inter–domain analogies. Hence, discussion of recognition will proceed in terms of successively more elaborate forms of search constraint.

The most effective but least interesting solution to constraining search proposed in the reviewed studies is to simply give the reasoner a source analog. Many studies do this as a simplifying assumption (e.g., Evans, 1968; Kling, 1971a; Brown, 1977 or Burstein, 1983) either to avoid the complexities of memory organization and retrieval or because their purpose is to examine the use of didactically presented analogies in concept acquisition. Burstein, for example, gives the simulated BASIC student, CARL, both the analogy (e.g., a variable is like a box) and examples of its use. As noted by Kedar–Cabelli (1984), this also simplifies elaboration of the given analogy.

Somewhat more interesting but only moderately more ambitious, some authors conduct what amounts to an exhaustive search of a memory with relatively primitive organization. Becker (1969), for example, apparently considers the entire contents of a memory which indiscriminately records each assertion encountered as input, composed on the basis of collected justifications, or posited as a production. Munyer (1981) proposes exhaustive search as well, although simulating an associative memory to speed search. In both cases, the result of exhaustive search might be a *candidate set* of potential analogs, one (or perhaps several) of which have to be chosen for elaboration of the analogy to proceed. In many cases, the selection of a "best" candidate analog occurs during elaboration and evaluation. For example, Munyer suggests an agenda–like control structure prioritized by the "degree of certainty" for competing analogical views.

Winston's (1980, 1982) proposal for a form of object–oriented search is somewhat more ambitious with regard to constraints imposed on recognition. As initially described (1980), candidate analogs are recognized through a process of "classification–exploiting hypothesizing" in which candidates receive votes during a process which polls ancestors of target objects in a concept hierarchy. Winston's later proposal (1982) involves indexing of acquired rules by the types of objects embedded in their right–hand sides. Hence, on the basis of object types in a target situation, Winston retrieves candidate rules which make predictions about those types of objects. McDermott (1979) proposes a similar mechanism by implicitly indexing known methods by the types of objects and actions involved. While these schemes for recognition of candidate analogs may prove effective for relatively simple intra–domain analogies (e.g., reasoning about water pressure situations), it is not clear they would support more abstract forms of analogy required when reasoning about situations from different domains (e.g., thinking about psychological dynamics in terms of physical hydraulics). There are subtle ways around this problem, of course. Greiner (1985), for example, partially instantiates a target query using a given correspondence between target and source (e.g., Current ↦ FlowRate), and then performs a a complicated but bounded search for an appropriate abstraction. What is subtle about this approach is that the point of correspondence provided by the hint enables recognition (eventually) of an abstractly–related source. In general, a purely object–oriented, bottom–up approach to recognition of candidate source analogs could miss analogies based on relatively abstract correspondence between target and source domains.

As a further level of sophistication with respect to constraints imposed on recognition processes, a variety of relational indexing schemes are proposed. These are somewhat similar to Winston's rule indexing proposal discussed above, but prefer more abstract forms of knowledge for construction of indices and memory organization. Carbonell (1981) best exemplifies this sort of recognition scheme in his proposal for an "invariance hierarchy" over types of represented information. The invariance hierarchy could be seen as an amplification of Gentner's primarily syntactic proposal for "structure mapping" in which higher–order relational information (e.g., cause relations) is preferred over lower–order attributes (e.g., unary predicates asserting object color or temperature). Carbonell's enumeration of information types in decreasing order of invariant transfer (e.g., goals, plans and causal structure are typically preserved in an analogical transfer) constitutes a more detailed and primarily semantic account of what Gentner considers to be higher–order relations. Specification of such an invariant ordering may be important for recognition of analogies in that, according to Carbonell, memory is organized around (i.e., indices are based on) precisely the sorts of knowledge structures likely to be transferred without variation when reasoning analogically. Thus, recognition proceeds by extracting goals, plans or causal structure from the target situation as a retrieval cue and then searching a memory organized according to these types of cues.

This sort of relational indexing is evident in a number of studies reviewed in this paper. Carbonell (1983a) proposes retrieval of solution traces according to a similarity metric which is composed of state descriptions (initial and goal states) and constraints. Although not described in detail, memory in the ARIES system seems to be organized in a manner which facilitates retrieval with this sort of similarity metric (actually a retrieval cue in this context). Dyer (1983a, 1983b) explicitly describes this form of recognition in both the BORIS and MORRIS systems. Memory is organized around instances of planning failures (TAU's) and violation of planning metrics while reading a narrative text triggers a retrieval process in which adages and abstractly related (through similar planning disruptions) narratives are recalled. Both of these recognition schemes, by virtue of indexing and retrieval on the basis of relatively abstract information in the target situation, would allow for recognition and retrieval of genuinely novel metaphors or analogies.

In summary, much of the work on recognition aspects of analogical reasoning can be described in terms of the nature of constraints applied when searching for candidate analogs. Alternative solutions include:

1. giving the reasoner an analog directly,
2. exhaustively searching for a source analog,

3. organizing the search around object–level entities in the target description,
4. and organizing the search around more abstract information extracted from the target situation (e.g., goals, plans or causality).

