4th AAAI-Sponsored Knowledge Acquisition for Knowledge-Based Systems Workshop
Banff, Canada, October, 1989

Analogy In Knowledge Acquisition: Description of An Analogical Tool

Debbie Leishman - Knowledge Science Institute, University of Calgary
& Institute for Research in Construction
(leishman@cpsc.UCalgary.CA)

Abstract

Learning and reasoning by analogy provides one mechanism for acquiring and effectively using knowledge. This paper describes the development of an Analogical Tool that performs this learning and reasoning and which is based on evaluations of partial correspondences over conceptual graphs. The tool results from instantiation of a general framework for analogy that arises from analysis of previous work and characterizes analogies as "common generalizations". Using conceptual graphs [Sowa, 1984] as the knowledge representation scheme for the general framework, the tool forms analogies and performs analogical inferencing. Analogy formation in the tool derives minimal common generalizations as preferred "stronger" analogies. This formation is under the control of four formation evaluations and two ordering evaluations. The paper then presents and example of learning problem solving schemas and their subsequent use in solution hypothesis.

1 Introduction

Analogy has many possible uses in a knowledge acquisition cycle. In the early stages of elicitation, experts often prefer and find it easier to input their knowledge in the form of cases or descriptions of problem solving situations rather than in explicit rules [Witten & MacDonald, 1987]. Analogy can be used here for taking these cases and forming general problem solving schemas which can later be used for reasoning about a similar case. Descriptions of these cases may come from text as well as from an expert and may require complex descriptions. The Analogical Tool described in this paper performs this kind of schema induction and subsequent solution hypothesis. Analogy can also be used to determine similarities and differences between input of multiple experts to a knowledge acquisition tool. Further on in a knowledge acquisition cycle, analogy can be used for exploration in an existing knowledge base. Analogies found between structures can provide a basis for hypothesizing new knowledge followed by evaluation of the plausibility of this knowledge. Similarly, in a knowledge base of cases, analogy can be used to determine where and how to incorporate new cases. Analogy is also useful in the very simple case of copy and edit when adding new facts to a knowledge base as in the CYC project [Lenat, 1986].

The Analogical Tool described in this paper represents complex analogues taken from natural language and forms minimal common generalizations as preferred analogies. This tool could be used for any of the purposes described above but will be described here as forming problem solving schemas and later using them for solving similar problems.

This paper will first develop a general framework for analogy which instantiates to the Analogical Tool when conceptual graphs are used for the knowledge representation scheme. Following this, the Analogical Tool will be described in terms of forming analogies and perfoming analogical inferencing based on these analogies. Finally the tool will be used on an example from Gick and Holyoak to produce schema abstractions.

1.1 Analysis of Previous Work on Analogy

Analysis of colloquial connotations and psychological processes of analogy has generated research which can be classified as 1) Abstract theories of analogy; 2) Psychological models of analogy; and 3) Evaluation criteria for analogy.

Abstract theories view analogy as an algebraic or logical process. Comparison of works in this area reveals the common theme that analogy is the formation of a plausible partial correspondence between parts of the participating analogues. Indurkhya [1985], for example, describes a theory of analogy and metaphor called Approximate Semantic Transference which forms AT-MAPS. These maps represent an approximately coherent partial mapping of terms between source and target domains. Gaines and Shaw [1982] explicate analogy as a partial correspondence between systems represented as mathematical categories. Russell's use of determinations [Russell, 1987] also requires the formation of a plausible partial correspondence between terms.

Psychological models of analogy attempt to account for how and in what circumstances people reason from analogies. Comparison of works in this area reveals the common theme that a critical part of any model of analogy is the knowledge that is used and how it is represented. For instance, Gentner's Structural Mapping Theory [1983] is based on a principle of systematicity which attempts to maintain chains of relations when forming correspondences between domains. The choice of higher order predicate calculus as the knowledge representation scheme aids in detection of these chains.

Evaluation criteria for analogy focus on the problem of what is a plausible analogy—why one analogy is seen as more plausible than another. Comparison of works in this area reveals the need to control the formation of plausible partial correspondences through evaluation criteria which relate to psychological notions of plausibility and the applicability of analogies. For example, evaluation criteria which use generalization/ specialization hierarchies are central to the CYC project being conducted at MCC [Lenat et al, 1986], as well as to the work of Hofstadter and Mitchell on analogical reasoning [1988]. Winston's [1981] early paper on analogical reasoning contains many useful criteria such as the use of case relations and matching of terms sharing the same relations between them; this latter criterion is also evident in Gentner's work.

1.2 A General Framework for Analogy

The above discussion based on classifying works on analogy into the three categories leads to conclusions which can be seen as comprising critical functions of analogy. Any computer emulation must thus contain the following three functions:

1) Semantic schemas for representing knowledge of the domains involved in the analogies.

2) Algorithms for finding partial correspondences between abstract structures.

3) An Evaluation relation for ordering analogies by plausibility.

These critical functions can also be seen as arising from the description of analogy given by Polya [1954]. He describes analogy as a type of plausibility reasoning showing similarity on a definite conceptual level and defines analogy in terms of forming a "common generalization" of analogues.

