Abstract

Analogy is one reasoning method used extensively and effectively by humans. Because analogy is so widely used, it is felt that giving computers this ability will make them more useful in supporting and emulating human reasoning. This thesis describes the development of a principled Analogical Tool which deals with this issue.

The Analogical Tool results from instantiation of a general framework for analogy. This framework is based on 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 supports analogical inferencing based on them.

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 formation evaluations operate to determine how and what parts of the analogues are generalized and the ordering evaluations partially order the generalizations. Analogical inferencing based on the tool proceeds by hypothesizing new information in a target analogue based on information in a similar source analogue. This inferencing results in extension of the target analogue and formation of a new minimal common generalization.

Testing of the Analogical Tool on examples of analogy taken from significant research, shows that the tool compares well with other systems. Examples range from simple analogy formation, through schema abstraction to metaphor understanding. Further critical evaluation of the tool shows limitations arise due to relevancy issues. These relevancy limitations are intrinsic to systems using inference based on similarity. An extensibility principle is introduced to deal with some relevancy limitations in inferencing based on the tool. Other relevancy issues are dealt with by the introduction of determination rules from Russell [Russell, 1987].

Acknowledgements

My sincerest thanks and appreciation go first to my advisor Brian Gaines. His guidance and patience have made this thesis possible. Somehow there was always time in his schedule and just the right insight at just the right time.

This research also owes much to those in the KSI lab; Mildred Shaw, Ian Witten, Dan Freedman, Saul Greenberg, Dave Maulsby, Darrell Conklin, Maurice Sharp, Dan Mo, Brent Krawchuk, Brian Woodward, Craig Jackson, Thong Phan, David Pauli and especially Rosanna Heise. They have attended many seminars and given many helpful suggestions that have contributed to this thesis. Beyond this they have all been morale boosters when it was most needed.

Thanks also goes to my constant companion Sadie who was there during many long hours at the computer. I would also like to thank my family who have always felt I could do this even when I wasn't sure. Last but certainly not least there is Richard. His support and understanding have not only helped me complete this thesis but have made it an enjoyable and worthwhile experience.

Contents

Approval ii
Abstract iii
Acknowledgements iv
Contents v
List of Figures viii

1 An Introduction to Analogy 1
1.1 The Meaning of Analogy 1
1.1.1 A General Definition 2
1.1.2 Uses of Analogy 2
1.2 Analogy and Purpose 5
1.3 Aim and Objectives 7
1.3.1 Aim 7
1.3.2 Objectives 7
1.4 Meeting the Objectives - A Synopsis of the Thesis 8

2 Analyzing Previous Work on Analogy 10
2.1 A Classification for Analogy 10
2.1.1 Psychological Models of Analogy 12
2.1.2 Mathematical Theories of Analogy 18
2.1.3 Evaluation Criteria for Analogy 21
2.2 A General Framework for Analogy 24
2.3 Technical Addendum to Chapter 2 30

3 Conceptual Graphs 34
3.1 Simple Graphs 34
3.1.1 Concepts and Relations 35
3.1.2 The Type Lattice 35
3.1.3 Background Knowledge 37
3.1.4 A Logical Mapping 38
3.2 Compound Graphs 38
3.2.1 Propositions 38
3.2.2 Co-reference Links 40
3.2.3 Negation, Modality and Tense 40
3.2.4 A Logical Mapping For Compound Graphs 41
3.3 Specialization and Generalization 43
3.3.1 Specialization 44

3.3.2 Generalization 46
3.4 Conceptual Graphs for Analogy 48

4 The Analogical Tool 50
4.1 Minimal Common Generalizations 51
4.1.1 Correspondence and Projections 53
4.2 Analogy Formation 53
4.2.1 Formation Evaluations 54
4.2.2 Ordering Evaluations 56
4.3 Analogical Inferencing 58
4.3.1 Analogical Inferencing in the General Framework 58
4.3.2 Analogical Inferencing in the Analogical Tool 58
4.4 Object-Oriented Formulation of the Algorithm 62
4.4.1 An Object-Oriented Analogical Tool 62
4.4.2 About the Algorithm 63
4.4.3 PseudoCode for the Analogical Tool 65

5 Testing the Analogical Tool 69
5.1 Example 1 - A Simple Teaching/Irrigating Analogy - CYC Project 69
5.1.1 Analogy Formation 70
5.1.2 Comparison of CYC to the Analogical Tool 73
5.2 Example 2 - Schema Abstraction - Gick and Holyoak 73
5.2.1 Analogy Formation 74
5.2.2 Schema Abstraction 76
5.2.3 Analogical Inferencing 77
5.2.4 Comparison of Gick and Holyoak With the Analogical Tool 80
5.3 Example 3 - The Heat/Water Analogy - Dedre Gentner 82
5.3.1 Analogy Formation 83
5.3.2 Analogy Inferencing 86
5.3.3 Comparison of Gentner's Theory With the Analogical Tool 86
5.4 Example 4 - The Sky is Crying Metaphor - Bipin Indurkhya 87
5.4.1 Analogy Formation 87
5.4.2 Comparing Indurkhya's AST Theory With the Analogical Tool 92
5.5 Summary 94

6 Relevancy Limitations and Solutions 96
6.1 The Heat/Water Analogy Revisited - Dedre Gentner 96
6.1.1 Analogy Formation 98
6.1.2 Analogical Inferencing 100

6.1.3 Comparison of Gentner's Theory With the Analogical Tool..... 101
6.2 Determinations - Stuart Russell..... 102
6.2.1 Problems in the Correspondence Base of the Analogical Tool..... 103
6.2.2 Using Russell's Determinations in the Tool..... 103
6.3 Summary..... 109
7 Conclusions and Further Work..... 110
7.1 Summary and Achievements..... 110
7.2 Discussion: The General Framework and the Analogical Tool..... 111
7.3 Discussion: Addressing Relevancy Limitations..... 114
7.4 Further Enhancements to the Analogical Tool..... 115
References..... 119

List of Figures

Figure 1.1 Terms in Analogy..... 3

Figure 2.1 A Classification for Analogy..... 11
Figure 2.2 Diagrammatic Example of the source Water Analogue..... 13
Figure 2.3 Analogy as A Common Generalization..... 25
Figure 2.4 A General Framework for Analogy..... 27
Figure 2.5 Analogical Inferencing in the General Framework..... 29
Figure 2.6 Diagram Defining Generalization..... 31
Figure 2.7 Analogy as Mutual Generalization and the Implied Partial Correspondence..... 32
Figure 2.8 Extensibility of an Analogy..... 33

Figure 3.1 Before Adding Wild-Feline..... 36
Figure 3.2 After Adding Wild-Feline..... 36
Figure 3.3 A Conceptual Graph of a Complex Sentence..... 39
Figure 3.4 Various Boolean Combinations..... 45
Figure 3.5 Two Canonical Graphs..... 45
Figure 3.6 Restriction of the Second Graph in Figure 3.5..... 45
Figure 3.7 Join of the Two Graphs in Figure 3.6..... 45
Figure 3.8 Simplification of Figure 3.7..... 45
Figure 3.9 A Common Generalization v and a Common Specialization w..... 47

Figure 4.1 Forming Common Generalizations..... 52
Figure 4.2 A Portion of the Type Hierarchy..... 68

Figure 5.1 Input Analogues to the CYC Example..... 71
Figure 5.2 Output from the CYC Example..... 72
Figure 5.3 Input Analogues to the Gick and Holyoak Example..... 75
Figure 5.4 Output from the Gick and Holyoak Example..... 76
Figure 5.5 Input Analogues to Gick and Holyoak Example Showing Inferencing..... 78
Figure 5.6 Use of the Type Hierarchy for Labelling of Concepts..... 81
Figure 5.7 Input Analogues to the Gentner Example..... 84
Figure 5.8 Output from the Gentner Example..... 85
Figure 5.9 Input Analogues to the Indurkhya Example..... 89
Figure 5.10 Input Analogues to the Indurkhya Example..... 90

Figure 5.11 Output from the Indurkhya Example 91
Figure 5.12 Output from the Indurkhya Example 92

Figure 6.1 The Heat/Water Analogy from Gentner 97
Figure 6.2 Input Analogues for the Gentner Example. 99
Figure 6.3 Output from the Gentner Example..... 100
Figure 6.4 Input Analogues for the Russell Example..... 104
Figure 6.5 Output from the Russell Example 105
Figure 6.6 Determinations as Focus of Assertion in the Russell Example. 107
Figure 6.7 A Portion of the Type Hierarchy 108

Figure 7.1 An Overview of the Thesis..... 112

Chapter 1
An Introduction To Analogy

The use of analogy pervades everyday situations and provides one mechanism for reasoning when classical deductive methods are insufficient. Analogies drawn between two or more systems, *the analogues*, identify similarities and differences between them, for their own sake or for purposes of plausible inferencing. Often, just the explication of the similarities and differences is the main purpose behind formation of an analogy, such as when analogies are used for teaching or for providing better understanding of a subject. In other situations, formation of an analogy precedes the use of that analogy for further inductive reasoning. When reasoning is performed from an initial analogy, we typically think of one analogue, the *source analogue*, as providing plausible hypotheses about another analogue, the *target analogue*. In this case, what we know about the source analogue, combined with known similarities and differences between the source and target, provides plausible evidence for hypothesizing further information about the target. Analogical reasoning of this type is used by people to investigate new areas of scientific research or for reasoning using a similar case from past experience.

Because analogy is so widely used by people, it is felt that providing computers with this ability will make them more useful in supporting and emulating human reasoning. Research reported in this thesis addresses the issue by stating the development of an Analogical Tool which forms analogies and supports analogical inferencing based on them. The design of the tool takes a principled approach building on past work in analogy leading to the formulation of a general framework for analogy. Use of conceptual graphs [Sowa, 1983] for the knowledge representation scheme of the framework results in a computational tool which compares well with other computational emulations of analogy from the literature.

1.1 The Meaning of Analogy

This section presents dictionary definitions of analogy and clarification of terms, as well as actual uses of analogy. Dictionary definitions give a good idea of what analogy means, and

by seeing actual uses we arrive at a more complete understanding of the meaning of analogy.

1.1.1 A General Definition

The American Heritage Dictionary defines analogy as the following: 1) Correspondence in some respects between things otherwise dissimilar; 2) An inference that if two things are alike in some respects they must be alike in others. These definitions point to two distinct aspects of analogy; as a similarity structure and as a process which uses this structure. Figure 1.1 shows how these two definitions interact and defines terms used throughout the thesis. In this figure we see that analogy consists of the two elements specified above; a *static analogy* structure and the process of *analogical reasoning*. Analogical reasoning consists of two kinds of processes; *analogy formation* and *analogical inferencing*. In analogy formation, *static analogy* structures are formed and ordered. Particular to this thesis within analogy formation, schema abstractions or general problems solving principles can also be formed and metaphors understood. The static analogy structures can be further used in *analogical inferencing*. Here, hypotheses are generated in one analogue based on their existence in another. This inferencing again produces a static analogy structure which can be used for more *analogical reasoning* if required. Finally, as shown in Figure 1.1, *relevancy issues* can affect the *analogy formation* and *analogical inferencing* processes. Figure 1.1 gives a good definition of terms in analogy but gives little insight into how these definitions are used by humans. For this we need to look at when and how people use analogy.

1.1.2 Uses of Analogy

People use analogy in many different ways for many different purposes. Below are just some of the many uses:

case based reasoning - this refers to intra-domain analogies or analogies formed and reasoned over using various cases from a single domain as in the practice of law [Sridharan and McCarty, 1981]. In this use of analogy, the concepts and many of the relationships between the concepts are known a priori, which greatly simplifies the formation of an analogy. As an example of case-based reasoning in law, consider the case of Smith vs. Wesson which establishes a precedent for assault cases [Winston, 1980].

