# Analogy as a Constrained Partial Correspondence Over Conceptual Graphs

**Debbie Leishman**

Knowledge Science Institute & Department of Computer Science
University of Calgary
Alberta, Canada T2N 1N4

## Abstract

This paper describes an analogical tool based on the formation of partial correspondences between knowledge structures represented as conceptual graphs. The formation of correspondences is controlled by evaluation criteria for ordering analogies by plausibility that relate naturally to operations on conceptual graphs. The tool and its underlying theory are tested empirically against critical examples of analogical reasoning in the literature. To clarify relations between the examples, reasoning patterns and knowledge representation scheme, an object-oriented implementation is used whose operation is simple and transparent.

## 1 Introduction

Analogical reasoning is based on the formation of a plausible partial correspondence between knowledge structures. The problem for computational emulations of analogy is to determine what these knowledge structures are and what constitutes a plausible correspondence. There are a variety of applications of analogical reasoning which can help direct the search for a solution to this problem. Analogy is used in metaphor understanding [Indurkhya, 1985, Lakoff and Johnson, 1980], in scientific exploration [Gentner, 1987, Hesse, 1966], in argumentation and rhetoric [Hesse, 1966], in schema abstraction for problem solving [Gick and Holyoak, 1983] and for teaching [Murphy et al., 1963]. Examples of these different applications of analogy, particularly problematic cases, can serve as test cases for a computer emulation of analogy.

This paper presents an analogical tool which embodies critical functions of analogical reasoning derived from the analysis of previous research in the field. It is shown that the choice of a conceptual graph model for knowledge representation ensures that these functional requirements are met and also brings with it certain evaluation criteria which constrain the search for analogies. The tool is described in three ways, firstly in terms of the conceptual graph formalism, secondly in terms of the evaluation criteria contained within it, and thirdly, in terms of the basic algorithm. The final section discusses how well the analogical tool performs on examples from the literature and shows it to be effective in the derivation of analogies and in reasoning based on them. The simple modular structure of the algorithm clarifies the relations between the argument forms of analogical reasoning and general properties of the knowledge representation schemes used.

## 2 Analysis of Prior Studies of Analogy

Analysis of colloquial connotations and psychological processes of analogy has generated research which can be classified as follows:

1) Abstract theories of analogy;

2) Psychological models of analogy;

3) Evaluation criteria for analogy.

Analyzing works in each of these areas results in a definition of critical functions essential to the notion of analogy. These critical functions can be found in many computational emulations of analogy although often they are not explicitly stated.

*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. 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. Lakoff and Johnson [1980] argue that the human conceptual system is comprised of a limited number of non-metaphorical base concepts supplemented by metaphorical concepts based on experiential gestalts. These gestalts can be thought of as analogues, and a partial correspondence formed over terms they contain allows us to understand metaphors such as ARGUMENT IS WAR. From this psychological model, Lakoff and Johnson note that the most difficult part of constructing a computational emulation of metaphor understanding will be in properly representing the knowledge contained in the experiential gestalts.

*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]. Criteria based on connectivity can be found in Indurkyha's [1985] AST theory and in Gentner's [1983] systematicity principle. Winston's [1981] early paper on analogical reasoning contains many useful criteria such as the use of case relations and matching of terms which share the same relations between them; this latter criterion is also evident in Gentner's work.

This three-way analysis leads to the following critical functions which are necessary for a computational emulation of analogy:

1) Algorithms for finding partial correspondences between abstract structures;
2) Semantic schemas for representing knowledge of the domains involved in the analogies;
3) An evaluation relation for partially ordering analogies by plausibility.

The analogical tool described in this paper implements each of these functions as well defined modules. The evaluation criteria arise naturally from the knowledge representation scheme chosen, namely conceptual graphs. These criteria include those used by other researchers. The resulting algorithm clearly defines the rationale for the analogies formed, and is readily tested on examples of analogy taken from the literature.