From this description by Polya we can describe a general framework for analogy which contains within it the three critical functions. We still need some kind of semantic schemas for representing knowledge of our domains, but we can now view the formation of partial correspondences between analogues as arising naturally from formation of a common generalization of the analogues. Those parts of the analogues which have been generalized to a common concept are also those which are put into correspondence. Also implicit in Polya's description of forming a common generalization is that there is often more than one possible common generalization which can be formed. In an effort to find these "stronger" common generalizations we need some kind of ordering relation. This points to our third critical function, the need for an evaluation relation for ordering analogies by plausibility.

The general framework for analogy in Figure 1 shows two analogues and their common generalizations. Each analogue consists of parts, and each of these parts is "part-of" the analogue. The common generalization also contains parts and these parts are each a "generalization" of some part of each of the analogues. The parts of the two analogues which have a common generalization define the parts which are in correspondence with each other. Finally we need an ordering relation which determines the "stronger" analogies. In most applications of analogy, the "stronger" analogies correspond to minimal common generalizations. Figure 1 shows there can be many common generalizations formed but only some are minimal. These generalizations are minimal within our ordering relation because they contain the most information; they are the most specific of the possible generalizations. There can typically be more than one of these minimal common generalizations.

Analogical inferencing can also be described in the terms used above. To perform inferencing analogically from a minimal common generalization means to look for an even more specific generalization. Usually one analogue (the source analogue) is better known than the other (the target analogue). The source analogue is often used to suggest the existence of non-evident structure in the target analogue. In this way some "part-of" the source analogue is hypothesized as existing in the target analogue.

2 Conceptual Graphs for Analogy

Use of conceptual graphs [Sowa, 1984] as the knowledge representation scheme for the general framework described above leads to development of an Analogical Tool. A single conceptual graph states a proposition and contains concept nodes, relation nodes, and arcs between them. This structure allows analogues to be represented as finite bipartite graphs. Concept nodes represent entities, attributes, states and events while relation nodes specify relationships between concepts and include case relations among others. Each concept in a conceptual graph is part of a lattice of concepts. This lattice supports inheritance, and concepts higher in the lattice are more general than lower ones. Concepts to be used in a domain must be specified beforehand.

Graphs can also express more complex sentences that contain individuals, modality, nested propositions and co-references as shown in Figure 2. The sentence for this graph is shown at the top. Co-reference links are shown with dashed lines and the nested propositions are shown by the boxes surrounding subgraphs. Concepts in a conceptual graph can be generic, such as [Person] or can refer to particular individuals such as [Person: Sam]. This concept refers to a specific individual *Sam* of type *Person* where *Sam* is specified in the referent field of the concept. The modal concepts *Think* and *Believe* in Figure 2 take nested conceptual graphs as their objects. These nested graphs are *Propositions* whose referent field is the set of conceptual graphs being asserted. They are also referred to as context boxes and a box with no type label defaults to a proposition (context) box. Also present in this figure are co-reference links which show anaphoric references and express an equality relation between concepts. Co-reference links often join pronouns to their referents or refer to a concept previously mentioned. In Figure 2 for example the phrase, "Sam thinks that the house has a kitchen," refers to a house that Sam already knows about. This requires a co-referent to house outside the context of the house having a kitchen.

Each conceptual graph is also embedded in a larger semantic net which contains background knowledge that enriches the meaning of the graph.

2.1 Specialization and Generalization

The conceptual graph model defines specialization operations used for selectional constraints on sentence formation and generalization operations for logic and model theory. Specialization of a conceptual graph can be done in two ways. First a particular concept