Of these proposals, the latter might be preferred by virtue of the fact that it both provides for a particularly useful memory organization scheme and allows recognition of analogies between target and source situations which differ markedly in terms of content (e.g., choosing a restaurant and scheduling a meeting as discussed in the section on Dyer's work). Furthermore, as reported in Dyer (1983a) and Schank (1982), these approaches appear to be receiving some empirical support as models of human memory organization and retrieval. These issues are, of course, not settled. As mentioned in the previous paper during concluding remarks about access to problem solving sources, what will be retrieved probably depends upon complicated interactions between several factors: what the reasoner attends to in the target situation, what is available in the store of source experiences, and the degree to which the reasoning context during retrieval matches the encoding context of a stored source. As psychological models of memory organization and retrieval become more sophisticated, computational approaches to recognition may benefit. The converse may also be true. Kolodner (1983ab) is a good example of the confluence of computational and psychological approaches to recognition.

7.2. Elaboration of the analogical ground

Of the four processing components discussed in this paper, elaboration has undoubtedly received the largest share of attention in both computational and psychological approaches to analogical reasoning. As with recognition of a candidate analog, there are a number of general issues or goals which the elaboration of an analogical ground between target and source domains must address. In particular, it is necessary to construct some form of a correspondence or matching between aspects of target and source representations. This correspondence shapes and, in some cases is shaped by, the specific kinds of knowledge that can be transferred between domains. For example, when McDermott (1979) tentatively identifies a method for achieving a current goal (e.g., washing an object by using a method for painting objects), the usefulness and, hence, viability of the correspondence between the two situations (washing and painting) depends in large part on the "mutability" of antecedent conditions for painting in terms of conditions surrounding the new goal of washing. Thus, despite separate consideration of elaboration and evaluation of the analogical ground employed in the review of specific studies in preceding chapters, these two processes are rather closely intertwined. The purpose of elaboration is to establish and perhaps incrementally extend a correspondence between target and source domains which will support an effective transfer of knowledge. The viability and effectiveness of this correspondence as it is extended is constantly subject to evaluation. To the extent possible here, however, aspects of evaluation will be discussed separately in a later section.

In establishing some correspondence between domains, at least from a computational point of view, one must start somewhere. Winston (1980), for example, begins by placing objects (e.g., characters in a story line) in correspondence and then scoring the fit of surrounding relational structures in a bottom–up fashion termed "constraint–directed matching." Dyer (1983a), in contrast, uses relatively abstract relational information concerning planning failures as a starting point in a top–down elaboration of correspondence between narrative episodes. In either case, the elaboration process starts with a partial correspondence which appears to arise out of information generated during the recognition process. Just as elaboration and evaluation cannot properly be viewed as independent processes, so too are recognition and elaboration interdependent. The particularly attentive reader may have been wondering how in the previous discussion of recognition processes (specifically the increasingly elaborate use of semantic constraint to guide identification of analogs) little mention was made of choosing some best candidate analog out of what must often be a set of viable candidates. In fact, it seems likely (at least from a computational point of view) that some elaboration of correspondence for target and source pairings (where there are multiple source candidates) must be conducted in order to choose a particular candidate. Likewise the elaboration process relies on some minimal partial correspondence supplied by the recognition process. This interdependency does not seem particularly unreasonable when one considers that in order for a retrieval cue to be effective in recognition, some aspects of the target situation must correspond to a source candidate, yielding an initial kernel of correspondence.

7.2.1. Constraints on elaboration of an analogy

As with the recognition of candidate analogical sources, constraints of various forms play a crucial role in elaboration of an analogical ground. To the extent that these constraints arise independently either from the source or target domain, it is reasonable to discuss them as aspects of elaboration. Constraints that arise through an interaction between domains will be discussed in the section dealing with evaluation processes. "Independent" constraints, as portrayed in computational approaches to analogical reasoning, appear to come from two sources. First, and of least complication but considerable importance, are constraints that arise within the representation of domains being placed in correspondence. These will be termed **internal constraints**. Second, and of considerable variety, are constraints that originate in reasoning processes or knowledge sources which are not exclusively associated with the domains under consideration but might more reasonably be called background knowledge. These will be termed **external constraints**.

Internal constraints arise primarily from a consideration of which aspects of the source domain are particularly "salient." Discussed earlier within the context of feature–based psychological models of metaphor comprehension (e.g., Malgady and Johnson, 1980), the notion here is simply that of promoting those aspects of the domain which are most important or criterial in describing that domain. Hence, when considering the metaphor, "Dew is a veil," the fact that a veil covers objects might be selected as a candidate for establishing correspondence between domains over the fact that a veil may be made of cotton. Salience in this instance is relatively independent of context or interactions between target and source domains. This strategy is employed by computational enthusiasts as well. For example, Winston (1978) proposes construction of transfer frames by "find(ing) slots known to be important, find(ing) slots that no sibling of the source has, (or) find(ing) slots with values that no sibling of the source has" (p. 152). Examples of such slots or values, respectively, are PURPOSE, HUMPS for camels (Winston doesn't give an example here), or wood as a value for MATERIAL in a description of a ball. Thus, salience as a within–domain constraint helps to determine which aspects of the source domain should be selected in establishing a correspondence between domains.