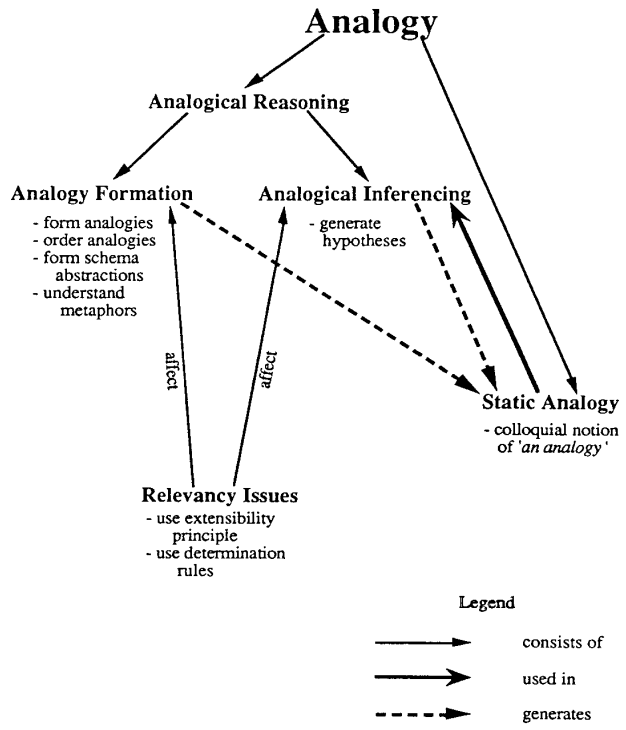


Figure 1.1 Terms in Analogy

Smith pointed a rifle at Wesson to frighten him. The rifle was not loaded, and was therefore harmless. Nevertheless, an assault has taken place because Wesson did not know that the rifle was not loaded. Subsequently, Smith versus Wesson is retrieved when the

case of Villain versus Victim is considered. Villain pointed a pistol at Victim in order to frighten him. The pistol was a harmless toy. Reasoning by analogy suggests asking if Victim knew that the pistol was harmless.

scientific investigation - This use of analogy has been well documented [Gentner, 1986; Hesse, 1966]. Here, similarities between disparate domains cause further similarities to be hypothesized. This in turn leads to testing of these hypotheses for their truth or falsehood. For example, French scientist Sardi Carnot extended an analogy between heat and water flow. Some commonality had already been observed between these, but it was Carnot who used analogy to hypothesize that the amount of heat transmitted between two bodies may be proportional to the difference between the temperatures of the two bodies. This is analogous to the known fact that the power produced by a waterfall is proportional to the amount of fallen water. This hypothesis was tested and found to be true.

argumentation/rhetoric - In this use of analogy, the listener is told of a partially filled analogy between a domain he or she is familiar with and another domain in which the speaker wishes to make a point. By continuing to add to the analogy, the speaker tries to convince the listener of the speaker's line of reasoning and final conclusion. The listener is almost forced to agree with the conclusion because of agreement with the analogy itself.

An example of the use of rhetoric for political purposes is given by the analogy that the relationship between father and child is the same in many respects as that between state and citizens [Hesse, 1966]. Firstly, the listener is reminded that the father is responsible for the maintenance, welfare and defense of a child in much the same way the state is responsible for these duties to its citizens. Building on this, it is argued that other relationships should also hold. The speaker then makes his final point - the citizen owes respect and obedience to the state in the same way a child owes respect and obedience to its father. If the listener has followed in agreement as the analogy was constructed, he is likely to be persuaded that he does owe respect and obedience to the state. Had a politician told the listener this without the above argument, he may have been less likely to agree.

metaphor - Metaphor is the use of analogy for communication. Typically, metaphors can not be observed in the real world, but force us to explore how the objects mentioned can be brought into correspondence [Russell, 1987]. Thus, behind all metaphors is a structural analogy [Indurkhya, 1985]. An example of a metaphor is - "The sky is crying".

We know that a sky does not really cry but we explore the metaphor to find its meaning. Here we are asked to find similarities between the domain sky and the domain crying in order to understand what it means for a sky to cry. In our search, we find that a person crying is similar to rain falling from the sky, and we understand the metaphor.

schema abstraction for problem solving - Once an analogy is drawn between domains, it is often abstracted and used to facilitate transfer of information [Gick & Holyoak, 1983]. Gick and Holyoak show that an analogy is first formed, then abstracted using a more general representation of the analogues, and later used in problem solving. Gick and Holyoak performed many experiments to test the use of schema abstraction for problem solving and found that subjects who noticed and formed an abstract schema from two analogues, were able to use it later to solve a similar type of problem. The analogy was formed, abstracted and saved for later use.

teaching by using analogies - Analogy is used in education when a new concept to be learned is shown to be similar to another concept that is already understood. Once the objects in the two systems are put into correspondence for the pupil, many of the relationships and laws understood in the well known area aid in the understanding of similar relationships in the new area [Murphy et al., 1963].

Teaching by using analogies often results in a better understanding of two well defined domains also. In an example taken from a text of engineering analogies [Murphy et al., 1963, pp. 71-73], an entire analogy is presented, correspondences are formed and finally the similarities are used to further an understanding of both domains.

1.2 Analogy and Purpose

The many and varied uses of analogy shown above serve to exemplify a dichotomy which exists in the study of analogy. This dichotomy involves whether or not analogy is *purpose-directed*. Most researchers agree that analogy involves mapping some relational structure between analogous situations. Typically, there are many possible mappings between the analogues but only some of them are relevant or useful. Researchers who argue for *purpose-directed* analogy believe a relevant analogy can only be generated if the purpose for forming the analogy is one of the inputs to an analogy processing system [Kedar-Cabelli, 1988]. Other researchers feel that a purpose need not be supplied as input

for sound, plausible analogies to be formed [Gentner, 1983]. In this case, the purpose may influence what analogues are initially chosen and how the resulting analogy is used, but is not necessary for formation of an initial analogy. Forming the initial analogy is controlled by evaluations and more than one analogy may be supplied as input to a further reasoning process. An inspection of the uses above shows both views to be valid in that instances of each are found.

Support of a *purpose-directed* view of analogy is given by examples such as case-based reasoning and argumentation. In case based reasoning input of a purpose is crucial to drawing the relevant analogy. For instance, a corroborating case can only be used correctly if the purpose it will be used for is known. In the use of analogy for argument and persuasion, a system needs a goal or purpose of the argument in order to form the relevant analogy; i.e. in order to map the most relevant structure between the analogues.

Support for a *non purpose-directed* system is given by uses such as schema abstraction, teaching by analogy, scientific investigation and metaphor understanding. In each of these cases, the analogy is formed independent of any particular purpose. In Gick and Holyoak's schema abstraction, an analogy is formed, abstracted and later some transfer of problem solving information is performed. In teaching by analogy, the analogy is formed then analyzed to highlight similar relationships between the objects. In scientific investigation, typically one or more analogies are formed between domains and hypotheses are formulated. There is no purpose here beyond finding out if similarities do exist and then hypothesizing by transfer between the domains. Even in metaphor understanding there is no purpose involved outside of exploring the domains indicated by the metaphor for their similarities; leading to an understanding.

Although both views, *purpose-directed* and *non purpose-directed*, are valid and depend on the use of the analogy, this thesis takes the more general and modular *non purpose-directed* approach. An analogy is first formed under the control of evaluations. This structure can then have further analogical inferencing proceed from it, but there is no particular purpose required. This seems a reasonable choice to explore because it *is* the more general, modular approach. With a better understanding of analogy, the next section outlines the aim and objectives of the thesis.

1.3 Aim and Objectives

This section states the major aim of the thesis and breaks it down into various objectives for the thesis to meet. These objectives can be seen as outlining many of the remaining chapters.

1.3.1 Aim

To investigate the fundamental properties of computer based analogy and to use these properties as a basis for analogy formation and analogical inferencing.

1.3.2 Objectives

1) To analyze the processes of analogy as studied in human psychology.

2) To analyze mathematical theories of analogy.

3) To analyze the constraints on analogy formation to determine how plausible analogies are formed and ordered.

4) To formulate a general framework for analogy which incorporates results from the above analyses.

5) To design and implement an Analogical Tool for analogy formation based on the general framework for analogy.

6) To apply the Analogical Tool to published examples of analogy to demonstrate the abilities of the tool.

7) To investigate the extension of the Analogical Tool into the area of analogical inferencing based on results from the tool's performance in analogy formation.

8) To determine limitations of the Analogical Tool.

1.4 Meeting the Objectives - A Synopsis of the Thesis

The aim and objectives of the thesis are met by the design and testing of an Analogical Tool that is an instantiation of a general framework for analogy. This framework is based on work by Polya [1954] and embodies critical functions realized from an analysis of previous work on analogy. Using conceptual graphs [Sowa, 1984] as the knowledge representation scheme for the general framework, the Analogical Tool forms analogies and supports analogical inferencing based on them. Empirical testing of the tool against examples of analogy from the literature shows that the Analogical Tool compares well with other systems. The chapters of the thesis describe formulation of the general framework, the design, implementation, testing and critical evaluation of the Analogical Tool and a description of analogical inferencing based on the tool.

Chapter 2 analyzes previous works on analogy and arrives at three critical functions. These functions are shown to also emerge from a description of analogy given by Polya [1954]. This description characterizes analogy as the formation of "common generalizations" of two or more analogues and is presented as a general framework for analogy.

Chapter 3 discusses the knowledge representation model of conceptual graphs. Conceptual graphs were designed by John Sowa [1984] and have a mapping to and from natural language and a mapping to logic. The conceptual graph model is used for the knowledge representation scheme of the general framework. This chapter gives the reader enough background in conceptual graphs to understand the Analogical Tool presented in chapter 4.

Chapter 4 describes a computer emulation of analogy called the Analogical Tool. The tool is an instantiation of the general framework and uses conceptual graphs as the knowledge representation scheme. It is shown that the Analogical Tool derives analogies as common generalizations and "stronger" analogies are seen as minimal common generalizations. The tool is also described in terms of evaluation criteria used in analogy formation. A description of analogical inferencing based on the tool is given and lastly, the actual pseudocode of the tool is presented.

Chapter 5 presents the results of testing the Analogical Tool on various examples of analogy from the literature. The examples are from significant researchers in the field and show the strengths of the Analogical Tool. Each of the examples shows how analogy formation occurs and where possible, how analogical inferencing would proceed. Each example is also compared with the original researcher's results.

Chapter 6 shows that limitations of similarity based systems such as the Analogical Tool come mainly from issues of relevancy. Two examples which exemplify these limitations are presented and solutions are given.

Chapter 7 summarizes and discusses the major points of the thesis and presents further enhancements for the tool.

Chapter 2
Analyzing Previous Work on Analogy

This chapter analyzes previous work on analogy, by categorizing them as *psychological models, mathematical theories* and *evaluation criteria* for analogy. Comparison of works in these categories leads to an analysis of the critical functions of analogy. Following this comparison, a general framework for analogy is presented based on studies by Polya [1954]. Polya describes analogy as a "common generalization", and the *general framework* developed in this chapter provides a formal structure underlying such generalizations. The general framework is also shown to embody the critical functions of analogy and is adequate to allow for analogy formation and analogical inferencing.

2.1 A Classification for Analogy

Study of colloquial and psychological meanings of analogy has resulted in research which can be classified into three main areas:

1) Psychological Models of Analogy

2) Mathematical Theories of Analogy

3) Evaluation Criteria for Analogy

As shown in Figure 2.1, analyzing works in each of these areas leads to conclusions of critical functions of analogy. These critical functions can be found in many of the computational emulations of analogy although often they are not explicitly stated. They are also part of the foundation of the Analogical Tool which forms analogies and supports analogical inferencing based on them.

Some of the works being compared and discussed in this chapter appear in one or more of the above categories. This is done to facilitate comparison of various aspects of some of the works. While these categories do not represent the only way that works on analogy can be compared, they do serve to highlight the critical functions.