## 3 Conceptual Graphs

The semantic schema requirement above is satisfied in this work by the use of the conceptual graph model as described by Sowa [1984]. 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 [Sowa, 1984]. 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 1. 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

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

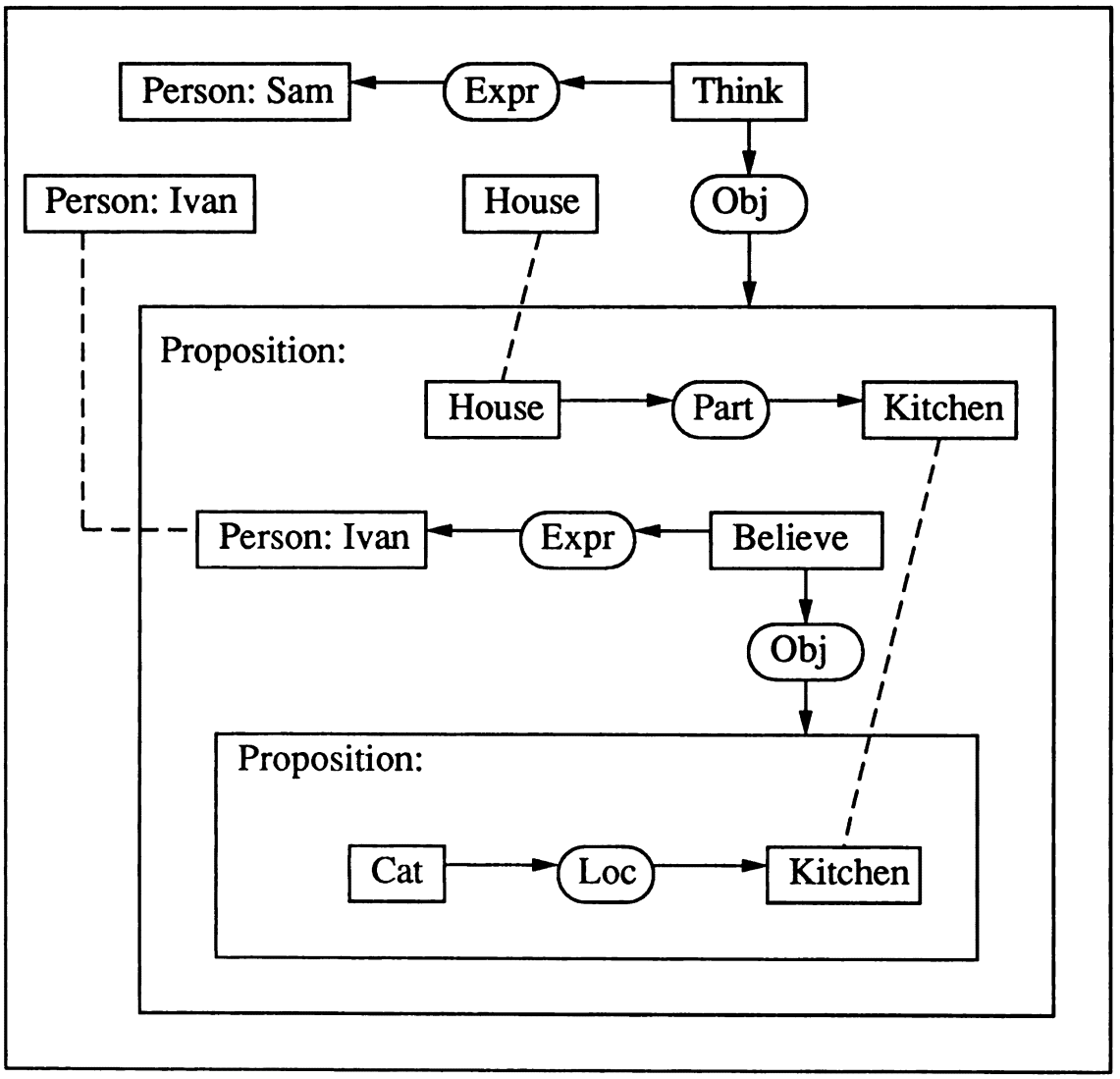


**Figure 1 Conceptual graph for a complex sentence**

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 1 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 1 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.