Constraints on elaboration which arise from reasoning processes or knowledge sources other than domain descriptions are varied. One of the simplest uses of external constraint is Winston's (1978) preferred selection of existing transfer frames. Of course in the case of novel analogies or metaphors, one may not be able to take advantage of such explicit previous experience. Perhaps an equally simple strategy is that of requiring that some paired aspects of the elaborated correspondence be of the same syntactic form or semantic type. For example, Evans (1968) immediately discards potential transformation rules from C to choice components which differ from A to B transformations in the number of subfigures added, removed or matched. Winston's (1980) summative similarity metric used for scoring matches provides an example of promoting correspondence matchings which preserve type information between target and source.

More complex but probably essential are external constraints which arise from the context in which analogical reasoning occurs. This is most easily observed in metaphor comprehension as when, for example, Shakespeare has Romeo compare Juliet to the sun (Gentner, 1982). Rather than inferring that Juliet is flatulent or excessively hot, most readers construct a somewhat more congruous interpretation with respect to the ongoing context: Romeo is describing a woman he rather keenly desires. Such a strategy can also be observed in analogical reasoning during problem solving where the goal–directed nature of reasoning provides a context in which particular aspects of domain knowledge are preferred. As discussed in the first paper of this series, Holyoak's (1985) notion of pragmatic analogical reasoning relies on problem solving goals and constraints a context to constrain elaboration of an analogical ground. The strongest advocate of goal–directed analogical reasoning in the studies reviewed here is Kedar–Cabelli (1984, 1985), who proposes using information reflecting the purpose of reasoning in the domain under consideration (i.e., acquisition of functionally–relevant object concepts) to focus the elaboration of an analogy. Thus, when elaborating the analogical ground relating a retrieved object instance to a target instance, the explanation for how the source satisfies the purpose of learning guides elaboration of how the instances are alike.

A final form of external constraint, and one which makes considerable demands on background knowledge, is the use of a "domain theory" as advocated by Kedar–Cabelli. As described earlier, a domain theory contains background knowledge used for inferring function from structure. For example, a ceramic cup can be grasped because it has an insulated handle. Using this knowledge, particular aspects of the target are given preference in the search for elaboration of a correspondence with the source since these aspects are likely to be important in satisfying the explicit purpose. Thus knowledge of how structural characteristics of objects give rise to functionality guides the process of viewing one object as another so that both achieve the same function (e.g., drinking hot liquids). In this way, varied forms of background knowledge are used to constrain which aspects of a target and source can participate in the analogical ground, a form of constraint which Kedar–Cabelli claims is essential if analogical reasoning is to be tractable. Greiner's

(1985) use of abstractions supported by a starting theory for source and target domains is essentially similar.

In summary, knowledge both internal and external to the target and source situations being compared can be used in constraining the elaboration of an analogical ground between these situations. In either case, these constraints promote particular aspects of represented knowledge over other aspects. Particularly in the case of using background knowledge to constrain elaboration, the strong reliance on existing knowledge structures in analogical reasoning becomes evident. As seen in the first paper of this series, analogical reasoning can serve as an epistemological bridge spanning old and new knowledge (Petrie, 1979).

7.2.2. Characteristics of the elaboration process

Reviewing approaches to analogy discussed earlier in this paper, it is clear that the elaboration of a correspondence between target and source domains is a process of varying complexity. In some studies, a homogeneous mapping between target and source representational descriptions is achieved, and this mapping directly provides a solution to the problem which prompted analogical reasoning. For example, Evans' (1968) ANALOGY system selects a strongest transformation rule reflecting a 1–1 mapping between subfigures of solution components subject to a fixed set of transformations. Similar characterizations could be made of Becker (1969), some of Winston's work (1978, 1980), psychological approaches to proportional analogy problems, and comparison–based theories of metaphor comprehension. In contrast, other studies describe the elaboration process as one of considerable complexity. Carbonell (1983a, 1983b), for example, establishes a partial correspondence over initial and goal states and then describes a complicated search space of plan transformations or attempts to satisfy plan justifications in an effort to find a solution path in the target problem domain. The notion of "repair" introduced by Burstein (1983) or the attempt to "justify" a new case with respect to purpose through use of a domain theory as described by Kedar–Cabelli (1985) suggest similar complexity in the construction of an effective correspondence between analogous situations.

Simple, relatively homogeneous correspondence as an end in itself stems from a relatively simple view of analogy: analogy consists of a transformational mapping which, when applied, renders two superficially dissimilar situations virtually identical. In this conception, the real work of analogical reasoning is composed of processes which incrementally extend aspects of the source situation into the target situation such that the coherence of the mapping between situations is maintained. More complex forms of elaboration, by contrast, take analogical reasoning to be much more open–ended. An elaborated correspondence serves as a basis for advancing hypotheses rooted in a full understanding of the source domain into the less well understood target domain. These hypotheses must be verified within the target domain, giving rise to a process of experimentation. Experimentation, in this sense, refers to an interplay between processes of elaboration and evaluation.