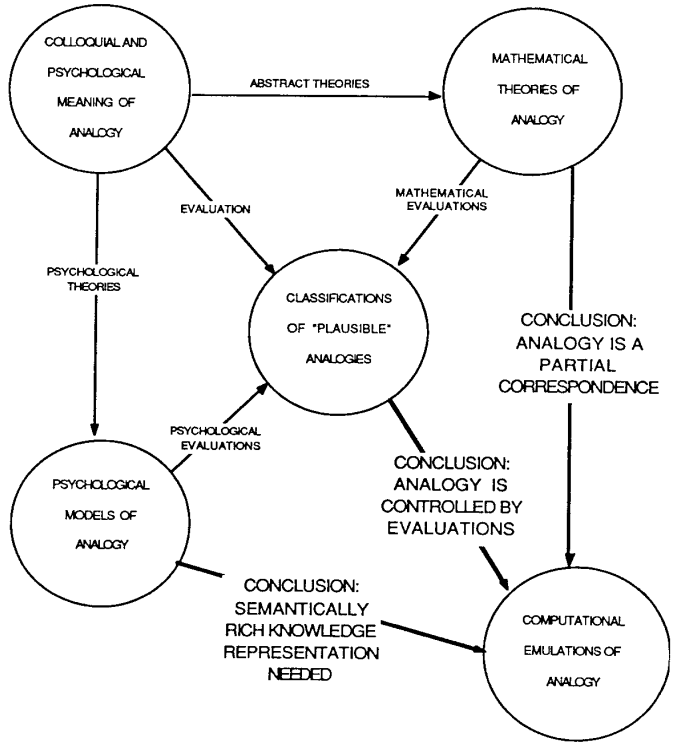


Figure 2.1 A Classification for Analogy

2.1.1 Psychological Models of Analogy

Psychological models of analogy attempt to account for how and in what circumstances people reason from analogies. These theories state that an important part of any theory of analogy is the knowledge that is used and how it is represented.

DEDRE GENTNER

The central concept of Dedre Gentner's Structure Mapping Theory [Gentner, 1983] is the assertion that a relational structure that normally applies in one domain can be applied in another domain. This idea is embodied in Gentner's *systematicity principle* which states that interconnected chains of relations are important in analogy but object attributes are not.

Gentner's theory is based on four assumptions:

1) Psychologically, domains and situations are viewed as systems of objects, object-attributes and relations between objects.

2) Knowledge is represented as propositional networks of nodes and predicates. The nodes represent concepts treated as wholes; the predicates applied to the nodes express propositions about the concepts.

3) Distinctions are made between object attributes in a predicate calculus representation, where attributes are predicates taking one argument; and relations are predicates taking two or more arguments. A further distinction is made between first order predicates which take objects as arguments and second order predicates which take propositions as arguments.

4) These representations, including the distinctions between different kinds of predicates, are intended to reflect the way people construe a situation, rather than what is logically possible.

A diagrammatic example of Gentner's knowledge representation is shown in Figure 2.2. This figure represents the source water analogue from the heat-water flow analogy described by Carnot. In this analogy, Carnot compares the difference in temperature between two bodies to the height of a waterfall. Here Carnot finds the common relational structure between the two analogues to be (Power Depends-on Difference and Amount).

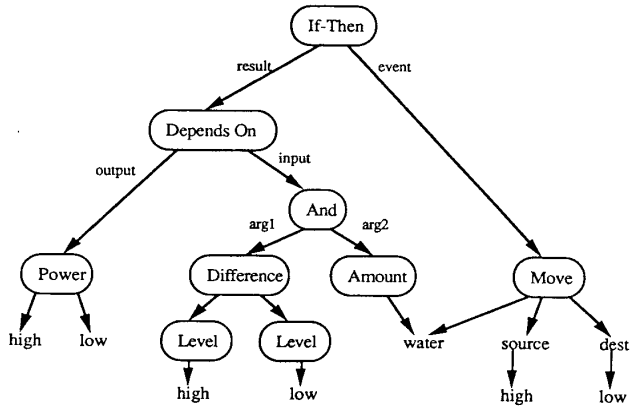


Figure 2.2 Diagrammatic Example of the source Water Analogue

From psychological studies of people, Gentner has found that there is a strong focus on relational information when analogies are interpreted. In order to represent this relational structure, Gentner chooses a higher order predicate calculus representation. This representation allows for the expression of second order predicates. Using this representation, the *systematicity principle* states that a predicate that belongs to a mappable system of mutually interconnecting relations is more likely to be imported into the target analogue than an isolated predicate. When choosing which objects to put into correspondence, choose objects which are part of a chain of interconnected relations, and in particular choose higher order relations and the relations they take as arguments.

From this discussion, it can be seen that Gentner's primary objective is a desire to represent analogies as humans do. From studies of people, Gentner notes that the important aspect of analogy is the maintaining of relational structures across domains.

BIPIN INDURKHYA

Indurkhya's 1985 PhD thesis entitled "A Computation Theory of Metaphor Comprehension and Analogical Reasoning", focuses on the ability to account for a number of characteristics of metaphors and metaphor understanding. These characteristics are a starting point from which Indurkhya defines his Schema Language knowledge representation and Approximate Semantic Transference theory (AST). AST formalizes the notion of coherence across domains by forming AT-MAPs, or approximately coherent partial mappings of domain tokens. Tokens are the vocabulary of a domain and can be constants, operations or predicates. Structural constraints define a domain and show how tokens are inter-related.

Indurkhya disagrees with Gentner's mapping of relations while ignoring attributes. Because any token can be mapped between domains in his theory, Indurkhya essentially allows mapping of both relations and attributes. This makes AST theory more general than Gentner's structure-mapping theory. Other important aspects of metaphor are embodied in the tokens found in the Schema Language. Tokens can be internal, external or derived. External tokens can be associated with another domain and are represented because Indurkhya believes people bring more to the understanding of a metaphor than just tokens generic to the concept being understood. Derived tokens show the different ways a token can be derived and allow for different viewpoints of a domain. These derived tokens also bring redundancy to the representation language; an aspect Indurkhya feels is important for metaphors.

Another important feature of metaphors is that they tend to highlight certain parts of source and target domains while down-playing others. This feature is captured by the partial nature of AT-MAPS. Only those structural constraints and tokens which are part of the AT-MAP are highlighted; all other structural constraints and tokens within them are down-played. Also, the ability to form more that one AT-MAP from two systems captures the possibility of there being more than one interpretation of a metaphor. In particular, different people often tend to understand the same metaphor slightly differently. Another important characteristic of metaphors is the resemblance between them and scientific models used for hypothesizing. In this case, Indurkhya's AST theory uses an operator called Positing Structure. This operator allows new tokens and new structural constraints to be added to the target domain in a manner constrained by the structural constraints of the

source. New tokens are created in the target domain and are then mapped to unused tokens in the source. In this way, new constraints can be hypothesized about the target domain based on what is known about the source domain. The third characteristic of metaphors which is inherent in AST is the ability to name a system of relationships in the target domain for which no literal term exists. For example, in understanding metaphors drawn between directed acyclic graphs and families, this characteristic allows for the notion of sibling to be postulated in the directed acyclic graph domain, because of the appearance of this term in the family domain. An operator called the Augmentation operator gives AST this ability. This operator uses derivations involving derived tokens of the source domain to create new derived tokens and derivations in the target domain.

In his thesis, Indurkhya starts from identified psychological aspects of metaphors and metaphor understanding before defining his theory of Approximate Semantic Transference. Because of his desire to embody certain characteristics of metaphor in the theory, Indurkhya finds it necessary to extend predicate calculus into his Schema Language of tokens with types and kinds. From this representation scheme Indurkhya goes on to define AST which entails the formation of AT-MAPS that can be incrementally extended using two operators, Positing structure or Augmentation. Each of these extensions to the theory is also driven by the desire to account for certain psychological characteristics of metaphor.

MARK BURSTEIN AND BETH ADELSON

Mark Burstein and Beth Adelson's theory of analogy deals with the use of explanatory model types [Burstein and Adelson, 1987]. Their goal is constrained purpose-directed analogical retrieval and mapping which uses a taxonomy of mental model types, and incorporates multiple analogical models. Burstein and Adelson state that no single analogical model can be found which completely describes a target system. When humans generate full and correct explanations of a system, they often need to use multiple partial models of varying types and at varying levels of abstraction. Burstein and Adelson found that models of different types are distinguished by different kinds of relations.

So far Burstein and Adelson have identified three structural models: componential, topological and geometric; and four dynamic models: functional/behavioral, causal, aggregate and synchronous. Using this taxonomy, their theory suggests that the type of model selected strongly affects the selection of base domain features and relationships for

mapping. Burstein and Adelson are testing the theory on humans to ascertain whether or not they map only a coherent subset of a base domain which is reflective of the purpose of a learning task.

Burstein and Adelson are going one step further than either Gentner or Indurkhya by classifying model types. They recognize that when people form analogies and reason from them, they map relational structures. Burstein and Adelson extend this and say that when people form analogies, they choose appropriate types of models which constrains the search for structural relations to map. When humans are searching for a behavioral explanation of a target domain for example, they will choose a source domain model which is also behavioral. This type of model contains certain types of relations, and it is only these relations which are considered in the mapping process.

GEORGE LAKOFF AND MARK JOHNSON

Lakoff and Johnson [1980a, 1980b] are interested in foundations of the human conceptual system. They suggest this system is composed of "non-metaphorical" and "metaphorical" concepts. The "non-metaphorical" concepts comprise the base structure of a conceptual system; they emerge from experiences and are defined in their own terms. There are three types of "non-metaphorical" concepts:

1) spatial orientations - up-down, in-out.

2) ontological concepts - entity, substance, etc.

3) structural experiences and activities - eating, moving, transferring objects from place to place etc.

"Metaphorical" concepts are understood and structured in terms of the "non-metaphorical" concepts. Abstract concepts like time, emotions, and the mind are defined metaphorically in terms of concepts that are more concrete like space, motion, food, etc. Typically, it takes a cluster of metaphors to fully define these abstract concepts. Each metaphor which defines a concept tends to highlight certain aspects of the concept and down-play others.

An important claim made by Lakoff and Johnson [1980a] is that "metaphorical" concepts are based on complex experiential gestalts. These are multidimensional structured

wholes arising naturally from experience. For example, to understand the metaphor ARGUMENT IS WAR, a person requires a gestalt structure for Conversation which contains among other things, who the participants are; the parts of a conversation; and the stages of a conversation, such as the beginning, central part and end. Also required is a gestalt structure for War which contains similar parts: who the participants are; the parts of a war; and the stages of a war. Understanding a conversation as being an argument involves being able to superimpose the multidimensional structure of the War concept upon the corresponding structural dimensions of conversation. As in analogy, understanding also requires putting certain terms from each structure into correspondence in order to understand for example, talking, in terms of physical fighting. Once this correspondence has been formed, we can think of winning or losing an argument, gaining or losing ground, or being on the defensive.

Lakoff and Johnson say that an important distinction to make is between an experiential gestalt and a representation of it. Experiential gestalts are structures through which humans categorize personal experiences and external occurrences. The job of cognitive scientists is to construct adequate representations of these structures. The authors state that no adequate representation has yet been been constructed that can describe the complexity of these experiential gestalts.

SUMMARY - Psychological Models

In each of the above theories, the desire to form analogies in a way similar to that of humans has led to a realization of the importance of the knowledge representation chosen. For Gentner, analogies map relational structures between target and source domains. To do this she needs a knowledge representation which can express higher order predicates. Indurkhya wants his theory of metaphor comprehension to capture many of the characteristics of metaphor. To do this he needs to define a new knowledge representation scheme and a new theory based on that representation. Burstein and Adelson's theory is based on a taxonomy of models types which help constrain the search for analogies. Their knowledge representation needs to be able to adequately represent these models. For Lakoff and Johnson, the study of the human conceptual system brings a theory of "metaphorical" concepts built from primitive "non-metaphorical" concepts. From this psychological model, Lakoff and Johnson note that the most difficult part of constructing a

computational emulation of metaphor is properly representing the knowledge contained in the experiential gestalts.

The major conclusion of this section is a semantically rich knowledge representation is fundamental to any analogy system. A knowledge representation scheme must be used which can adequately express the knowledge of each of the analogues.

2.1.2 Mathematical Theories of Analogy

These abstract theories view analogy as a mathematical or logical process. A comparison of several works in this area shows a common underlying theme of formation of plausible partial correspondences between parts of the analogues participating in an analogy. In theory, searching for all possible partial correspondences can be an intractable problem for analogues with many parts. In practice, the real problem of finding analogies has been finding ways to reduce the search for these correspondences and making this a tractable problem.

GAINES AND SHAW

In Brian Gaines and Mildred Shaw's paper Analysing Analogy [1982], the authors state that an analogy is a partial correspondence between systems represented as categories. The analogy is the entity which shows this partial correspondence. This partial correspondence must preserve some of the structure common to the categories and must be relevant, significant or meaningful in some way. In order to be relevant, an analogy must contain at least the minimal structure which is sufficient to express the essence of the analogy. Gaines and Shaw recognize that there can be many analogies formed and they assign truth values to them, making an ordering of analogies possible.