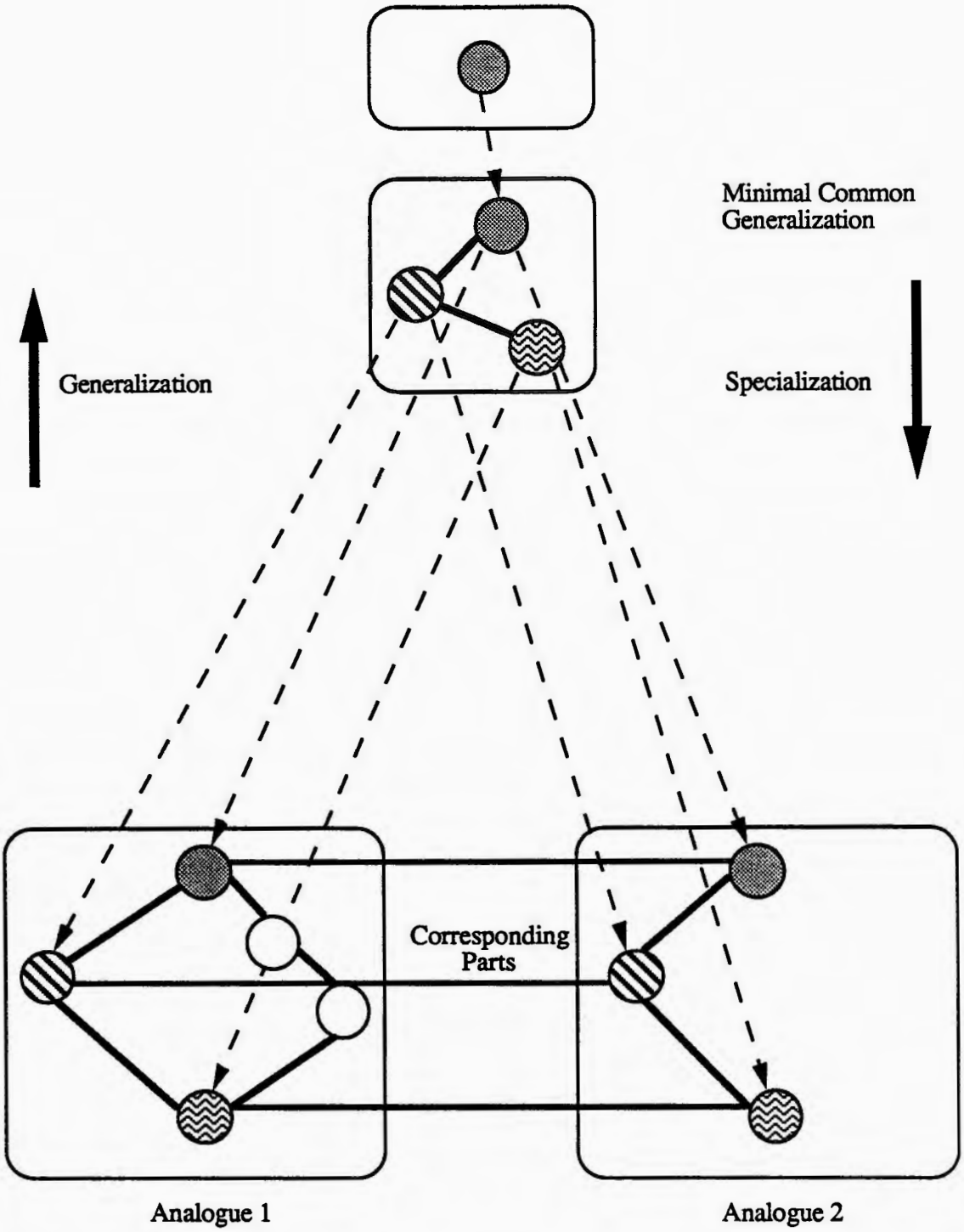

Figure 1 A General Framework for Analogy

can be restricted, either by addition of an individual to a generic concept or by replacing a concept with one of its subtypes from the concept lattice. Second, a conceptual graph can be specialized by making the graph more extensive. This amounts to adding more concepts and relations to a graph. Rules for when and how these specialization operations take place are specified by the conceptual graph model [Sowa, 1984]. These specialization rules and operations are not truth preserving but enforce constraints on sentence formation.

Generalization of conceptual graphs is truth preserving and if a graph u is a specialization of a graph v, written u <= v, then v is a generalization of u. Generalization of conceptual graphs defines a partial ordering called a generalization hierarchy. Graphs can be generalized in three ways:

1) Any subgraph is a generalization of the original graph;

2) Replacing a type label with a supertype generalizes a graph;

3) Deleting an individual from the referent field generalizes a graph.

These specialization and generalization operations have important uses in analogy formation and analogical reasoning.

Conceptual graphs were chosen as the knowledge representation scheme for the Analogical Tool because of their linguistic and logical basis. They are a semantically rich scheme which captures much of the fullness of natural language while maintaining necessary logical connections. The choice of conceptual graphs is based on several other reasons as well:

1) They have a mapping to logic and thus are compatible with any system based on logic.

2) Their straightforward mapping to and from natural language allows for a linguistic description of analogues.

3) They are computational in that they have well defined algorithms for manipulation of the graphs.

4) Their definition and associated operations contain elements which correspond naturally to the general framework for analogy.

The notions of "part-of", "generalization" and the "generalization ordering" relation of the general framework for analogy described above have corresponding elements in the conceptual graph model. Parts in the conceptual graph model consist of concepts, relations and the arcs between them. Generalization in the conceptual graph model happens in two ways, by climbing the type hierarchy and by taking subgraphs of an original graph. Generalizations are also partially ordered in the conceptual graph model. These corresponding elements define how the Analogical Tool forms and orders analogies which are seen as minimal common generalizations, and also allows for analogical inferencing to be performed within a single framework. Thus, instantiation of the general framework using conceptual graphs results in a computational tool which can be seen to both form analogies and perform analogical inferencing. The inferencing is not an add-on explicit part of the system, rather it is an integrated extension of the existing framework.

3 The Analogical Tool

3.1 Minimal Common Generalizations

In developing the general framework, it was noted that most applications of analogy prefer minimal common generalizations as the "stronger" analogies. The conceptual graph formalism contains a notion of minimal common generalization which directly corresponds. Figure 3 gives a simple example of three ordered common generalizations which can be formed from the two analogues shown.