7.3. Evaluation of the analogy

As in elaboration, imposition of *constraints* provides the central story line for activities during evaluation of a developing analogy. Assuming that a process of elaboration is incrementally proposing additions to a growing correspondence between source and target domains, the goal of evaluation is to examine the plausibility of these extensions within the target domain. In addition, some form of feedback is desirable at a more global level, suggesting whether effort being expended during analogical reasoning might better be devoted to some alternative form of problem solving behavior. For example, Carbonell (1986) mentions a perseverance threshold for falling back on knowledge–weak search. As suggested earlier, a distinction can be made between sources of constraint which arise from domain or background knowledge when target and source are considered separately and sources of constraint which arise through domain "interaction." For evaluation processes, the latter source of constraints will be of central importance.

As with other process components, there is some diversity of proposals for identifying constraints based on domain interaction. From this variety, two general issues can be abstracted:

1. *confirmation* of knowledge extended from source to target domains and
2. *repair* of inappropriate extensions.

Obviously, these process issues are not easily separable as exclusive concerns of evaluation. Instead, processes of elaboration and evaluation are strongly interdependent in the sense that both operate on a correspondence mapping which serves as a bridge between source and target domains. Confirmation of information transferred over this bridge and repair of its structure in the face of incomplete or inappropriate transfer should properly be seen as the result of elaborative and evaluative processes working together. What results should be relevant within the reasoning context.

7.3.1. Confirming an extended analogy

For confirming knowledge extended from the source to the target domain, two distinct approaches are evident in computational studies. Extended knowledge can be *tested* with respect to validity or usefulness in the target domain, or the reasoner can attempt to establish a *justification* for inference or action in the target domain which mirrors a justification for similar activities in the source domain. In essence, these approaches are identical in their attempt to validly instantiate knowledge extended across domains. However, reusing a justification is distinguished by providing additional constraints on *what* to consider as supporting information in the source. Both these approaches are clearly related to psychological conceptions of preserving relational structure when constructing a mapping between domains (e.g., Gentner, 1983a).

Confirmation by *testing predictions* amounts to evaluating the plausibility of information extended in the elaborated correspondence mapping by considering prototypical knowledge of the target domain or by examining the utility of transferred knowledge in some ongoing reasoning process. As an example of confirmation using target expectations, Winston (1978) "filters" or selects transferred knowledge on the basis of common slot types between the source and a "typical" representation of the target concept. In Winston's later work (1980), an abductive reasoning process is invoked in which causal antecedents of desired conclusions in the target domain must either be verified by existing knowledge of the target or requested from an external teacher. In both cases, knowledge of typical aspects of the target is used to verify predictions from the source domain.

More ambitious evaluative measures weigh the problem solving efficacy of transferred information. For example, Carbonell (1983a) proposes a form of means–ends–analysis to transform analogous solution paths. T–ops used to alter solution paths are indexed by path differences which they reduce. Hence, a similarity metric used to detect differences and provide the difference table for T–ops (essentially an indexing scheme) must be very knowledgeable about desirable or undesirable solution forms. Perhaps less special purpose, Greiner's (1985) NLAG must be able to deductively determine that a source conjecture, instantiated within the target domain, is *useful* for solving the target problem. In contrast, Burstein (1983) describes critical interactions between CARL and a teacher in which the latter gives explicit feedback on problem solutions (e.g., whether the response to a question is correct). In all three cases, the success of transferred knowledge for reaching a target solution contributes to evaluation of the developing analogy.

Establishing justifications for activities in the target domain which mirror valid justifications for similar activities in the source domain also plays an important role in evaluation. Justification in this context refers to some representational description of the "reasons" which support a particular inference or action in a domain. Becker (1969) is an early advocate of this approach with his concept of "motivated" analogical correspondence. In his work, motivating facts are collected which support placing unlike objects in correspondence. Winston's (1978) justification frames are an explicit mechanism for capturing those aspects of a target description which must be present if a known analogy (i..e., transfer frame) is to be useful. As described by Winston, a justification frame for an analogy between a table and a cube which involves a common purpose for these objects (e.g., to eat or write) might consist of recording that both target and source descriptions must characterize the objects as being of medium size, having a very flat top and being very level. This concept of functional justification is extended by Winston, et al. (1983) and used to good purpose by Kedar–Cabelli (1985).

Using a justification to capture aspects of a situation which satisfy some purpose is given an important role in several other computational studies of analogical reasoning. Brown (1977) describes "plan justifications" as collections of assertions which relate steps in a solution plan to represented facts about the task domain found in a goal description. Having found a justified source solution, justifying assertions must be confirmed as the candidate solution is "inverse–mapped" into the target domain. If justifying assertions cannot be confirmed, further work with the existing analogy or introduction of a new analogy may be required. Replaying justifications also plays a central role in Carbonell's derivational approach to analogical reasoning (1983b, 1986). Here, justifications are stored as part of a derivational trace of decisions made in solving a source problem (e.g., programming quicksort in PASCAL). Given an analogous target problem (e.g., programming quicksort in LISP), justifications or reasons for making particular choices among actions in the source derivation must be confirmed in the construction of a solution in the target problem or replaced by alternative justifications if an analogy is to succeed. Kedar–Cabelli (1985) uses an explanatory justification within the source domain (e.g, the structural reasons why a ceramic cup can be used to drink hot liquids) both to constrain and to confirm analogous reasoning in the target. Each of these projects exemplifies the notion of evaluating an extended analogical correspondence by appealing to background knowledge supporting such extensions within source and target domains.