### 3.1 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 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.

### 3.2 Conceptual Graphs for Analogy

There are three reasons why conceptual graphs are a good choice for the knowledge representation scheme of an analogy emulation program:

1) They have a direct mapping to predicate calculus and thus can be incorporated into any reasoning system based on logic [Sowa, 1984];
2) They contain constraints which constitute evaluations for forming partial correspondences as well as an evaluation relation for partially ordering analogies by plausibility;
3) Their straightforward mapping to and from natural language allows for a linguistic description of analogues.

## 4 Implementing the Analogical Tool

This section outlines the implementation of the analogical tool in terms of the conceptual graph operations used and the resulting specifications for algorithms.

### 4.1 Minimal Common Generalizations

In the conceptual graph formalism, the natural partial correspondence on which to base the formation of an analogy is the *minimal common generalization* of the graphs corresponding to two or more analogues. A common generalization specifies a graph which is a generalization of all analogues in an analogy. There can be more than one of these generalizations and as stated previously, they will be ordered in a generalization hierarchy. Figure 2 gives a very simple example of three ordered common generalizations which can be formed from the two analogues shown.

The lowest generalization on the page is the minimal common generalization. It is the most specific of the three, first in concept restriction (from *Animate* to *Person*) and second in extent (it contains the maximal number of nodes and arcs). A minimal common generalization is a generalization as low as possible in the generalization hierarchy —such that no other is more specific.