"Sam thinks that the house has a kitchen and that Ivan believes that there is a cat in the kitchen"

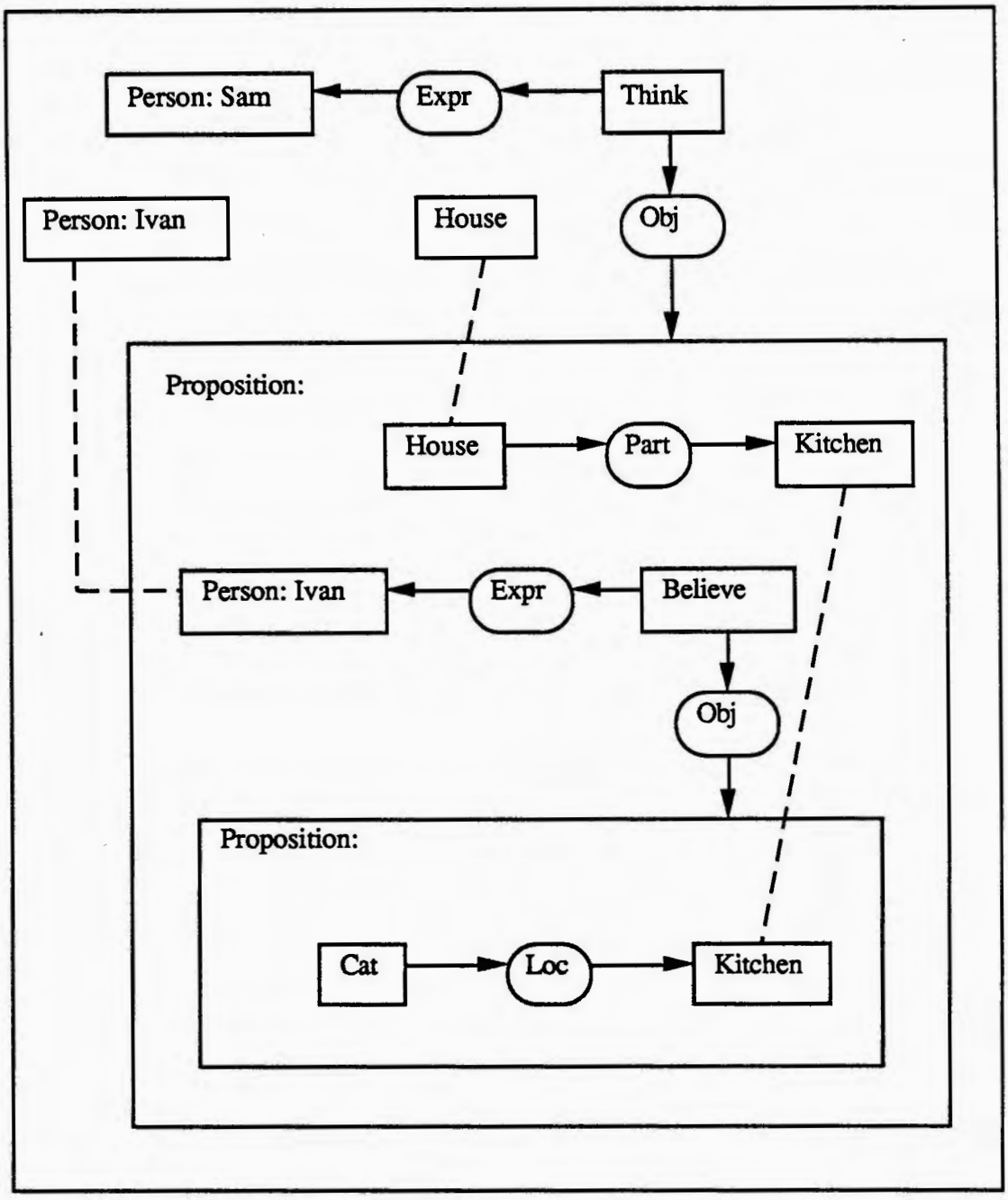

Figure 2 A Complex Sentence Represented in Conceptual Graphs

The lowest generalization on the page is the "stronger" minimal common generalization. It is the most specific of the three, first in concept restriction (from Animate to Person and from Act to Eat) and second by extent of the subgraph represented (it contains the maximum number of nodes and arcs). This minimal common generalization is the

generalization as low as possible in the generalization hierarchy - such that no other is more specific. These minimal common generalizations which the Analogical Tool derives:

1) Are maximally extensive since the generalization chosen for each of the corresponding concepts is the lower bound of all possible common generalizations.

2) Preserve linkages in the graphs which can be seen as minimal structural criteria for plausibility.

3) Are truth preserving, a necessary logical criterion for plausibility.

4) Minimize conflicts in the type hierarchy, a minimal semantic criterion for plausibility.

In forming the minimal common generalizations, the Analogical Tool uses four formation evaluations to search for the generalizations, and two ordering evaluations to establish which of these are minimal. Within the conceptual graph model, graphs are generalized in two ways. First, they are generalized by choosing subgraphs of an original graph and second, by using more general concepts according to the type hierarchy. The formation evaluations and ordering evaluations presented below correspond directly to these two types of generalization.