7.3.2. Repairing a faulty analogical extension

In addition to confirming or justifying analogically extended information between domains, many computational approaches explicitly address issues of how to recover when an extension of the correspondence between domains fails. The specific nature of recovery depends, of course, on the seriousness of failure. Difficulties encountered in attempts to instantiate transferred information generally require less ambitious forms of repair, while inappropriately transferred information may result in considerable further reasoning or abandoning the foundering analogy altogether.

In McDermott's ANA (1979), repair of mutable precondition failures is used to properly instantiate transferred planning information between source and target domains. ANA encounters difficulties in using an analogy by generating inappropriate subgoals, transferring insufficient constraints, or transferring unnecessary constraints. Difficulties are recognized as actions are attempted in the paint shop environment. Repair in each case depends on a combination of background knowledge of what sorts of entities can participate in particular activities and direct feedback from the "task master." For example, loading a spraying machine with paint when the goal is to wash an object can be repaired by substituting water for paint when loading the spraying machine. Underspecification of transferred methods can lead to planning failures like stacking too many objects in a constrained area. These failures are repaired by generating additional subgoals which transform problematic but "mutable" aspects of the plan either on the basis of known methods (e.g., removing an object from a crowded area) or on the basis of advice from the task master. Finally, difficulties with over-specified plans (e.g., moving an obstructing object to a distant location when a neighboring location would suffice) can be repaired by asking the task master which constraints are unnecessary. In summary, McDermott describes a number of strategies for repairing improperly instantiated plan components. These strategies are guided by domain knowledge or external advice. According to McDermott, if substitutions are insufficient for repairing a faulty plan, the problematic analogy is abandoned and alternate analogous methods are sought. In many respects, McDermott's description is quite similar to subsequent work by Carbonell on transformational analogies in problem solving (1983a).

A number of authors propose using multiple analogies when knowledge extended from source to target in an initial analogy proves inappropriate. Burstein (1983) describes the integration of multiple analogies in an effort to repair incorrect predictions by the learner regarding the effects of simple assignment statements. While these pedagogical strategies occur frequently in human learning, the approach leads to complicated processing issues. Analogies at differing levels of abstraction must be integrated into a usable concept, avoiding what Halasz and Moran (1982) characterize as a "baroque collection of special–purpose models." (p. 34) As another example, Anderson *et al.* (1984) describe the presentation of simplifying examples by a teacher which serve as analogous templates for problematic aspects of a LISP function being written by the learner. In both cases, errors introduced by an inappropriately elaborated ground or correspondence between source and target situations are repaired by the introduction of additional analogies which must be integrated with the original analogy. This approach appears psychologically plausible. Clement (1981, 1982), discussed in depth in the first paper of this series, describes the use of intermediate "bridging analogies" by expert problem solvers in physics to aid in establishing a ground between source and target problem descriptions.

In summary, the process by which a correspondence mapping between analogically related domains is incrementally extended appears inherently nondeterministic (Hayes-Roth, 1978). Extensions to an existing correspondence can be viewed, at best, as tentative hypotheses reflecting a partial matching between source and target domains. Interactions between these domains serve as an invaluable source of constraint on elaboration and are uncovered in a process of evaluation. Evaluation occurs at many levels, including:

1. evaluating *predictions* with respect to expectations supported by knowledge of the target domain,
2. examining problem solving *utility* on the basis of transferred information,
3. establishing *justifications* in the target domain which mirror supporting reasoning for analogous inferences in the source domain,
4. and attempting *repair* of inappropriately extended information.

As a result of the evaluation process, aspects of the correspondence mapping may be changed or deleted, additional analogs may be integrated to suggest differing hypotheses concerning the target domain, or the original analogy may be abandoned altogether in favor of an alternate line of reasoning.

7.4. Consolidation of analogical reasoning

Processes of recognition, elaboration and evaluation result in some form of correspondence between target and source domains. How to consolidate this correspondence in a fashion which will benefit future reasoning performance in these or similar domains must be addressed if the reasoner is to "learn by analogy." Although most authors concentrate on consolidation in the wake of successful analogical reasoning, there may be opportunities for consolidation even in cases where analogical reasoning fails.

Probably the simplest form of consolidation involves direct recording of information effectively transferred from source to target domain. Of the computational studies reviewed in this survey which address consolidation issues, most perform this simple form of learning. Brown (1977), for example, directly "lifts" mappable plans and code which contribute to a solution in the target domain. In a similar fashion, McDermott (1979) describes consolidation as a process of recording productions which suggest actions should similar conditions arise in future planning episodes. In both cases, consolidation is distinguished by the fact that what is learned is strongly context specific with little or no generalization occuring.

A more ambitious form of consolidation involves storing the correspondence mapping which results from successful analogical reasoning so that the mapping might be available for use later under similar circumstances. In essence, this amounts to storing the process of analogical reasoning itself in the hopes that elaboration and evaluation might be circumvented in the future. Winston (1978), for example, suggests the formation and storage of transfer and justification frames. In subsequent reasoning tasks, recognition of potential analogs proceeds by first attempting to reuse an existing transfer frame, providing that a stored justification can be satisfied in the target situation.