This work specifically identifies an analogy as a partial correspondence with a truth value. The remaining problem to be solved here is how this truth value is determined and then used to order the analogies.

STUART RUSSELL

Stuart Russell is concerned with the truth or soundness of propositions inferred by analogy [Russell, 1987]. He gives the following example: Bob's car and John's car share the property of being 1982 Mustang GLX V6 hatchbacks. This cannot be used correctly to

infer that Bob's car is painted red because John's car is painted red; but this knowledge does make the inference that if John's car is worth about 3500 dollars, then Bob's car is worth about 3500 dollars a probable inference. Clearly, the plausibility of the conclusion depends on information not provided in the premise. What Russell says is needed, is background knowledge that, when added to the premise of the analogy makes the conclusion follow soundly. He proposes the use of determination rules which logically contain this necessary background knowledge. For instance, if the above example included a rule which said, knowing that a car is a Mustang GLX V6 hatchback is sufficient to determine its value, we could use this to soundly infer the cost of Bob's car. It is finding the similarities between determinations and particular instantiations of them that drives the analogical inferencing power of this method.

In order to use these determination rules, the terms from the source analogue must be put into correspondence with terms in the target analogue. For example, if a determination rule states that: Make(X) determines Value(X) and the database contains the following facts:

Make(CarB) = Mustang GLX V6 and

Value(CarB) = 3500 dollars and

Make(CarJ) = Mustang GLX V6

We still need to put CarB into correspondence with CarJ before the general determination rule can be applied to determine that the Value(CarJ) = 3500 dollars. Even in the use of a logical theory of analogy using determinations, a partial correspondence must be formed between terms in the source and target. In this case all possible correspondences are not considered, only those which can be used as instantiations of the general determination rules.

MAKOTO HARAGUCHI

Makoto Haraguchi's objective is to formulate a mathematical theory of analogy and analogical reasoning [Haraguchi, 1986]. To this end, he sees four problems to be tackled: The first problem is what description language to use. Haraguchi chooses first order predicate calculus since this allows a system to perform deduction and induction as well as analogy.

The second problem is how to describe analogies in first order predicate calculus. Haraguchi states that generally, an analogy is a partial agreement or likeness between objects that are unlike in other ways. Analogous objects have portions, one from each analogue, which can be viewed as identical in some way. In this sense, the objects are partially identical and Haraguchi considers analogy to be a partial identity. The partial identity in this case is between two systems represented as parts with relations between the parts. Each system in first order predicate calculus is represented as a set of facts.

The third problem seen by Haraguchi is to find an analogy, given the above sets of facts. He calls this problem the analogy detection problem. Several partial identities can exist between analogues and Haraguchi sees a need to order them. The fourth problem in analogy is deriving conclusions or what Haraguchi calls analogical reasoning.

Haraguchi's paper deals mainly with problem three - analogy detection and ordering. Analogies are partial correspondences and the ordering criteria used by Haraguchi are dependent on trying to form maximal partial correspondences between analogues.

BIPIN INDURKYHA

Indurkhya's PhD work is one that appears in all three categories of this chapter. In this section, the work is analyzed for its mathematical properties. Indurkhya's theory of analogy and metaphor, Approximate Semantic Transference (AST) is based on three assumptions:

1) A metaphor consists of an application of one or more terms of the source domain to one or more terms of the target domain.

2) All metaphors are based on some structural analogy between the source and target domains.

3) The notion of coherency plays a central role in determining the aptness of a structural analogy.

Approximate Semantic Transference (AST) uses a knowledge representation scheme called Schema Language or SL, which is derived from first order predicate calculus. Using this language, a domain is described by a set of structural constraints, which show how tokens, the vocabulary of the domains, are inter-related.

As stated earlier, AST formalizes the notion of coherence across domains. Operationally, AST forms AT-MAPs, or approximately coherent partial mappings of tokens. An AT-MAP is partially coherent because Indurkhya bounds coherency with a specified time limit. Once an initial mapping of tokens between the source and target domains has taken place, structural constraints in the source domain which use these tokens are transformed into similar constraints in the target. This transformation consists of replacing source tokens by the mapped target tokens and transferring the entire structure to the target domain. At this point a check for coherency between this new constraint and the existing target structural constraints is done. If a new structural constraint in the target domain has not been shown incoherent within a specified time limit, coherency is assumed.

As can be seen by this description, Indurkhya's theory consists of formation of an AT-MAP or partial correspondence between tokens in the source and target domains. These tokens can be internal, external or derived.

This formation of an AT-MAP is what was referred to in §1.1.1 as the static analogy. Indurkhya feels that once this analogy is formed, a higher level system can use it to understand metaphors or to extend it for various purposes.

SUMMARY - Mathematical Theories

Each of the works described above treat analogy and analogical inferencing mathematically. Each of them characterize analogy as a plausible partial correspondence. What is being put into correspondence or why is not always agreed upon, but formation of a static analogy structure containing plausible partial correspondences between parts of the analogues is common to each of these works.

2.1.3 Evaluation Criteria for Analogy

This section presents evaluation criteria used by researchers in their computer emulations of analogy. These controlling evaluations constrain the search for analogies while ensuring that only plausible analogies are formed. Controlling evaluations also serve to order the analogies.

One of the earliest attempts at computer emulation of analogy was that by Patrick Winston [1980]. Pertaining to representation, Winston states that what needs to be

represented can be expressed in simple English sentences. Much of what is contained in these sentences is case grammar information. This case information constitutes an evaluation criteria used by Winston when partial correspondences are formed between situation parts of analogues. Winston uses three other evaluation criteria also. First, objects should be put into correspondence which share the same relations between them. Second, causal relations hold a very high importance and a higher score is given to an analogy which puts them in correspondence. Third, teachers can decide what is important in an analogy and thus constrain the search for the "stronger" analogies.

In Gentner's work on analogy the crucial evaluation criterion is the *systematicity principle*. This principle operates on a selection criterion which says that among many possible predicate matches, favor those that form coherent systems of mutually interconnecting relations. In particular this principle ignores object attributes, and favors higher order relations.

Russell Greiner's 1985 PhD thesis entitled "Learning by Understanding Analogies" [1985] has one central evaluation criterion. It uses common abstractions, or coherent clusters of facts to constrain the search for possible analogies. Only useful analogies are formed which solve a problem using abstractions that encode solutions to past problems. The computer emulation designed by Greiner, NLAG, uses an analogical hint of the form A is like B, and a target problem to map known facts about B (source analogue) onto proposed conjectures about A (target analogue). The use of common abstractions, which are defined a priori, greatly constrains which conjectures must be made in the target in order for the system to solve the problem.

Indurkhya's thesis [1985] uses the notion of maintaining coherency across AT-MAPS as a main evaluation criterion. Any transfer of structural constraints from the source analogue to the target must not destroy coherency of the target. He introduces another evaluation criterion in the form of a salience measure. This measure gives a higher salience value for tokens which are used more often. This value is used to order which structural constraints are checked first for coherency. Other evaluation criteria come in the form of heuristics. One important heuristic is a structural connectivity heuristic. It says that a token X is structurally connected to a token Y if they appear together in at least one structural constraint. This heuristic is used in forming the set of plausible extensions of an AT-MAP.

In Stuart Russell's PhD thesis [1985], the main evaluation criterion is use of determination rules. These rules are higher level domain knowledge which state the relation between one set of variables that determine the values of another set of variables. They provide a way of specifying what is sufficiently relevant to decide a question prior to knowledge of the answer to the question.

MCC (Microelectronics and Computer Technology Corporation) is conducting a project called CYC [Lenat et al., 1986] to build a large knowledge base of real world facts, heuristics and reasoning methods, particularly analogy. Analogy is used mainly to help a user choose existing frames in a frame based representation, to copy from; and for some editing. Researchers in the CYC project have identified four dimensions of analogy so far. Each dimension has its own evaluation criteria which help answer questions like when it is and isn't worth looking for analogies, when and how to extend existing analogies and when and how to use the analogies. For example, the first dimension (variations in the level and degree of matching) uses these four evaluation criteria:

1) look for identical slots and values
2) look for identical slots with different values
3) map slots with similar attributes - identical values in different slots
4) use a generalization/specialization hierarchy of concepts when both the names of the slots and the values they contain match but are not identical

CYC researchers also expect to need a taxonomy of types of analogies as well as types of analogical reasoning. This will lead the way to a kind of meta-analogical reasoning.

Hofstadter and Mitchell, in their paper "Concepts, Analogies and Creativity" [1988] see analogy as a translation process using slippages based on a slipnet network of concepts. The microworld the authors work in uses transformations of strings and asks: string1 is to string2 as string3 is to ? The network consists of nodes which are core concepts and links which are relationships between the nodes. The links have a length, and the shorter the length the stronger the resemblance between the concepts. Nodes in the net also have a semanticity measure which ensures that general and abstract concepts have a deeper influence on the analogy. The slipnet embodies what concepts are, how they overlap and how under pressures, one concept can slip into another related one. Hofstadter and

Mitchell believe that analogies are created by perceiving deep similarities between situations. They favor the following evaluation criteria measures:

1) slippages involving nodes with a high degree of semanticity

2) slippages with small distances in the slipnet

3) correspondences supported by many slippages - i.e. many aspects of the situations are being taken into account

4) correspondences that are compatible with existing correspondences - parallel slippages

SUMMARY - Evaluation Criteria

The above discussions of various researchers shows all of them use evaluation criteria to form and order plausible analogies. Many different evaluation criteria are evident, although certain ones are common to many of the researchers. For example, the criterion of mapping coherent relational structures can be found in Winston's work, Gentner's work and Greiner's work. Use of concept hierarchies is another criteria common to Winston's work, the CYC project and Hofstadter and Mitchell's work. Finally, the importance of structural connectivity can be seen in both Gentner's work and Indurkhya's work. Many of these same evaluation criteria exist in the Analogical Tool. There, they are seen as emergent properties which result from formulating a general framework for analogy.

2.2 A General Framework for Analogy

The above discussion based on classifying works on analogy into the categories of 1) Psychological Models of analogy, 2) Mathematical Theories of Analogy and 3) Evaluation Criteria for Analogy leads to conclusions which can be seen as comprising critical functions of analogy. Any computer emulation of analogy must thus contain the following three critical 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]. Polya describes analogy as a type of plausibility reasoning showing similarity on a definite conceptual level. He talks of analogy in terms of forming a "common generalization" of analogues. The following quote states Polya's description of analogy:

> "If you notice some similarity between two objects (or some system of objects) and intend to reduce this similarity to definite concepts, you think analogically. For example, you notice some similarity between two theories A and B; you notice some common points. Perhaps, you think it will be possible some day to imagine a more comprehensive theorem H that brings out all the essential common points and from which both A and B would naturally follow. If so you begin to think analogically."

He goes further and says if A is implied by H and B is implied by H then A is analogous to B, H is a common generalization of A and B, and A and B are specializations of H. This relation is shown in the following diagram [Polya, 1954].

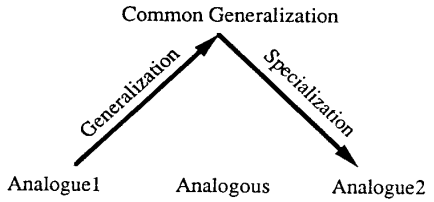


Figure 2.3 Analogy as A Common Generalization

Polya also talks about the importance of the relational structures of analogous systems. He says that analogy differs from other forms of similarity because there is a need to clarify an analogy by getting down to clear concepts. One form of clarifying an analogy is to find where they agree in their relational structures.

From this description of analogy by Polya we can describe a general framework for analogy which contains within it the three critical functions that are needed for a

computational emulation of analogy. 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 as Polya calls it, are also those which have been put into correspondence. Again this correspondence is partial because A and B are not identical. 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. Polya hints at this when he says that the common generalization H above will bring out all of the essential common parts; that is, of the many possible common generalizations which can be formed, one or few of these are more worthwhile than the others. 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.

Figure 2.4 gives a diagrammatic view of this general framework for analogy. Using this figure we can clarify the general framework. This clarification is done on a very colloquial level here. In §2.3 a more mathematical discussion of the general framework is given. At this point the aim is to present a description of analogy which is general enough yet minimal enough to just describe analogy.