3.2 Formation Evaluations

There are four formation evaluations which work together to derive the common generalizations. Three of the four can be seen as working to decide what subgraphs are chosen for the resulting common generalization. These subgraphs must be common to both analogues, and they decide what parts of the analogues are put into correspondence. Any combination of subgraphs common to both analogues will form a generalization, but specific aspects of the conceptual graph model define what are the more plausible subgraphs. These aspects, which are the formation evaluations, also help to cut down on the computational complexity of the search by specifying that only certain subgraphs are considered for the resulting common generalization.

Same Relation Evaluation The first evaluation requires same relations between two pairs of potentially corresponding concepts. As shown in Figure 3 if [Girl] and [Person:Sue] are put into correspondence, then it is possible to put [Eat] and [Eat] into correspondence because they share the same (Agent) relation between them. If on the other hand Analogue 1 had contained [Girl] <- (Expr) <-[Think] instead, even though it would normally be possible to put [Girl] and [Person:Sue] in correspondence and also [Think] and [Eat] in correspondence (because they are both acts), using this evaluation, they cannot be put into correspondence because they no longer share the same relation between them. This evaluation is a result of the projection operator which requires that the same relation exist between connected corresponding concepts.

This evaluation defines one constraint on the subgraphs chosen for the common generalization. It is a very strong constraint which permits only agents to match to agents, objects to match to objects, etc.

Co-reference Links Evaluation Co-reference links define an equality relation between concepts. Thus, they define another structure which must be maintained when choosing subgraphs for the common generalization. If two concepts which both have co-referents are put into correspondence, then their co-referents can only be put into correspondence with each other - not with any other concepts. For example, if A=B and C=D where A, B, C and D are concepts in a conceptual graph and = represents the