Hoping to acquire knowledge with somewhat wider applicability, many projects propose a process of consolidation which involves some form of generalization over target and source domains. At the most atomic level, Becker (1969) and Winston (1980, 1982, 1983) describe the acquisition of generalized rules reflecting causal inferences common to selected aspects of both target and source domains. Becker's system attempts such generalizations under its own initiative (e.g., that if a man is a fireman, then he wears red suspenders). Winston explicitly guides the generalization process through suggestions of a teacher. Of more complexity, a number of authors argue for consolidation by forming generalized plans or problem solving derivations common to both target and source domains. Gick and Holyoak (1983) argue strongly for this sort of learning from a psychological point of view, while Carbonell (1983a, 1983b) is a strong proponent from a computational perspective. In both cases, the manner in which generalized plans are stored strongly determines their utility during subsequent reasoning. Finally, Burstein (1981, 1983) argues for concept formation through analogies applied at multiple levels of abstraction.

While consolidation processes discussed above all rely on information resulting from successful analogical reasoning, Carbonell (1983a) argues that learning opportunities also exist when analogy fails: processes of recognition and evaluation can be altered on the basis of information obtained during failures. When recognition processes fail to suggest any candidate analogs, alternate problem solving methods may reach a solution. A post–mortem analysis may suggest that the found solution is in fact similar to a previously known problem solution. In this case, Carbonell recommends "tuning" the similarity metric used in the recognition process so that known solutions will be recognized in the future. Likewise, if a recognized analogy fails to produce a solution, the similarity metric could be specialized to suppress false recognition in the future. Similar approaches hold for tuning the enablement conditions of transformation operators which are not recognized as applicable or fail once they are recognized. While none of these suggestions appear to have been implemented, they illustrate a role which failures may play in changing the process of analogical reasoning.

In summary, a variety of mechanisms for "learning by analogy" are proposed. These range from simply recording the direct results of analogical transfer or the structure of the analogy itself to an ambitious interest in altering the process of analogical reasoning. Most proposals involve forming generalized rules or plans which capture common aspects of target and source domains. Given the dramatic role envisioned by some for "learning by analogy," we should ask if learning while reasoning by analogy is in some fashion unique or clearly different from other forms of learning. In review, proposals for "learning by analogy" in AI are not very different from proposals for learning without analogies. The same techniques and difficulties would seem to apply.

As a context within which learning occurs, however, analogical reasoning may provide some interesting issues. When new situations are understood *as if* they were instances of known situation classes, a hypothetical element is added to problem solving which both complicates and constrains the learning task. Complications arise in that errorful conceptualizations must be expected: recognition and elaboration of a source anticipates the formation of instance classes in a manner which may be sufficient for the current reasoning context but prove faulty in general. Hence learning mechanisms must support the incremental evolution of concepts. Constraints arise in that the learning task is strongly connected to existing knowledge sources, potentially bypassing the need for extensive experience in learning superficially variant problem solving strategies.

8. Epilogue and Prospectus

Before considering the future of analogy in computational approaches to reasoning and learning, we should review what progress has been achieved towards these ends in the past twenty years. According to Minsky, reviewing Evan's thesis shortly after completion,

> ... it is becoming clear that analogical reasoning itself can be an important tool for expanding artificial intelligence. I believe it will eventually be possible for programs, by resorting to analogical reasoning, to apply the experience they have gained from solving one kind of problem to the solution of quite different problems. (Minsky, 1966, p. 251)

Perhaps eventually, but not yet. After reviewing computational studies of reasoning and learning by analogy, it is easy to conclude that relatively little has been accomplished. Judging from the scarcity of implemented computational analogizers in the AI literature, reasoning about a new situation by effectively using analogously related previous experience is a very hard problem. The computational components of recognition, elaboration, evaluation and consolidation developed and discussed in preceding sections not only reflect the academic topography of work done in AI and related disciplines, but these components also are indicative of what characteristics of analogical reasoning and learning by analogy are particularly difficult, open research problems.

As mentioned earlier, most work on analogical reasoning from a computational perspective has addressed issues of elaboration and evaluation, slighting complex issues of recognizing candidate analogs and consolidating information generated during their use. For each process component, the most ambitious and (arguably) most promising computational approaches have not been fully implemented. Examples include dynamically organized memories which support recognizing useful analogies, a focus on contextual relevance for elaboration, reusing justifications for evaluation, and construction of generalized problem schemata for consolidation. While suggestions for computational mechanisms have proliferated in recent years, few of these proposals have proven useful in actual implementation. Instead, implemented computational models usually demonstrate carefully crafted solutions to isolated problems. If only from a performance point of view, further computational studies of analogical reasoning would do well to concentrate on the following:

1. *Recognition* of useful source analogs should be possible when a large store of candidates are available in memory. At some level, recognition almost certainly depends upon matching encoding and retrieval contexts. As an additional constraint, organization of the memory store should be suitable for reasoning tasks other than retrieving analogies.