Figure 2.4 shows two analogues and their common generalizations. Each analogue consists of parts, and each of these parts is "part-of" the analogue. These parts are what Polya refers to when he talks of specifying common points between analogues. 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. Generalization here is defined as any relation that is inherited by every part of the generalization. The form of generalization being described also corresponds to the notion of generalization as a transformation that loses information and adds none. There can be no more information contained in the common generalization than there was in either of the analogues. 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 2.4 shows there can be many common generalizations formed but only some are minimal. These generalizations are minimal within our ordering

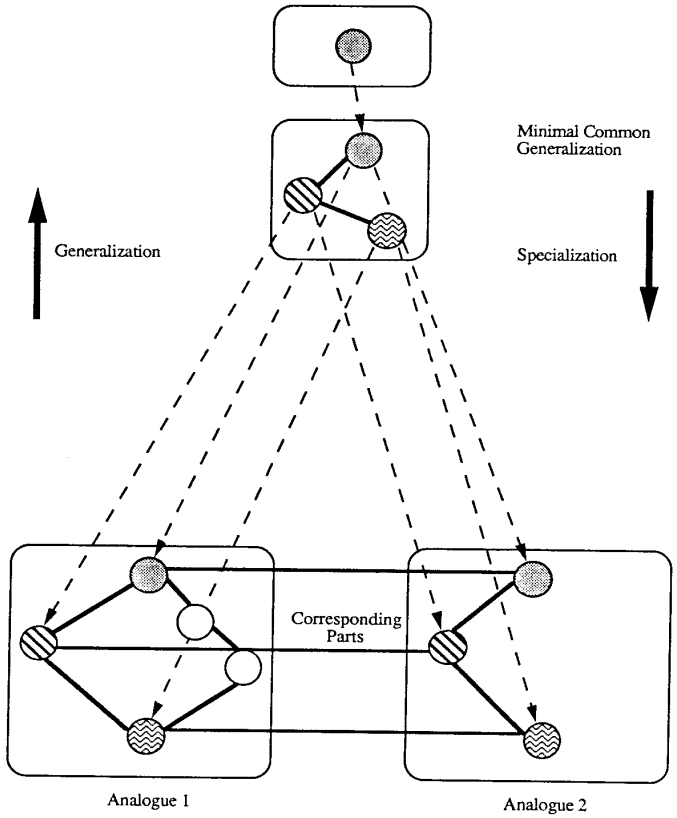


Figure 2.4 A General Framework for Analogy

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 and they can be seen as more plausible because they express the maximum amount of similarity between the analogues.

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. With this structure now in place in the target analogue, the minimal common generalization is extended and new parts are generalized. This new generalization is more specific than the original minimal common generalization. This process is depicted in Figure 2.5 by use of the new dashed part in the target analogue and in the new more specific minimal common generalization. Any new parts hypothesized as existing in the target analogue in this way must be checked for consistency and validity. These new hypotheses are plausible through analogical inferencing but not necessarily true.

Most researchers described in the previous sections use the idea of common generalizations for analogy either explicitly or implicitly. For example, Greiner [1985] uses common abstractions which are common generalizations, as a general problem solving principle. Using pre-defined abstractions, the source and target analogues are seen as specializations of the common abstraction. Gick and Holyoak [1983] use schema abstractions as general problem solving principles. These schema abstractions are also common generalizations that are re-usable by specializing them for a particular problem. Gentner's use of the systematicity principle can be seen as defining a priori what the common generalization will look like. For Gentner, the formation of the common generalization is driven by the common relational structure found or hypothesized in the analogues.

This thesis shows that the general framework for analogy described by Polya and clarified here is adequate to allow for powerful analogical reasoning when conceptual graphs are used for the required knowledge representation scheme. Chapter 3 describes conceptual graphs in enough detail for all aspects of the Analogical Tool to be understood.

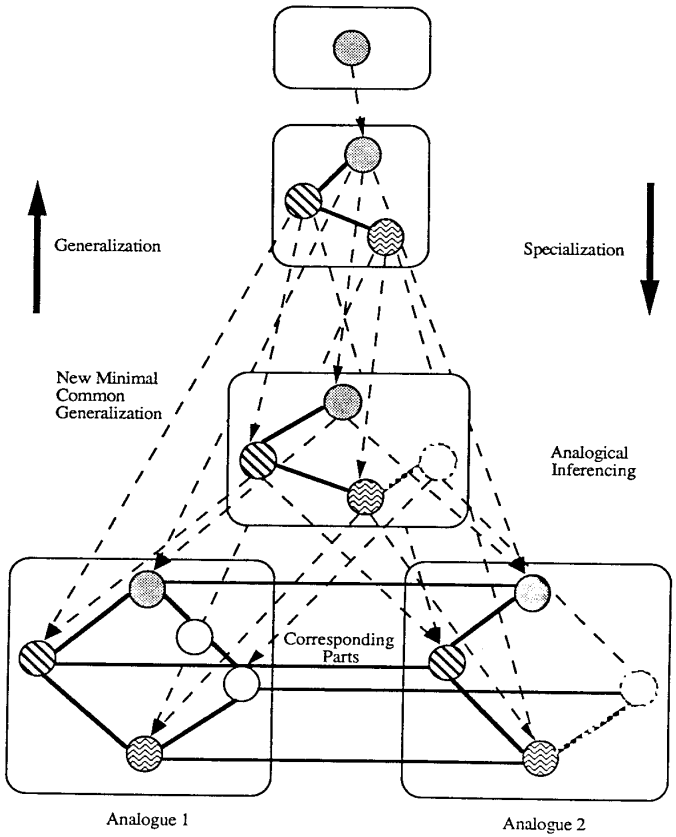


Figure 2.5 Analogical Inferencing in the General Framework

Chapter 4 shows how the notions of "part-of", "generalization" and the "generalization ordering" relation of the general framework have a direct correspondence in the conceptual graph model and allow for implementation of the Analogical Tool.

2.3 Technical Addendum to Chapter 2

This section presents a mathematical discussion of the general framework for analogy. It is not part of the main development of the thesis but presents an initial step towards a theoretical representation of the general framework.

The basic primitive used is a class of knowledge structures (KSs) which are undefined but can be thought of as statements about a world. To be able to derive analogy relations between KSs, that is structure-preserving partial correspondences, we need a few simple operations. To establish a correspondence, we need the capability to test that two KSs are the same in some sense. To allow the correspondence to be partial at a structural level, we need the notion of one KS being 'part-of' another in some sense, and the capability of selecting the parts. To allow the correspondence to be partial at an informational level, we need the notion of one KS being a 'generalization' of another, that is of losing existing information and not adding any new information. We use a set-theoretic vocabulary for the primitives needed.

Thus consider:

1. A set K, of *knowledge structures*

2. A reflexive, symmetric, transitive relation of *equality* on K, =, such that it is possible to determine for all k,e∈ K if e=k, that is whether e is equal to k.

3. A relation of *part-of* on K, π, such that it is possible to determine by some procedure for all k,p∈ K if pπk, that is whether p is a part-of k, and which is consistent with the relation of equality:

$$\forall k,k' \in K,\ k=k' \Rightarrow \forall p \in K:\ p\pi k,\ p'\pi k',\ \exists p' \in K:\ p=p' \qquad (1)$$

that is, equal structures have equal parts.

4. The *part-of* relation also defines a natural partial order on K, $\supset$, such that:

$$\forall k,j \in K,\ k \supset j \Leftrightarrow \forall p \in K:\ (p\pi k \Rightarrow p\pi j) \vee (\exists i \in K:\ k \supset i \wedge i \supset j) \quad (2)$$

The second term is put in to obtain a transitive closure since π may not itself be transitive.

5. A relation of *generalization* on K, γ,such that is possible to determine for all k,g∈ K if gγk, that is whether g is a generalization of k, by the following test:

$$\forall k,g \in K,\ g\gamma k \Leftrightarrow \forall g' \in K:\ g'\pi g,\ \exists k' \in K:\ k'\pi k \wedge g'\gamma k' \qquad (3)$$

That is, g is a generalization of k if every part of g is a generalization of some part of k, that is, if Figure A1 holds.

6. The *generalization* relation also defines a natural partial order on K, $\geq$, such that:

$$\forall k,g \in K,\ g \geq k \Leftrightarrow g\gamma k \vee (\exists b \in K:\ g \geq b \wedge b \geq k) \qquad (4)$$

That is the relation, $\geq$, is the transitive closure of the relation, γ.

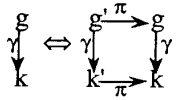


Figure 2.6 Diagram Defining Generalization

Note that generalization is defined as *any* relation that is inherited by every part of the generalization. This corresponds to a wide range of colloquial meanings of generalization as a transformation that loses information and adds none. We have subsumed all aspects of structure-preservation in the *part-of* relation applied to knowledge structures. Note that this subsumes many structures within the world that the KS represents. For example, if a logical calculus were being used for knowledge representation, its entailment relation, $\rightarrow$, would be a natural choice for the definition of π, that is pπk $\Leftrightarrow$ k$\rightarrow$p.

We have not required the basic relations *part-of* and *generalization* to be transitive. It is often the case that components are treated as indivisible at one level, and divisible at a lower level. We may not want to think of a very high level generalization as being in itself a

generalization of something very concrete, but rather think of the chain of generalizations joining them. In both cases the order relations obtained by transitive closure of the *part-of* and *generalization* relations correspond to intuitions that something about these relations should be transitive.

Formation of Analogies

The formation of an analogy between two knowledge structures may be expressed in terms of the primitives defined. An analogy is a structure-preserving, partial correspondence between two dissimilar KSs, the analogues. Generalization is used to reduce the dissimilarity until a common generalization for both analogues is found. In analogy formation, it is usually required that the generalization is minimal, this being considered the "stronger" analogy. Minimal generalizations are not necessarily unique. Thus a KS, a, is an analogy between two KSs, a' and a", if it is a mutual generalization and there is no generalization strictly lesser than it satisfying this condition, that is:

$$a\gamma a' \wedge a\gamma a'' \wedge \sim(\exists b \in K: a\gamma b \wedge \sim(b\gamma a) \wedge b\gamma a' \wedge b\gamma a'') \qquad (5)$$

This establishes a partial correspondence between a' and a" because, from (3), to each part-of the generalization, a, there corresponds a part of the analog, a', and a part of the analog a", so that we can regard these parts in correspondence through a relation, «a», such that:

$$\forall p', p'' \in K: p'\pi a' \wedge p''\pi a'', p'\langle\langle a\rangle\rangle p'' \Leftrightarrow \exists p \in K: p\pi a \wedge p\gamma p' \wedge p\gamma p'' \qquad (6)$$

Figure A2 shows these results in diagrammatic form.

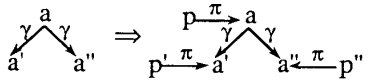


Figure 2.7 Analogy as Mutual Generalization and the Implied Partial Correspondence

Thus the formation of analogies can be expressed in terms of the weakly constrained constructs which we have defined for a very general notion of a knowledge structure. The

following section shows that this formulation is adequate to allow for analogical inferencing using the same constructs.

Analogical Inferencing

Analogical inferencing can be defined in terms of the existing primitives by noting that it looks for an analogy which is a lesser generalization than that defined in (6) by allowing the target analogue about which an inference is to be made, a" say, to be extended in such a way as to make this possible. Figure A3 is a variation on Figure A2 showing an extended analogy, a, involving an extended analogue, a",. The part-of relations on the right are similar to those in Figure A2 but now involve the possibility that p" is *not* part of a" but rather a hypothesized extension of it.

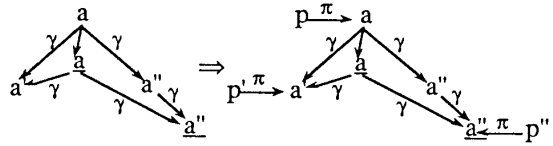


Figure 2.8 Extensibility of an Analogy

Chapter 3
Conceptual Graphs

The Conceptual Graph model proposed by John Sowa [1984] is a method of representing mental models which people use to understand the world. The importance of the conceptual graph model is the way in which it combines a mapping to and from natural language and a mapping to logic. Conceptual graphs themselves contain concepts, relations and arcs between them. Concepts represent any entity, attribute, action, state or event that can be described in natural language, the relations show the roles that each concept plays and the arcs serve as connectors between the two. A conceptual graph asserts a proposition and takes the form of a finite connected bipartite graph. These graphs can be written in either a graphical representation or in a linear form to conserve space.