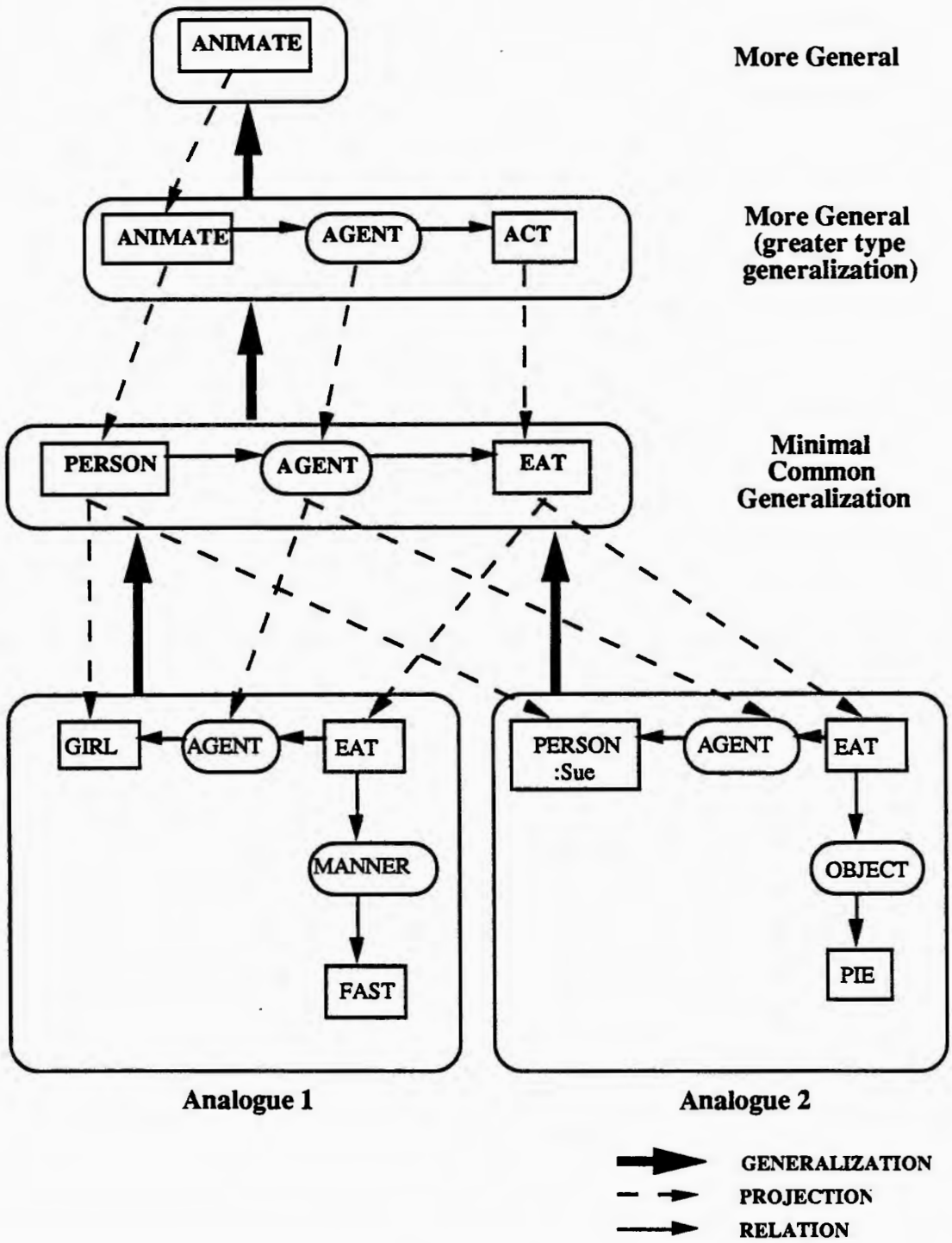

Figure 3 Forming Common Generalizations

co-reference link, then putting A and C into correspondence requires that B can only be put in correspondence with D or with nothing. Breaking this evaluation means the equality relation between concepts is being invalidated. The co-referent subgraph is a very strict evaluation that must be maintained.

Proposition Evaluation Proposition concepts which contain conceptual graphs in their referent fields define a *contains* relation between the proposition concept and the graphs within it. This relation defines yet another subgraph evaluation. This subgraph evaluation requires that if proposition concepts from two analogues are put into correspondence, then their respective contained graphs are candidates to be put into correspondence only with each other. The subgraph chosen here for the common generalization is the proposition concept with some or none of its contained graphs.

Comparable Concepts Evaluation This fourth evaluation conforms to the second way a graph can be generalized; that is by climbing the type hierarchy. Concepts are comparable if there exists a common generalization of them in the type hierarchy that is not the universal type. This evaluation uses the notion of type abstraction to determine if concepts are comparable and can be generalized. This evaluation works in conjunction with the part-of, subgraph evaluations by determining if specific concepts within a chosen subgraph can be generalized. Generalization using the type hierarchy also specifies particular general concepts for the resulting common generalization, as shown in Figure 3. There we see Person as a generalization for [Girl] and [Person:Sue] for example. In this way a common generalization results in the kind of general problem solving abstraction used by researchers such as Gick and Holyoak [1983] and Greiner [1985].

3.3 Ordering Evaluations

The formation evaluations are a consequence of specifying an analogy as a common generalization. Specification of "stronger", more plausible analogies as minimal common generalizations has implications for ordering them. The two ordering evaluations order the generalizations and conform to the generalization criteria for conceptual graphs in much the same way the formation evaluations do. The first ordering evaluation conforms to the subgraph generalization criterion. For minimal common generalizations, the subgraphs chosen are the maximally extensive or most specific common generalizations. The second ordering evaluation conforms to generalization using the type hierarchy. Minimal common generalizations are also those which choose the lowest common supertypes for each pair of corresponding concepts as defined by the type hierarchy.

Deriving minimal common generalizations according to the first ordering evaluation amounts to ordering the common generalizations by their *extent*. The second ordering evaluation, *semantic distance*, uses the distance climbed in the type hierarchy when searching for a common supertype for corresponding pairs of concepts.

Using these ordering evaluations, as the common generalizations are formed, it can be determined if they are minimal common generalizations so far. With a complete search of all analogies, the evaluations derive the set of minimal common generalizations or "stronger" analogies.

3.4 Analogical Inferencing

The Analogical Tool described thus far forms analogies which are characterized as minimal common generalizations. These analogies are useful for many purposes such as schema abstraction [Gick and Holyoak,1983]. Other uses of analogy require that further reasoning take place from the analogies. This reasoning is referred to as analogical inferencing.

Performing analogical inferencing consists of extending a target analogue by hypothesizing that new concepts, relations and arcs exist in it, based on the existence of similar concepts, relations and arcs in the source analogue. These inductive hypotheses result in the formation of a new minimal common generalization which contains a generalization of these new hypothesized parts and the source parts they are inferred from.

The formation evaluations used by the Analogical Tool can help determine which hypotheses to make and also determine which hypotheses are valid. Although parts could be hypothesized from either of the analogues to the other, the tool only hypothesizes parts as existing in the target analogue based on their existence in the source analogue.

The Extensibility Principle

When a choice must be made between two or more minimal common generalizations as to what inferencing is more plausible, the *extensibility principle* gives a resolution. If for instance one of the generalizations is not extensible at all and the other is, the extensible generalization is used because analogical inferencing can occur. If on the other hand, both are extensible, then the one which is the most extensible is preferred. Each of the extensions must still conform to the formation evaluations, but the most extensible generalization is the new minimal common generalization. We will still be deriving minimal common generalizations.

4 An Example from Gick and Holyoak - Schema Abstraction

The following example from Gick and Holyoak [1983] is discussed in detail to show how analogy formation takes place in the tool and how analogical inferencing proceeds. A result of analogy formation is production of a problem solving schema. Inferencing using this general problem solving schema can later be used to solve similar problems.