2. *Elaboration* of a ground or mapping between source and target should be possible when the source content is both extensive and varied. Selection of what to transfer, as argued throughout these papers, depends heavily on a context of use for analogical reasoning.
3. *Evaluation* of transferred material, in general, requires inferences within the target domain which may rival the difficulty of the original problem. Hence analogical reasoning can be understood as a program for experimentation, supported by continued problem solving and even further analogies. It would be interesting to compare problem solving by analogy with more conventional techniques in some domain (e.g., means–ends–analysis in Carbonell's reconstructive framework).
4. *Consolidating* results of analogical reasoning, as described above, differs little from traditional machine learning techniques. However, storage and subsequent use (either directly or by analogy) of resulting concepts may prove otherwise. Direct comparisons of "learning by analogy" and learning without analogy should contribute to our understanding of more general learning principles.

One is left to wonder whether verified computational mechanisms will be forthcoming in the near future, or whether effective use of analogy in reasoning and learning is well beyond our current computational reach. I will argue that full computational realizations of analogy are still rather distant. Effective use of analogies depends on particularly troublesome issues having a rich, but often overlooked history in other academic disciplines. Recognition of analogies raises longstanding research questions of memory organization and use, whether by machines or natural reasoners. Elaboration and evaluation of a candidate analogy likewise raises difficult issues of focusing attention amidst the myriad details of stored experience. Consolidation in the wake of analogical reasoning depends on a fuller understanding of inducing useful predictive information when faced with an unpredictable experiential record. Hence, rather than being a *special form of reasoning or learning*, use of analogy and metaphor should be more accurately viewed as reasoning abilities which are thoroughly integrated into more mundane forms of intelligent behavior. Just as there is nothing particularly unusual about analogical reasoning in the intellectual repertoire, there is nothing particularly easy about analogical reasoning from a computational point of view.

The foregoing conclusion may appear a dark vision to the enthusiast. In contrast, I will argue that computational studies of analogical reasoning provide a valuable testbed for "real–world" reasoning and learning tasks. Hopefully this provides a realistic motivation for continued investigation of analogy and metaphor in reasoning and learning. Reasoning about the environment, whether based in mechanical or natural artifacts, presents seemingly endless inferential possibilities. Useful inferences must be drawn to obtain solutions to such tasks: existing knowledge in the task domain must be brought to bear in a fashion which effectively focuses the reasoning process. Analogy and metaphor serve just such a purpose. Without this ability, much of the experienced environment would be unintelligible.

References

Anderson, J.R., Greeno, J.G., Kline, P.J. and Neves D.M. (1981) Acquisition of problem–solving skill. In J.R. Anderson (Ed.) *Cognitive skills and their acquisition.* Lawrence Erlbaum Associates: Hillsdale, New Jersey, 191–230.

Anderson, J.R. (1983) *The architecture of cognition.* Harvard University Press: Cambridge, Massachusetts,

Anderson, J.R., Farrell, R. and Sauers, R. (1984) Learning to program in LISP. *Cognitive Science 8*, 87–129.

August, S.E. and Dyer M.G. (1985) Understanding analogies in editorials. *Seventh Annual Conference of the Cognitive Science Society*, 845–847.

Becker, J.D. (1969) The modeling of simple analogic and inductive processes in a semantic memory system. *IJCAI*, 655–668.

Brown, R. (1977) Use of analogy to achieve new expertise. AI–TR–403, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Burstein, M.H. (1981) Concept formation through the interaction of multiple models. In *Proceedings of the Third Annual Conference of the Cognitive Science Society*, 271–273.

Burstein, M.H. (1983) Concept formation by incremental analogical reasoning and debugging. In *Proceedings of the Machine Learning Workshop*, 19–25.

Carbonell, J.G. (1981) Invariance hierarchies in metaphor interpretation. *Proceedings of the third annual meeting of the Cognitive Science Society*, 292–295.

Carbonell, J.G. (1983a) Learning by analogy: formulating and generalizing plans from past experience. In R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Eds.) *Machine learning: an artificial intelligence approach*, 137–162.

Carbonell, J.G. (1983b) Derivational analogy in problem solving and knowledge acquisition. In *Proceedings of the Machine Learning Workshop*, 12–18.

Carbonell, J.G. (1986) Derivational analogy: a theory of reconstructive problem solving and expertise acquisition. In R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Eds.) *Machine Learning: An Artificial Intelligence Approach*, volume 2, 371–392.

Clement, J. (1981) Analogy generation in scientific problem solving. In *Proceedings of the third meeting of the Cognitive Science Society*, 137–140.

Clement, J. (1982) Analogical reasoning patterns in expert problem solving. In *Proceedings of the fourth meeting of the Cognitive Science Society*, 79–81.

Clement, J. (1983) Observed methods for generating analogies in scientific problem solving. Presented at the annual meeting of the American Educational Research Association, Montreal, Canada.

Dyer, G.M. (1983a) *In–depth understanding.* MIT Press: Cambridge, Mass.

Dyer, G.M. (1983b) Understanding stories through morals and remindings. *IJCAI*, 75–77.

Evans, T.G. (1968) A program for the solution of a class of geometric analogy intelligence test questions. In M. Minsky (Ed.) *Semantic information processing.* MIT Press: Cambridge, Mass, 271–353.