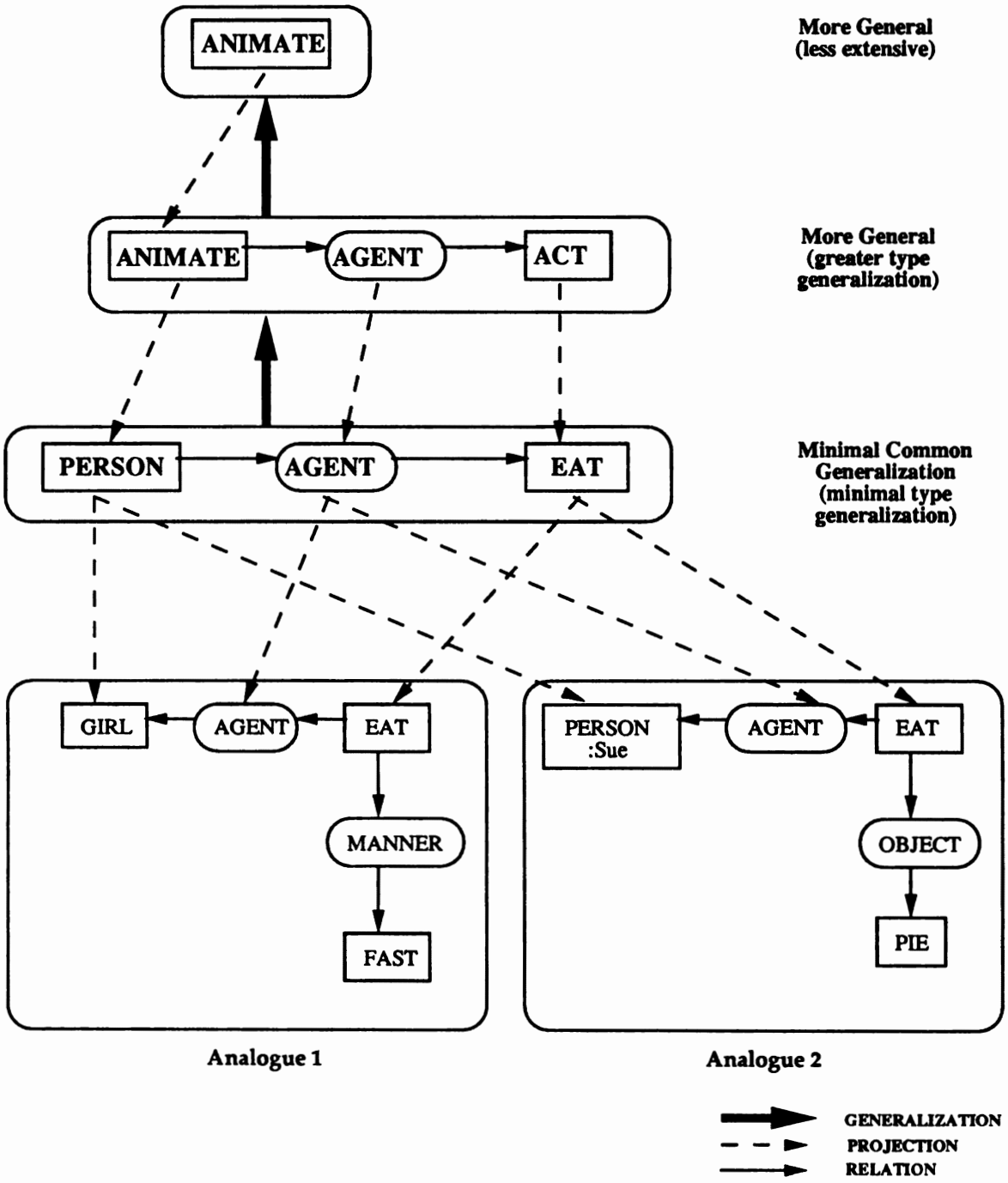


Figure 2 Forming Common Generalizations

It is these minimal common generalizations that the analogical tool derives. They satisfy requirements 1) and 3) specified in Section 2, that is of providing: a basis for algorithms for finding partial correspondences; and an evaluation relation for partially ordering analogies by plausibility.

A minimal common generalization:

- Is maximally extensive since the generalization chosen for each of the corresponding concepts is the lower bound of all possible common generalizations;
- Preserves linkages in the graphs which can be seen as minimal *structural* criterion for plausibility;
- Is truth preserving, a necessary *logical* criterion for plausibility;
- Minimizes conflicts in the type hierarchy, a minimal *semantic* criterion for plausibility.

There will typically be many generalizations of two analogues, a subset of which will be minimal. The analogical tool uses four formation evaluations to search for generalizations, and two general ordering evaluations to establish which of these are minimal. These are described in the following sub-sections.

### 4.2 Formation Evaluations

Four formation evaluations arise from the generalization criteria, and reduce search in the analogy operation. These evaluations are used to form the common generalizations.

The first two were presented above; firstly the requirement of the same relations between two pairs of potentially corresponding concepts; and secondly the requirement of type comparability. Two concepts will be comparable if there exists a common generalization for them which is not the absurd type. For example, an event and an attribute will not be comparable concepts, but two entities will be. The third and fourth formation evaluations emerge as analogues become compound graphs with co-reference links and nested context boxes.

Co-reference links form chains of equality between concepts in a graph. For the third evaluation, these chains of equality must be maintained when forming the common generalizations and putting concepts into correspondence.

The fourth evaluation requires that referents of corresponding proposition concepts (i.e. graphs) must be compatible. This means that if projections put two proposition boxes into correspondence in the analogy, then their referent subgraphs are candidates to be put into correspondence only with each other.

### 4.3 Ordering Evaluations

The formation evaluations above are a consequence of specifying an analogy as a common generalization. Specification of the more plausible analogies as *minimal* common generalizations has implications for ordering them. A minimal common generalization is maximally extensive. Because of the partial nature of the generalizations formed in analogy, this maximal extent becomes a measure of connectivity. The first general ordering evaluation is a measure of this connectivity.

This first evaluation, connectivity, can be subdivided into three specific components. The first, **Extent**, measures how much of the target domain has been matched with the source domain. A measure of 0 reflects a total match. This ensures that only the most extensive analogies are considered. The second component evaluates the **Kind-Of** connectivity that results in the most extensive analogies. Preference is given to long chains of connectivity and a high score here will indicate more connectivity and less fragmentation. The third component is a measure of **Extendability**. Given two equal analogies as ordered by the first two components of connectivity, the one with a greater capability of being extended through analogical reasoning is preferred. An example of this measure appears in the section on testing the analogical tool.

Minimal common generalizations are also those which choose the lowest common supertypes for each pair of corresponding concepts as defined by the concept lattice. A higher preference is given to analogies with the lowest semantic distance between corresponding concepts. The second general ordering evaluation says that for two analogies which are equally connected, the better analogy is determined by minimizing the distance climbed on the concept lattice for each pair of comparable concepts. This amounts to measuring the semantic distance of concepts according to this lattice. Concepts which are further apart correspond to weaker analogies.