In Figure 4, Gick and Holyoak use as an example a military problem and an analogous radiation problem. In the military problem, a commander wants to capture a fortress using a large army but cannot send the entire army down one road because they will be noticed. The solution is to send small parts of the army along many roads simultaneously. These sentences are represented in the four conceptual graphs at the top of Figure 4. Co-referent links are shown by the very angled arcs connecting concepts with the same name. Context boxes representing propositions are the boxes which contain subgraphs within them.

In the analogous radiation problem, a doctor wants to use rays to destroy a tumor. The rays cannot all come from one direction without killing the patient. The analogous solution is to send low intensity rays from multiple directions simultaneously. These sentences are represented in the bottom four conceptual graphs containing dashed nodes in Figure 4.

These two situations as represented are given to the analogical tool, which forms the same analogy as that presented by Gick and Holyoak. The following concept correspondences are formed by the tool:

(Doctor - Commander) (Rays - Army) (Destroy - Capture) (Tumor - Fortress)

(Want - Want) (Possesses - Possesses) (Powerful - Large) (Go - Go)

(Direction - Road) (One - One) (WeakRays - Group) (Many - Many)

(Simultaneously - Simultaneously)

The tool also automatically forms an abstraction similar to that described by Gick and Holyoak, by using the concept hierarchy to find common generalizations of the concepts. The general problem solving principle formed by the tool is:

A Person wants to use a Mobile Entity to perform a Violent act on a Stationary Entity.

A Person has a Mobile Entity with an Attribute of Strength.

The Mobile Entity cannot go along one Path.

The Solution is for Parts of the Mobile Entity to Go along Many Paths Simultaneously