The first section of this chapter presents simple graphs which form the core of the conceptual graph model and have a direct mapping to first order predicate calculus. The second section presents compound graphs which allow for expression of more complex sentences, and still have a mapping to logic. The third section discusses specialization and generalization as it pertains to the graphs. The final section outlines why the choice of conceptual graphs as the knowledge representation for analogy is a good choice and describes exactly what parts of the conceptual graph model are used by the Analogical Tool.

3.1 Simple Graphs

This section presents parts of the conceptual graph model that form a central core. This includes concepts, relations and the arcs between them. Also central to the conceptual graph model is the notion of a type lattice of concepts. Concepts appearing in the type lattice can be defined by necessary and sufficient conditions as described by Aristotle [Sowa, 1984] and by schematic or prototypical definitions as discussed by Wittgenstein [Sowa, 1984]. Also central to the model is the ability to map the graphs into first order predicate calculus.

An example of a simple graph is:

[Cat: #123] -> (State) -> [Sit] -> (Location) -> [Mat]

Which represents "A cat named 123 is sitting on a mat."

3.1.1 Concepts and Relations

Concepts represent the entities, attributes, actions, states or events found in natural language. In conceptual graph notation, they are shown as square boxes. A concept box contains two fields separated by a colon and shows a concept type on the left of the colon and a referent field on the right of the colon. In this way both generic concepts and particular individuals can be referred to. For example, [Person: *] or [Person] both refer to the generic concept of Person, while [Person: #123] or [Person: Sam] refer to particular individuals, one named Sam and one named 123. Every generic concept in the graph terminology is existentially quantified. Generic concepts act like variables in logic, while individuals are like constants in logic.

Relations in the conceptual graph model specify the role a concept plays and define the relationship between concepts. Relations are shown as circles in the graph notation and can have any number of arcs. For example (Past) is a monadic relation with one arc, (Agent) is a dyadic relation with two arcs and (Between) is a triadic relation requiring three arcs. The conceptual graph model borrows from Fillmore's case relations [Fillmore,1968] for many of its relations such as Agent, Patient, Instrument, etc. Other relations also exist beyond just case relations. These relations can exist in a type lattice and new relations can be defined in terms of existing relations.

3.1.2 The Type Lattice

Each concept in a conceptual graph represents something in the world and is specified by a type label in the conceptual graph model. These type labels exist in a hierarchy of types that supports inheritance of properties from supertypes to subtypes. This hierarchy is partially ordered by $\leq$ and types higher in the hierarchy are more general than lower ones. Concepts in a type hierarchy are natural or abstract types such as Person or Health. When role types are added such as Pet or Spouse, the type hierarchy becomes a lattice with the

universal type [T] as the most general type in the lattice, and the absurd type [⊥] as the most specific type in the lattice. With a type lattice defined we can describe the terms minimal common supertype and maximal common subtype.

A minimal common supertype of two concepts in the type lattice is the lowest or most specific common supertype of the concepts. Similarly, a maximal common subtype of two concepts in the type lattice is the most general of all possible common subtypes for the concepts. For example, Feline and Wild-Animal have common subtypes Jaguar, Lion and Tiger as shown in Figure 3.1, but none of them is a maximal common subtype. If the type Wild-Feline is added to the type lattice as shown in Figure 3.2, it becomes the maximal common subtype of Feline and Wild-Animal and a minimal common supertype of Lion, Tiger and Jaguar.

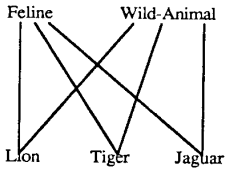


Figure 3.1 Before Adding Wild-Feline

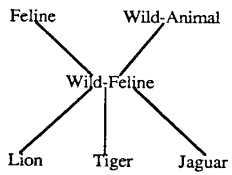


Figure 3.2 After Adding Wild-Feline

New type labels can be added to an existing type lattice by a variation of monadic lambda abstraction represented in conceptual graphs. This abstraction allows for definition of a new type in terms of an existing type. For example, a new type, Circus-Elephant can be defined in terms of an existing concept Elephant:

type Circus-Elephant(x) is

[Elephant: *x] <- (Agent) <- [Perform] -> (Location) -> [Circus].

This says that a Circus-Elephant is an Elephant that performs in a circus, where x is a formal parameter that defines the subtype, supertype relation; that is Elephant is a supertype of Circus-Elephant. The assumption here is that a type definition declares that a type label *t* is defined by a monadic abstraction $\lambda a\ u$ and is written, **type** t(a) **is** u. The body *u* is called the differentia of *t*, and type(a) is called the genus of *t*. The abstraction $\lambda a\ u$ may be

written in the type field of any concept where the type label *t* may be written. These type definitions can be used to define new concepts types as well as new conceptual relations. Use of lambda abstraction for type definitions in conceptual graphs also allows for type contraction and expansion, because a single type label can be used instead of the entire graph that defines it.

3.1.3 Background Knowledge

Conceptual graphs emphasize semantics, and each conceptual graph is meant to be embedded in a larger semantic net which gives meaning to the graph. There are several ways the larger net adds meaning. First, meaning is added by associating the grammar rules of language with each concept type label. Second, lattices of concept and relation types add meaning by defining the relationship between the concepts and relations at different levels of generality. Third, formation rules add meaning by defining constraints on how each concept may be linked to conceptual relations. Fourth, each conceptual graph may also be linked to some context or episode to which it is relevant. A fully defined knowledge representation system using conceptual graphs has all this information associated with it. The Analogical Tool uses some parts of this conceptual graph model with the assumption that full specification of the model is the ultimate goal.

Other background knowledge can be expressed in conceptual graphs using the kinds of schemata and prototype structures described by Wittgenstein [Sowa, 1984]. Conceptual graphs support definitions of schemata which show typical ways concepts can be used. These schemata incorporate domain specific knowledge and are defined using monadic lambda abstractions similar to type definitions. The difference is that a concept type may only have one type definition, but arbitrarily many schemata associated with it. The different senses of the concept Open, for instance, have different schemata for each sense. As an example, an agent may Open a door with a key, but a very different sense of the concept is shown by an agent Opening a meeting. Each of these senses of the word is described using a lambda abstraction and together form the schematic cluster for the concept Open.

Background knowledge can also be expressed as prototypical knowledge which defines a typical instance of a concept. Prototypes specify defaults that are true of a typical case but

not necessarily of any particular case. These prototypes are also defined using monadic lambda abstractions.

3.1.4 A Logical Mapping

The conceptual graph model defines the operator ϕ which maps simple conceptual graphs into formulas in the first order predicate calculus. For these simple graphs, the only logical operators which are needed are conjunction and the existential quantifier. For example, the conceptual graph:

[Cat: #123] -> (State) -> [Sit] -> (Location) -> [Mat]

maps into the following formula when the ϕ operator is applied:

$$\exists x \exists y \, (Cat(\#123) \wedge State(\#123,x) \wedge Sit(x) \wedge Location(x,y) \wedge Mat(y)).$$

Conceptual graphs are usually more concise than logical formulas because arcs on the graphs show the connections more directly than variable symbols.

3.2 Compound Graphs

Compound conceptual graphs allow for the expression of more complex sentences than can be described using simple graphs. The components which comprise compound graphs are discussed in this section and include nested propositions, co-reference links, modality, tense and negation. Figure 3.3 shows an example of a graph which contains most of these elements.

3.2.1 Propositions

A proposition is a concept whose referent is a set of conceptual graphs that are being asserted. The graphs being asserted are said to occur in the context of that proposition, thus propositions are also referred to as context boxes. Propositions can be nested inside of one another and Proposition is the default label for a box that has no other type label. Conjunction of two or more graphs is represented by drawing all the graphs inside a proposition. Figure 3.3 contains three nested propositions. The first one corresponds to the entire sentence shown at the top of the figure. Another proposition is the object of the

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

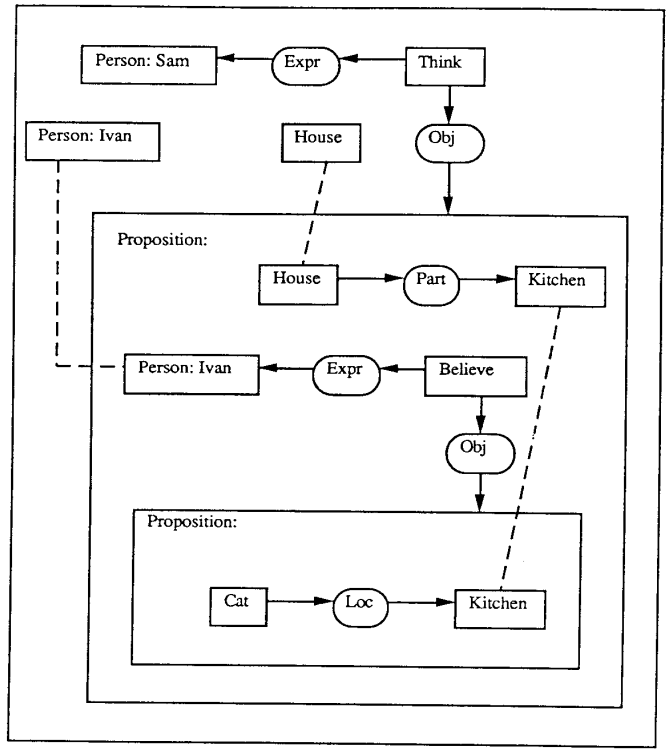


Figure 3.3 A Conceptual Graph of a Complex Sentence

Think concept. The third proposition is the object of the Believe concept, i.e. Ivan is expressing a Belief that [there is a cat in the kitchen].

3.2.2 Co-reference Links

Co-reference links in conceptual graphs show which concepts refer to the same entities within a graph. In a sentence, these links are expressed as pronouns or other anaphoric references. Figure 3.3 shows co-reference links using dashed undirected lines. These co-reference links are also referred to by Sowa [1984] as lines of identity and denote an equality relation between concepts. No concept can belong to more than one line of identity, but there may be many concepts chained together if a paragraph or other large piece of text is being represented.

Co-reference links often join pronouns to their referents or refer to a concept previously mentioned. In Figure 3.3 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.

3.2.3 Negation, Modality and Tense

Negation in conceptual graphs is denoted by the monadic relation (Neg). Negative propositions or contexts are expressed by attaching (Neg) to a particular proposition, which in turn denies the conjunction of the graphs in the proposition's referent field as in standard logic.

Tense and modality pervade natural language and thus need to be expressed. Conceptual graphs describe how tense and modality relate to the real world; they state for instance, whether something has happened, can happen, will happen or should happen.

Graphs with no mention of tense are assumed to be present tense. A monadic Past tense relation can be defined by the following:

relation Past(x) is

[Situation: *x] -> (PTime) -> [Time] -> (Succ) -> [Time: Now].

This states that the Situation occurs at a point in time, which has a successor, which is a time named Now.

Many modes exist in language and logicians have been successful in enumerating many of them. They have identified alethic modalities such as possibility and necessity, deontic modes for permission and obligation, epistemic modes for knowing and believing and others. Conceptual graphs support the representation of all forms of modal logic which can be constructed from standard concepts and relations. Figure 3.3 for example, shows the usage of the modes Think and Believe.

An example of a sentence which requires the use of tense, modality and negation is "Julian couldn't fly to Mars". This sentence can be represented by the following graph:

(Past) -> [(Neg) -> [(Psbl) -> [

[Person: Julian] <- (Agent) <- [Fly] -> (Dest) -> [Planet: Mars]]]].

Where a logic for these elements of compound graphs exist, the ϕ operator which was shown to map simple graphs into first order predicate calculus also maps compound graphs into logic.

3.2.4 A Logical Mapping For Compound Graphs

The logic of conceptual graphs is based on Charles Sanders Peirce's existential graphs. Since all boolean operators can be reduced to negation and conjunction, the standard Boolean combinations can be represented using nested negative contexts. Figure 3.4 shows the graph notations for various boolean combinations; where $\neg$ is an abbreviation for (Neg). Conceptual graphs can also represent universal quantification. Consider the graph for "An Ostrich does not fly":

(Neg) -> [[Ostrich] <- (Agent) <- [Fly]].