### 4.4 Correspondence and Projections

The formation of a minimal common generalization of two analogues involves putting the concepts and relations in one analogue in correspondence with those in the other, and hence in the generation of an analogy. This correspondence again has a natural interpretation in conceptual graphs as arising from a pair of *projections*.

As shown in Figure 2, formation of a common generalization results in projections from the generalization into each of the analogues (these are shown by the dashed lines). A projection maps graphs at higher levels of the generalization hierarchy into ones at lower levels. A projection is defined as follows [Fargues, 1986]:

Let **u** and **v** be two conceptual graphs. If **u <= v**, there exists a subgraph **u'** joined with some additional edges. The induced graph **u'** is called the projection of **v** in **u**. A graph **v** may have several distinct projections in the same given graph **u**.

The projection operation consists of finding a subgraph **u'** of **u** which satisfies the following conditions:

1) The conceptual relations in u' and v are identical;
2) The concepts c1, . . ., cn of u' are some restrictions (i.e. specializations) of the corresponding concepts d1, . . ., dn of v, as defined by the concept lattice;
3) If a relation r links two concepts di and dj in v, then it also links the concepts ci and cj in u'.

### 4.5 Object-Oriented Formulation of the Algorithm

The analogical tool is written in the object oriented language Smalltalk [Goldberg and Robson, 1983 ] in such a way that each of the formation and ordering evaluations exist as well defined, identifiable modules. The basic algorithm searches for analogies, with the formation evaluations serving as constraints that prune branches of the search tree; and the ordering evaluations ordering the analogies. Input consists of two analogues expressed as conceptual graphs. Output consists of displays of the analogies as lists of concept correspondences, together with their measures.

As analogies are being formed, potential concept matches are considered. These must pass all formation evaluations before the concepts can be put into correspondence with each other. For example, the requirement of same relations between concepts is a very strong constraint and permits only agents to match to agents, objects to objects, etc.

Only the most extensive analogies as measured by the **Extent** connectivity metric are considered further. These are then ordered using the **Kind-Of** connectivity metric. Any ties in ordering at this point are broken using the semantic distance of concepts metric.

At this point, if an analogy is **Extendable**, analogical reasoning can take place. This reasoning attempts to maximize the connectivity measure and minimize the semantic distance measure while adhering to the formation evaluation constraints. Examples of the tool in operation in the following section clarify these points.

## 5 Testing the Analogical Tool

The analogical tool has been tested on various examples of analogy from the literature. Examples have been taken from the CYC project [Lenat et al, 1986], from Gentner's work [1983], from Indurkyha's PhD thesis [1985], from Russell's PhD thesis [1985] and from Gick and Holyoak [1983]. Two of these examples are presented here. They show that the analogical tool performs effectively, using evaluations which are a result of the knowledge representation scheme chosen.

### 5.1 Example I

The first example, shown in Figure 3, is taken from work on schema abstraction by Gick and Holyoak [1984] and will be used in two ways by the analogical tool: first to form a general problem solving principle, and second to perform simple analogical reasoning.

In the first case, two analogous problems and their solutions are presented. An analogy is drawn between the two domains then abstracted to obtain a general problem solving principle. This abstraction can later be used to solve similar problems. 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