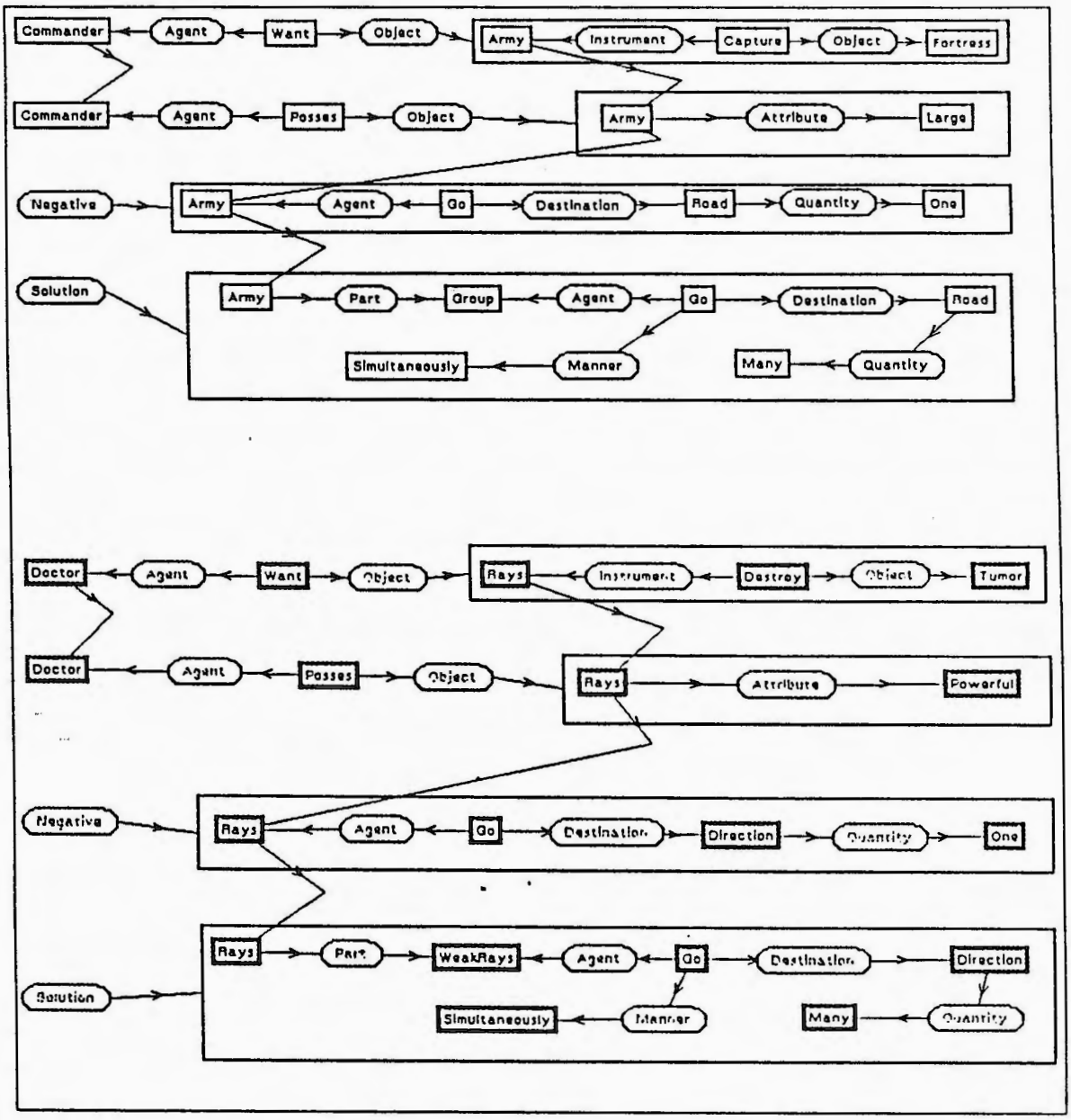

Figure 4 Gick and Holyoak - Schema Abstraction

This example can also be used to show how the tool can perform simple analogical reasoning. In this case the target analogue is the same with the exception that the solution to the radiation problem is not known. This is represented as an empty *Proposition* node attached to the *Solution* relation. When the tool reports the correspondences resulting from the best analogy formed, the *Solution* propositions from the target analogue is matched with the solution node of the abstracted schema. With substitution of corresponding terms inside the solution to the abstracted schema, a solution to the radiation problem is realized: to have parts of the rays go along many paths simultaneously.

5 Relevancy Limitations and Solutions

The Analogical Tool described thus far is a similarity based system and as such suffers from limitations due to relevancy issues. Drawing inferences from minimal analogies is an inductive process and thus potentially false. Use of the extensibility principle deals with some problems of relevancy and is similar to Gentner's [1983] use of the systematicity principle. But, analogical inferencing based on correspondence alone can be misleading if the major correspondences are not relevant to the objective of the inferencing. Russell [1987] defines one method of dealing with this problem using determinations. Determination rules are similar to dependency relations in database theories and are of the form: P determines Q. These rules are weaker than implications but serve as background knowledge which when added to the premises of an analogy, makes the conclusion follow soundly.

Current work on the Analogical Tool involves implementing the determinations rules using the type hierarchy. The type hierarchy appears to include much of the same information contained in the determination rules and a use of semantic distance over the type hierarchy can achieve the same effect. What we need to do when forming correspondences is to prefer and look for, correspondences which are semantically closer.

6 Summary

Analogy can be widely used in the knowledge acquisition cycle, in particular for schema abstraction and problem solving. This paper has described an analogical tool and shown its use in analogy formation, schema abstraction and analogical inferencing. Evaluation criteria have been derived for forming and ordering analogies by plausibility that relate naturally to operations on conceptual graphs.

References

[Gaines and Shaw, 1982] Brian Gaines and Mildred Shaw. Analysing Analogy. Trappl, R. Ricciardi, L. and Pask, G. Editors. Progress in Cybernetics and Systems Research Vol. IX. pp. 379-386. Washington: Hemisphere. 1982.

[Gentner, 1983] Dedre Gentner. Structure-Mapping: a Theoretical Framework for Analogy. Cognitive Science 7. 1983. pp. 155-170.

[Gentner and Seziorski, 1986] Dedre Gentner and M. Seziorski. Historical Shifts in the Use of Analogy in Science To appear in B. Gholson, W.R. Shadish, and A Graesser (Editors), Psychology of Science. 1986.

[Gick and Holyoak, 1983] Mary Gick and Keith Holyoak. Schema Induction and Analogical Transfer. Cognitive Psychology 15. 1983. pp.1-38.

[Hofstadter and Mitchell, 1988] Douglas R. Hofstadter and Melanie Mitchell. Concepts, Analogies and Creativity. Proceedings of CSCSI'88: Seventh Biennial Conference of the Canadian Society for Computational Studies of Intelligence. pp.94-101. 1988.

[Indurkhya, 1985] Bipin Indurkhya. A Computational Theory of Metaphor Comprehension and Analogical Reasoning. PhD thesis, University of Massachusetts, 1985.

[Lakoff and Johnson, 1980] George Lakoff and Mark Johnson. The Metaphorical Structure of the Human Conceptual System. Cognitive Science 4. 1980. pp. 195-208.

[Leishman, 1989] Debbie Leishman. An Analogical Tool; Based on Evaluations of Partial Correspondences over Conceptual Graphs. Masters Thesis. University of Calgary. 1989.

[Lenat et al., 1986] Doug Lenat, Maryank Prakash and Mary Shepherd. CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks. The AI Magazine. pp. 65-85.

[Murphy et al, 1963] Glenn Murphy, David J. Shippy and H. L. Luo. Engineering Analogies. Iowa State University Press. Ames, Iowa. 1963.

[Polya, 1954]. George Polya. Induction and Analogy in Mathematics. Volume I of Mathematics and Plausible Reasoning. Princeton University Press. Princeton. 1954.]

[Russell, 1987] Stuart Jonathan Russell. Analogical and Inductive Reasoning. PhD thesis. Stanford University. 1987.

[Sowa, 1984] John Sowa. Conceptual Structures. Addison Wesley Publishing Co. Reading Ma. 1984.

[Winston, 1981] Patrick Winston. Learning and Reasoning by Analogy. Communications of the ACM Vol. 23 No. 12 1980.

[Witten, & MacDonald, 1987] Ian Witten & Bruce MacDonald. Using Concept Learning for Knowledge Acquisition. Proceedings of the Second Knowledge Acquisition for Knowledge Based Systems Workshop. Banff, Alberta, Canada. 1987.