Gentner, D. (1982) Are scientific analogies metaphors? In D. Miall (Ed.) *Metaphor: problems and perspectives.* Humanities Press: New Jersey, 106–132.

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

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

Green, C. (1969) Theorem proving by resolution as a basis for question answering systems. In D. Michie and B. Meltzer (Eds.) *Machine Intelligence 4.*

Greiner, R. (1985a) Learning by understanding analogies. *Proceedings of the Third International Machine Learning Workshop*, June 24–26, 50–52.

Greiner, R. (1985b) *Learning by understanding analogies.* Ph.D. thesis, Stanford University. STAN–CS–85–1071.

Halasz, F. and Moran, T.P. (1982) Analogy considered harmful. In *Proceedings of the Conference on Human Factors in Computer Systems*, March 15–18, National Bureau of Standards, Gaithersburg, MD.

Hayes–Roth, F. (1978) The role of partial and best matches in knowledge systems. In D.A. Waterman and F. Hayes–Roth (Eds.) *Pattern–directed inference systems.* Academic Press: New York, 557–576.

Hobbs, J.R. (1983a) Metaphor interpretation as selective inferencing: cognitive processes in understanding metaphor (part 1). *Empirical Studies of the Arts 1*(1), 17–33.

Hobbs, J.R. (1983b) Metaphor interpretation as selective inferencing: cognitive processes in understanding metaphor (part 2). *Empirical Studies of the Arts 1*(2), 125–142.

Holyoak, K.J. (1985) The pragmatics of analogical transfer. *The Psychology of Learning and Motivation, Vol. 19*, 59–87.

Kedar–Cabelli, S. (1984) Analogy with purpose in legal reasoning from precedents: a dissertation proposal. LRP–TR–17, Laboratory for Computer Science Research, Rutgers University.

Kedar–Cabelli, S. (1985) Purpose–directed analogy. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, 150–159.

Kilpatrick, W. (1982) An a–frame model for metaphor. In *Proceedings of the International Conference on Cybernetics and Society*, 83–87.

Kling, R.E. (1971a) Reasoning by analogy with applications to heuristic problem solving: a case study. Ph.D. Thesis, CS–216, Stanford University.

Kling, R.E. (1971b) A paradigm for reasoning by analogy. *Artificial Intelligence 2*, 147-178.

Kling, R.E. (1971c) Reasoning by analogy as an aid to heuristic theorem proving. Presented at the IFIP Congress, Ljubljana, Yugoslavia.

Kolodner, J.L. (1983a) Maintaining organization in a dynamic long–term memory. *Cognitive Science 7*, 243–280.

Kolodner, J.L. (1983b) Reconstructive memory: a computer model. *Cognitive Science 7*, 281–328.

Laird, J.E., Rosenbloom, P.S. and Newell, A. (1983) Towards chunking as a general learning mechanism. *AAAI84*, 188–192.

Lebowitz, M. (1982) Correcting erroneous generalizations. *Cognition and Brain Theory 5*(4), 367–381.

McDermott, J. (1979) Learning to use analogies. *IJCAI*, 568–576.

Miller, C. (1983) Analogy and mathematical reasoning: a survey. PB84-144401. Leeds Univ., England.

Minsky, M. (1966) Artificial intelligence. *Scientific American 215*, 246–263.

Mitchell, T.M. (1982) Generalization as search. *Artificial Intelligence 18*, 203–226.

Mitchell, T.M., Keller, R.M., and Kedar–Cabelli, S.T. (1986) Explanation–based generalization: a unifying view. *Machine Learning 1*, 47–80.

Munyer, J.C. (1981) Analogy as a means of discovery in problem solving and learning. Ph.D. Thesis, University of California, Santa Cruz.

Newell, A. and Simon, H.A. (1972) *Human problem solving.* Prentice Hall: New Jersey.

Ringle, M. (1979) Philosophy and artificial intelligence. In M. Ringle (Ed.) *Philosophical perspectives in artificial intelligence.* Humanities Press: New Jersey, 1–20.

Schank, R.C. (1982) *Dynamic Memory: A theory of reminding and learning in computers and people.* Cambridge University Press: New York.

Sternberg, R.J. (1977) *Intelligence, information processing and analogical reasoning: the componential analysis of human abilities.* Lawrence Erlbaum Associates, Publishers: Hillsdale, New Jersey.

Winston, P.H. (1978) Learning by creatifying transfer frames. *Artificial Intelligence 10*(2), 147-172.

Winston, P.H. (1980) Learning and reasoning by analogy. *CACM 23*(12), 689-703.

Winston, P.H. (1982) Learning new principles from precedents and exercises. *AI 19*, 321-350.

Winston, P.H. (1983) Learning by augmenting rules and accumulating censors. In *Proceedings of the Machine Learning Workshop*, 2-11.

Winston, P.H., Binford, T.O., Katz, B. and Lowry, M. (1983) Learning physical descriptions from functional definitions. *AAAI83*, 433–439.

Winston, P.H. (1984) *Artificial Intelligence*, 2nd edition. Addison–Wesley Publishing Company: Reading, Massachusetts.