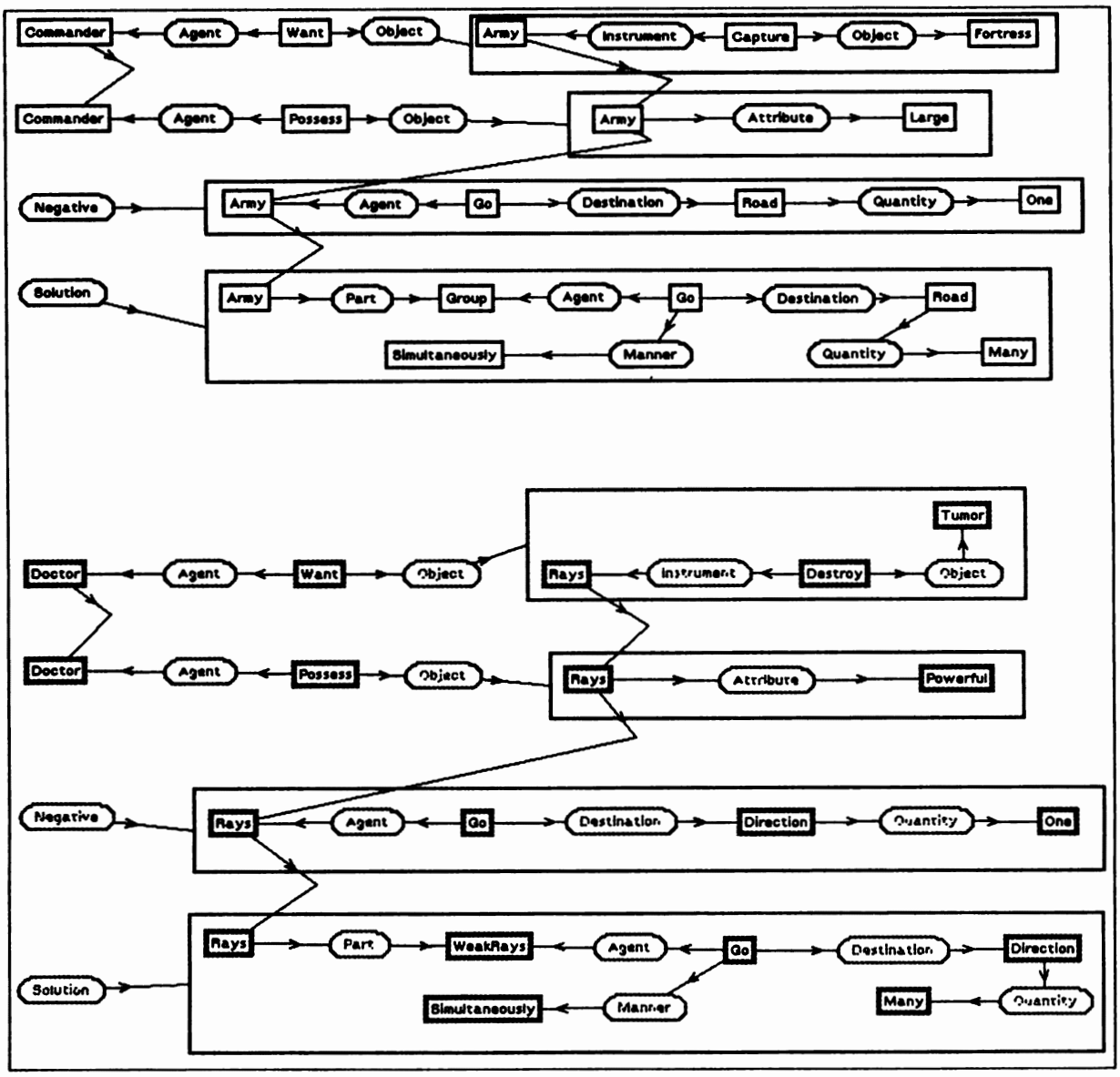


**Figure 3 Example of Schema Abstraction**

at the top of Figure 3. 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 3.

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 | Possess.....Possess |
| Powerful.....Large | Go.....Go |
| Direction.....Road | One.....One |
| WeakRays....Group | Many.....Many |

The tool also automatically forms an abstraction similar to that described by Gick and Holyoak, by using the concept lattice 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

Note that, although the sentences of the analogues appear in the same order in Figure 3, the tool will still find the analogy if the order of the sentences is changed.

This example can also be used to show how the tool can perform simple analogical reasoning. In this case the analogues are 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 two *Solution* propositions are matched together. With substitution of corresponding terms inside the solution to the military problem, a solution to the radiation problem is realized: to have parts of the rays go in many directions simultaneously.

### 5.2 Example II

Example two, shown in Figure 4, is taken from Gentner's description of a heat-water analogy used by the French thermodynamicist Carnot [Gentner, 1987]. This example is very significant because it contains a pitfall for over-simplistic analogy formation processes. The problem arises because two equal matches occur. An analogy formation algorithm must be able to determine which is the more relevant one. Use of the *systematicity principle* in Gentner's Structure-Mapping program results in the correct choice. For the analogical tool described in this paper, the more relevant match is correctly detected by the **Extendability** measure.