When the ϕ operator is applied we get the formula:

$$\neg\exists x\exists y\ (\text{Ostrich}(x) \wedge \text{Fly}(y) \wedge \text{Agent}(y,x)).$$

Since moving a quantifier into or out of a negative context converts $\forall$ to $\exists$ and $\exists$ to $\forall$, this

formula is also equivalent to:

$$\forall x \forall y\ (\neg \text{Ostrich}(x) \vee \neg \text{Fly}(y) \vee \neg \text{Agent}(y,x)).$$

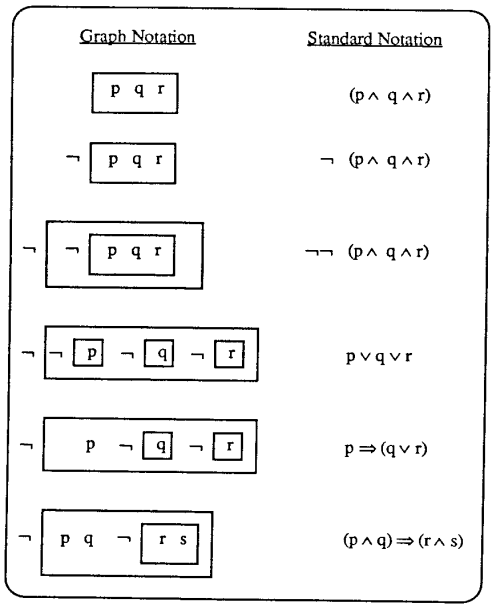


Figure 3.4 Various Boolean Combinations

So far, the ϕ operator can map positive and negative propositions into formulas in first order logic. The ϕ operator must also be able to map graphs containing co-reference links into logical formulas. Because co-reference links express an equality relation, the ϕ operator supports a mapping from conceptual graphs to first order predicate calculus with equality. For example consider the graph for Rosalie is not Rosann, which has a co-referent link from Rosalie to Rosann:

[Person: Rosalie] (Neg) -> [[Person: Rosann]].

When ϕ is applied this becomes:

$$\text{Person}(\text{Rosalie}) \wedge \neg (\text{Person}(\text{Rosann}) \wedge \text{Rosann} = \text{Rosalie}).$$

The scope of quantifiers in the graphs is determined by the level of nesting of concepts. Concepts appearing in outer levels dominate concepts at inner levels. For co-referent concepts their level is the outer most level they appear in.

Past and present tenses of conceptual graphs can be easily mapped by the ϕ operator into first order logic, but mapping modes in conceptual graphs to logical formulas requires the use of modal logic. Modal logic is the extension of standard logic to include necessity denoted by $\Box$, and possibility denoted by $\Diamond$. The formula $\Box$ p states that p is necessarily true and $\Diamond$ p states that p is possible. Thus translation of:

(Psbl) -> [[Person:Julian] <- (Agent) <- [Fly] -> (Dest) -> [Planet: Mars]]

by the ϕ operator results in:

$$\Diamond \exists x\ (\text{Person}(\text{Julian}) \wedge \text{Fly}(x) \wedge \text{Planet}(\text{Mars}) \wedge \text{Agent}(x,\text{Julian}) \wedge \text{Dest}(x,\text{Mars})).$$

Conceptual graphs can also support the expression of higher order logic where quantifiers can range over entire conceptual graphs.

3.3 Specialization and Generalization

The conceptual graph model defines specialization operations used for selectional constraints and generalization operations for logic and model theory.

3.3.1 Specialization

Every concept and relation used in conceptual graphs has a canonical graph associated with it that specifies the constraints on the pattern of concepts and relations that may be linked to it. For example the canonical graph for the concept Go is:

[Mobile-Entity] <- (Agent) <- [Go] -> (Dest) -> [Place].

This graph states that Go has a Mobile-Entity as its Agent and some Place as its Destination. The selectional constraints given by the canonical graphs rule out the formation of graphs which correspond to non-sensical sentences. The conceptual graph model defines formation rules as a kind of generative grammar for conceptual structures. New canonical graphs can be derived from other canonical graphs by use of four formation rules; copy, restrict, join and simplify. The formation rules join and restrict, specialize conceptual graphs and maintain selectional constraints but are not truth preserving.

The copy rule makes an exact copy of a canonical graph. The restrict rule replaces the type label of a concept with the label of a subtype as in [Girl] from [Person]. It may also convert a generic concept to an individual as in [Person] to [Person:Sam].

The join rule merges identical concepts. Two graphs can be joined by overlaying one graph on top of the other so two identical concepts merge into one. The result is that the relations that were linked to the identical concepts are now linked to the merged concept. Two different concepts may also be joined if a restriction can be performed on either or both concepts to make them identical. After such a join, identical relations may be linked to the merged concept and one other concept. These relations are redundant and the simplify rule erases one of them. Figures 3.5 to 3.8 show two canonical graphs which are first restricted then joined then simplified [Sowa,1984]. These formation rules define a kind of syntactic graph grammar for canonical graphs and also enforce certain semantic constraints on the graphs.

An extension of the simple join rule is the maximal join operation. A maximal join is a sequence of joins and simplifications applied to the matching nodes of two canonical graphs. The join is maximal because all joins that can be performed between two graphs are performed. This maximal join operation is not unique because two graphs may be

joined together in more than one way. The maximal join operation is important for conceptual graphs because it provides for many of the inferencing mechanisms used on the graphs.

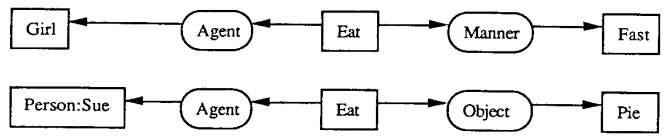


Figure 3.5 Two Canonical Graphs

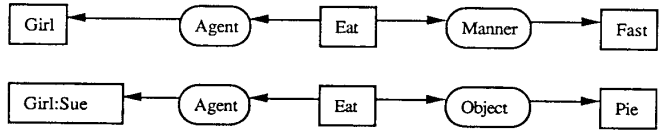


Figure 3.6 Restriction of the Second Graph in Figure 3.5

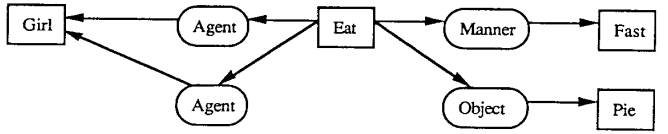


Figure 3.7 Join of the Two Graphs in Figure 3.6

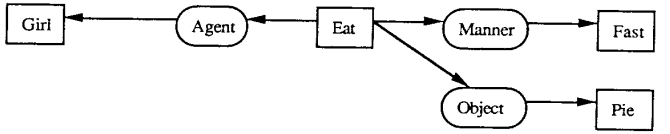


Figure 3.8 Simplification of Figure 3.7

3.3.2 Generalization

Generalization of conceptual graphs is truth preserving and if a conceptual graph u is canonically derived from a conceptual graph v (possibly with the join of other conceptual graphs), then u is called a specialization of v, written u <= v, and 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.

This generalization also means that the graph u with possibly more properties implies the simpler graph v.

If the graph u mentioned above is a specialization of the graph v, then there is a subgraph u' in u that represents the original graph v with possible restrictions. This subgraph u' is called the projection of v in u and a projection operator π maps graphs at higher levels of the generalization hierarchy into ones at lower levels. The more general graph v may have several 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'.

Common Generalizations

If two conceptual graphs are compared, we can describe a common generalization of the two graphs and possibly a common specialization. Any two graphs always have at least one common generalization because the graph [T] (the universal graph) is a generalization of all graphs. Figure 3.9 shows two graphs u1 and u2, their common generalization v and

their common specialization w. The graph v is a common generalization because it has projections into both u1 and u2. A common specialization of graphs exists only if two graphs can be joined on one or more nodes.

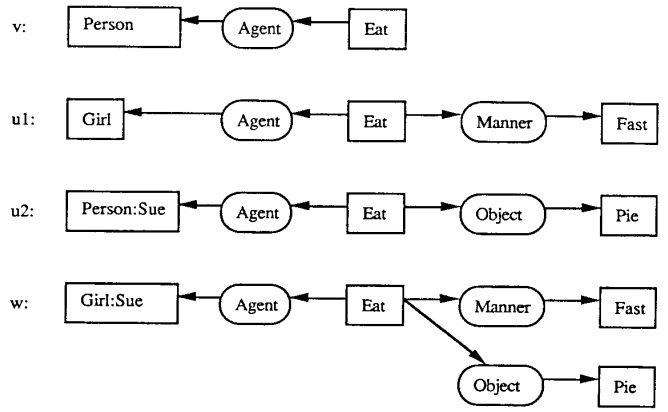


Figure 3.9 A Common Generalization v and a Common Specialization w

For any two graphs, there can be many common generalizations. Because these generalizations exist in a generalization hierarchy, some of them are minimal common generalizations. These minimal common generalizations are the most specific of the generalizations, first in concept restriction and second in extent (i.e. containing 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. Because the generalization hierarchy is not a lattice, two graphs may not have a unique minimal common generalization.

3.4 Conceptual Graphs for Analogy

There are several reasons why conceptual graphs are a good choice for the knowledge representation scheme of the general framework:

1) They have a mapping to logic as discussed above 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 expressive enough to be able to represent examples of analogy taken from the literature.

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

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

As was briefly mentioned above, the Analogical Tool described in the next chapter does not use all aspects of the conceptual graph model. Those parts which are used are enumerated here: Only generic concepts are used to describe analogues, individuals are not used. This means specifically that the referent field of concepts is not represented. Case as well as other relations are used but there is no hierarchy of relations. A type hierarchy is used only to show the generalization-specialization relations between concepts. There are no actual definitions, schemata or prototypes attached to the concepts, although this would be a natural extension to the system. All aspects of compound graphs are used by the Analogical Tool. The specialization canonical formation rules are not used as such, although the notion of specialization plays an important part in the process of analogical inferencing. Generalization is a very important part of the Analogical Tool along with the generalization hierarchy and the projection operator.

This chapter has described conceptual graphs in enough detail for a description of the Analogical Tool to be presented. The next chapter describes the Analogical Tool as an

instantiation of the general framework and describes the implementation of the Analogical Tool in the object oriented language of Smalltalk [Goldberg and Robson, 1983].

Chapter 4
The Analogical Tool

The general framework for analogy formation and analogical inferencing developed in §2.2 can be instantiated as a computational tool by using conceptual graphs as the knowledge representation scheme. This is facilitated by elements in the conceptual graph model which correspond directly to the notions of "part-of", "generalization" and the "generalization ordering" relation of the general framework. Firstly, the conceptual graph model defines *parts of* the analogues as the concepts, relations and arcs between them. Secondly, the "common generalization" of the general framework has a direct correspondence to the *common generalization* of conceptual graphs. Generalization of conceptual graphs occurs by climbing the type hierarchy or by taking subgraphs of an original graph. A *common generalization* in conceptual graphs is a graph which is a generalization of two or more other graphs. Thirdly, these generalizations are *partially ordered* in a generalization hierarchy, and this ordering corresponds to the requirement for a "generalization ordering" for the general framework.

This chapter describes the Analogical Tool in terms of how each of these direct correspondences is used to develop a computational tool which forms analogies and supports analogical inferencing based on them. Four aspects of the tool are described in this chapter and the ordering of them is significant. Each section builds on the last and describes the tool at a more specific level. First, it is shown that the tool forms minimal common generalizations as preferred "stronger" analogies. Second, analogy formation is shown to be under the control of four formation evaluations and two ordering evaluations. The formation evaluations operate to determine how and what parts of the analogues are generalized, and the two ordering evaluations partially order the generalizations. Third, it is shown that analogical inferencing based on the tool can proceed by extension of the target analogue and formation of a new minimal common generalization. Fourth, the Analogical Tool is described in its most specific form - as an object oriented implementation in the programming language of Smalltalk [Goldberg, Robson, 1983].

4.1 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 4.1 gives a simple example of three ordered common generalizations which can be formed from the two analogues shown.

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.

Many common generalizations are formed by the Analogical Tool but only a subset are the minimal common generalizations the tool searches for. The preferred minimal common generalizations preserve as much structure as possible of the analogues being generalized. An important aspect of the conceptual graph representation is the connectivity shown between concepts. Maintaining this connectivity in the generalization is an important structural criterion for more plausible analogies.