At the top of Figure 4 is a representation a person might have of a water flow situation. The person knows that if the pressure being exerted on water in a beaker is greater than the pressure being exerted on water in a vial, and if there is a fluid path from the beaker to the vial via a pipe, then this will result in a flow of the water through the pipe from the beaker to the vial. The person will also have other knowledge about the situation such as the diameter of the beaker being greater than the diameter of the vial.

The graph at the bottom of Figure 4, represented by dashed nodes with connecting arcs, is a description of what that same person might know about heat flow. In this case the person knows that the coffee is hotter than the ice-cube and that heat will flow from the coffee to the ice-cube. The person does not know that the heat flow is a result of the difference in heat between the ice-cube and the coffee.

These two analogues are presented to the analogical tool. In this example the tool forms two analogies with equal measures. Both analogies have formed the same correspondences between concepts except that in one *Temperature* has been matched to *Pressure*, and in the other *Temperature* has been matched to *Diameter*. Both of these analogies have the same connectivity rating as given by the **Extent** and **Kind-Of** connectivity measures.

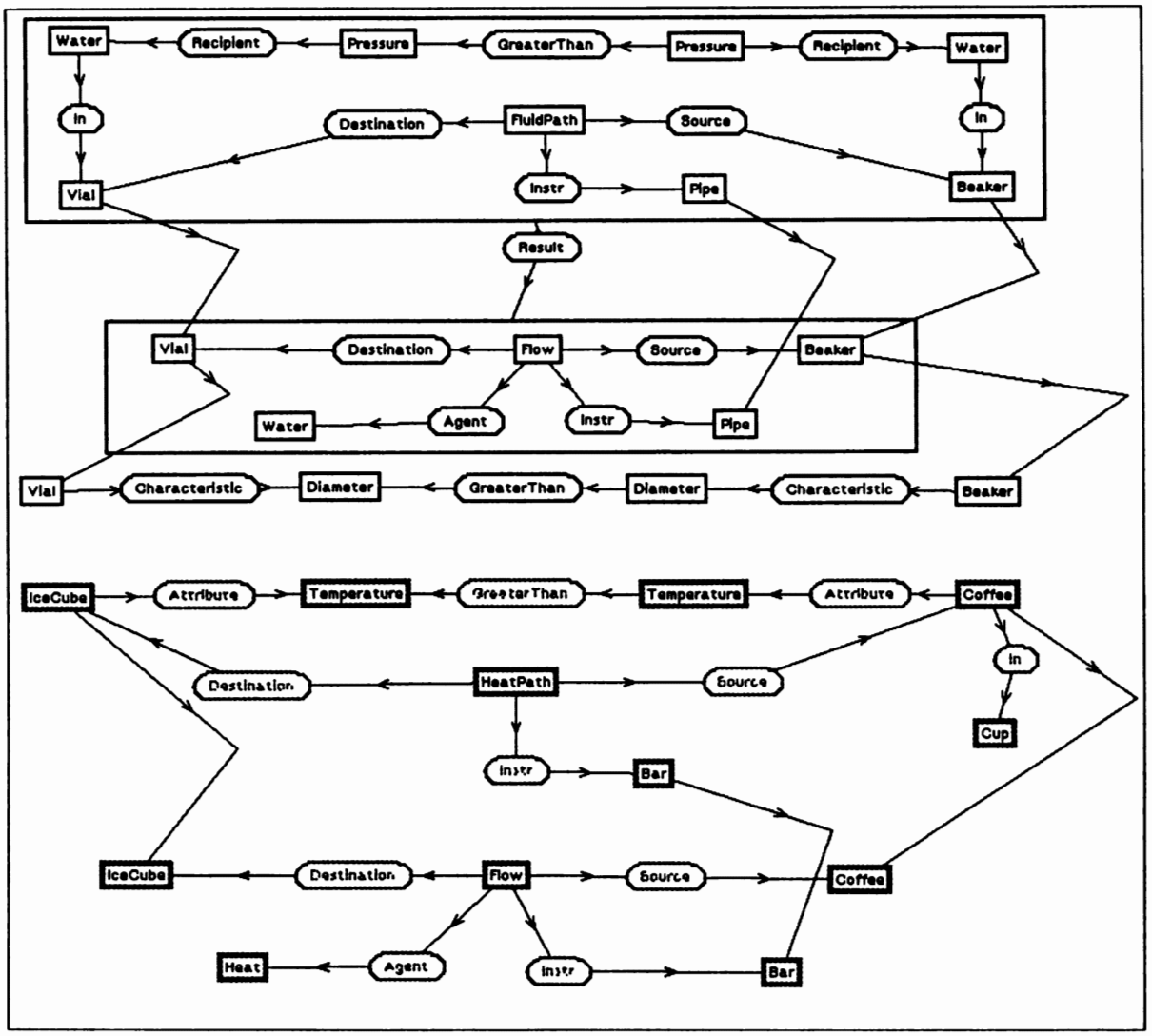


**Figure 4 Example of A Heat-Water Analogy**

Using the **Extendability** measure in the analogy which matches *Temperature* with *Pressure*, it is possible to hypothesize both of the context boxes from the source water domain as well as the result relation between them over to the target heat domain. This hypothesis will be consistent with all formation evaluation criteria. In contrast, the analogy which has matched *Temperature* with *Diameter* can only hypothesize the context box which surrounds the flow concept and its relations. It is not possible to hypothesize the other context box as existing without breaking the proposition evaluation criteria. Thus the **Extendable** ordering evaluation not only answers whether an analogy is extendable or not but can also help with decisions about which hypotheses are best to make when performing analogical reasoning.

The result of this use of **Extendability** is very similar to that obtained by Gentner's use of the systematicity principle. The difference is that rather than being an explicit driving principle like systematicity, the **Extendability** measure is more of an emergent property. This property is part of a system which has a strategy of maximizing ordering evaluations while adhering to the formation evaluation constraints.

## 6 Summary and Conclusions

This paper has described an analogical tool based on the formation of partial correspondences between knowledge structures represented as conceptual graphs. Evaluation criteria have been derived for ordering analogies by plausibility that relate naturally to operations on conceptual graphs. These have been tested empirically against critical examples of analogical reasoning in the literature.

The algorithms are written in Smalltalk to give a tool whose operation is simple and transparent so the essential relations between the examples, reasoning patterns and knowledge representation scheme are apparent.

The use of conceptual graphs for knowledge representation has been beneficial:

i) in allowing examples of analogical reasoning in the literature expressed in natural language to be simply translated to a computational representation;

ii) in providing a principled algebraic formulation of the essential operations of analogy formation and analogical reasoning that relates simply and naturally to the semantic constraints found in cognitive studies of analogy.

The notions of *analogy* and the relative *plausibility* of analogies are cognitively rich and it is not clear that any computational framework can capture them in full. There is certainly a very strong dependence on the underlying knowledge representation scheme that is used. The results of this study indicate that conceptual graphs naturally support the mathematical operations of forming partial correspondences and provide evaluation criteria that are strong enough to result in plausible analogies that correspond to psychological expectations.

## Acknowledgements

This work has been partially supported by the Natural Sciences and Engineering Research Council of Canada. I am grateful to the anonymous referees for their perceptive criticism; to John Sowa for helpful comments and access to his own research; and to Brian Gaines, Rosanna Heise, Brent Krawchuk, Mildred Shaw and Ian Witten for critical suggestions that have improved this paper.

## References

[Fargues, 1986] Jean Fargues. Conceptual Graphs for Semantics and Knowledge Processing. IBM Journal of Research and Development. Vol. 30 No. 1 1986.

[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, 1987] Dedre Gentner. Mechanisms of Analogical Learning. Report No. UIUCDCS-R-87-1381. Department of Computer Science. University of Illinois at Urbana-Champaign. 1987.

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

[Goldberg and Robson, 1983] Adele Goldberg and David Robson. Smalltalk-80 The Language and its Implementation. Addison Wesley Publishing Co. Reading, Ma. 1983.

[Hesse, 1966] Mary Hesse. Models and analogies in Science. Notre Dame, IN: University of Notre Dame Press. 1966.

[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.

[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.

[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.