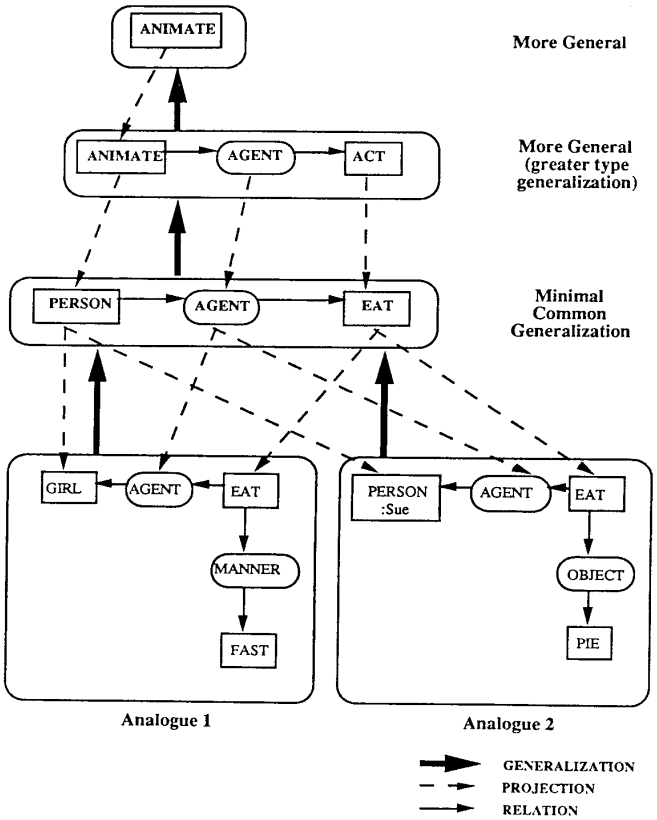


Figure 4.1 Forming Common Generalizations

Formation of the minimal common generalizations also preserves the truth value of each of the analogues. Because no information is added to the generalization that was not in the analogues, whatever truth value the analogues have is maintained in the generalization. This says nothing about the truth value of the analogues themselves, nor how they correspond to the real world; it says only that whatever that truth value, it is maintained in the generalization. This can be seen as a necessary logical criterion for more plausible analogies. Finally, use of the type hierarchy in finding the minimal common generalizations minimizes the semantic distance between concepts put into correspondence. We wish to generalize any two concepts just enough to include both of them.

4.1.1 Correspondence and Projections

The formation of a minimal common generalization of two analogues involves putting the parts of 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 the conceptual graph model as arising from a pair of projections between the parts.

As shown in Figure 4.1, formation of a common generalization results in projections from the generalization into each of the analogues (these are shown by the dashed lines). As described in §3.3.2, the projection operator maps graphs at higher levels of the generalization hierarchy into ones at lower levels. Each pair of projections from the generalization into the analogues also defines the parts that are put into correspondence. It is this combination of the generalization hierarchy and the projection operator which the Analogical Tool uses to derive the set of minimal common generalizations and to define the parts put into correspondence.

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. These formation and ordering evaluations also correspond to elements of the general framework.

4.2 Analogy Formation

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.

4.2.1 Formation Evaluations

There are four formation evaluations which work together to derive the common generalizations. Three of the formation evaluations 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.

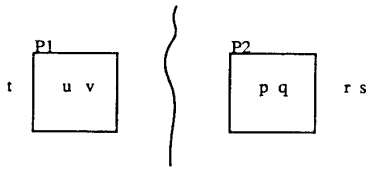
Same Relation Evaluation The first evaluation requires same relations between two pairs of potentially corresponding concepts. As shown in Figure 4.1, if [Girl] and [Person:Sue] are put into correspondence, then it is possible to put [Eat] and [Eat] into correspondence because they share the same (Agent) 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 As described in §3.2.2, 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 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.

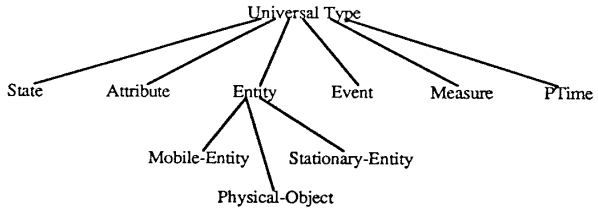
Proposition Evaluation Proposition concepts which contain conceptual graphs in their referent fields define a *contains* relation between the proposition concept and the graphs within it. This relation defines yet another subgraph structural evaluation. This subgraph evaluation requires that if proposition 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. For example, if P1 and P2 in the following diagram are put into



correspondence then subgraph u can be put in correspondence with either p or q but not with r or s since they are outside of the proposition. Accordingly, if P1 and P2 are in correspondence then r could be put in correspondence with t but not with u or v. This is because the proposition along with its referent graphs defines a particular kind of part, subgraph or unit which must be maintained.

If the propositions are not in correspondence with each other (consider for example that P2 does not exist but the graphs p and q within it do exist) then graphs contained in a proposition can be put in correspondence with other graphs not in a proposition. For example u and v can be put into correspondence with p and q as long as P1 is not in correspondence with any other proposition. In this case the subgraph unit of a proposition with its contained graphs is being generalized to just the graphs u and v contained within it. This generalization is being put into correspondence and chosen to appear in the resulting common generalization.

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. For example, given the following type hierarchy



We see that an Event and an Attribute will not be comparable concepts but the two Entities - Mobile-Entity and Stationary-Entity will be. 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 4.1. There we see Person as a generalization for [Girl] and [Person:Sue] for example. In this way a common generalization ends up being the kind of general problem solving abstraction used by researchers such as Gick and Holyoak [1983] and Greiner [1985].

4.2.2 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 described in the following paragraphs 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*. Because the minimal common generalizations are the most specific, they are the largest and have the most parts in correspondence. In order to arrive at a measure of this extent we need only count up how many of the target concepts, relations and arcs have been put into correspondence with parts of the source analogue. Since the target analogue is usually considered to be the less understood analogue and thus the less extensive analogue, it is sufficient to count what parts of it are in correspondence. This is a very simple ordering technique, but it is enough to obtain the minimal common generalizations with the largest *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. Given two analogies with equal measures of extent, the minimal common generalization of the two will be the one with the lowest supertypes for each of the concepts in correspondence. This second ordering evaluation says that for two analogies with equal extents, the "stronger" analogy is determined by minimizing the distance climbed on the type hierarchy for each pair of comparable concepts. This amounts to measuring the semantic distance of concepts; concepts which are further apart correspond to weaker analogies.

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 possible analogies, the evaluations derive the set of minimal common generalizations or "stronger" analogies.

†Actually, for convenience, the tool counts how many target arcs are not in correspondence. The opposite is presented here for clarity.

4.3 Analogical Inferencing

The Analogical Tool described thus far forms analogies which are characterized as minimal common generalizations. As discussed in §1.1.2, these analogies are useful for many purposes such as schema abstraction [Gick and Holyoak,1983], metaphor understanding [Indurkhya, 1985] and teaching [Murphy et al., 1963]. Other uses of analogy require that further reasoning take place from these analogies. This reasoning is referred to as analogical inferencing and is part of the second definition from The American Heritage's dictionary [1983] presented in §1.1.1; the inference that certain admitted resemblances imply probable further similarity.

4.3.1 Analogical Inferencing in the General Framework

In terms of the general framework, analogical inferencing is seen as forming a new minimal common generalization which is an extended specialization of an original minimal common generalization. This new specialized minimal common generalization is a result of extending one of the analogues, typically the target analogue. Because of the original similarity between the two analogues, information in the source analogue is hypothesized as existing in the target. Once this extension is made to the target, a new more specialized minimal common generalization is created which is an extension of the original and contains a generalization of the new parts in correspondence.

It must be mentioned that many possible extensions can be made to a target analogue. Decisions concerning which extensions to make are part of the particular knowledge representation chosen for the general framework.

4.3.2 Analogical Inferencing in the Analogical Tool

Analogical inferencing based on the Analogical Tool follows the same structure as described for the general framework. 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.

Choices for Inferencing

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 discussion here is restricted to only hypothesizing parts as existing in the target analogue based on their existence in the source analogue.

Choice of a minimal common generalization to be extended also brings with it the concepts that may be hypothesized. It was shown that projections from the minimal common generalization determine which concepts, relations and arcs from the analogues are in correspondence. The choice of possible hypotheses to make comes from any concepts in the source analogue which are not already in correspondence. The next choice from these is which concepts are better for hypothesizing as existing in the target.

Better choices for concepts to hypothesize will typically be connected concepts and relations, if possible. The reason is to take advantage of the concise connected properties of conceptual graphs. The very fact that concepts and relations are connected in the graphs is an important aspect of them. Information that is related to each other is shown as connected in the graphs, so we wish to hypothesize further connected information. Concepts can be seen as connected in the graphs in accordance with the formation evaluations described above, and certain hypotheses are preferred over others.

Firstly, co-reference links which express an equality relation are the most preferable concepts to hypothesize. If a source concept is part of a co-reference link, where one of the concepts is already in correspondence with a target concept, then hypothesizing the existence of the other linked concept as existing in the target will give a new labeled concept in the target. We are adding more information about a concept in the target analogue which we already know about. This is a fairly credible inference to make.

The second most preferred part to hypothesize as existing in the target analogue is a relation. In particular, there is very strong evidence for hypothesis of a relation between two concepts in a source analogue, where the two concepts are already in correspondence

with concepts in the target analogue. This can be seen as filling in a relationship, between concepts we already know about in the target analogue. For example, if the target analogue contains the concepts [Person:Joe] and [Person:Fred] and they are in correspondence with the source concepts [Person:Bill] and [Boy:Sam] respectively, and there exists the chain [Person:Bill] -> (Parent) -> [Boy:Sam] in the source, then we can hypothesize a similar Parent relation, [Person:Joe] -> (Parent) -> [Person:Fred] in the target analogue.

The next most preferred concepts to hypothesize are proposition concepts (context boxes). If concepts inside of an existing source proposition are already in correspondence with target concepts not inside of a proposition box, then there is strong evidence that this proposition box should exist in the target. The reason for the strength of this hypothesis is the knowledge that if this proposition box did exist it would complete a *contains* relation between a proposition and the graphs within it.

The next most preferred concepts to hypothesize are the reverse of the above. If two proposition boxes are in correspondence between the source and target analogues, then there is strong evidence for hypothesizing the existence of a concept inside of the source proposition box as existing inside of the corresponding target proposition box. The strength of the evidence again comes from the *contains* relation which exists between a proposition concept and the graphs contained in it. Because of the correspondence between the propositions, this relation is already partially set up.

If none of the concepts fits into the above choices then the next concept to be hypothesized should be one which is attached to a relation that is already in correspondence with a target relation. If the relations are already in correspondence then the relationship already partially exists. For example, if [Girl]<-(Agent)<-[Eat] exists in the source analogue, and [Person:Sue]<-(Agent) exists in the target analogue and is in correspondence with the above girl eating, then there is evidence for hypothesizing the [Eat] concept or a generalization of it in the target analogue because we already know that Sue is the agent of something.

Farther down on the preference list of concepts to hypothesize are those which are not directly connected, but which would be connected if we also hypothesized a relation as existing. For instance, in the above example if only [Girl] and [Person:Sue] were in correspondence and (Agent)<-[Eat] as part of [Girl]<-(Agent)<-[Eat] in the source analogue existed but was not in correspondence with anything, then first hypothesizing the

relation (Agent) in the target analogue would allow hypothesis of [Eat], resulting in [Person:Sue]<-(Agent)<-[Eat] in the target.

The final choice for concepts to hypothesize from the source analogue to the target analogue are those which are totally unconnected to any concepts already in correspondence. These concepts are seen as adding new information to the target based on information in the source, but the information added is not connected to what is already seen as similar in the two analogues. Again, because of the significance of connectivity in conceptual graphs, these kinds of hypotheses will not be as favored as those described above.

Once a concept is inferred as existing in the target analogue, it must be checked against the four formation evaluations to ensure none of them have been violated. Only when these concepts are validated can the new extended minimal common generalization come into existence. From this point, either more concepts can be hypothesized or if the concept does not conform to the formation evaluations, a new concept can be hypothesized.

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* can give 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.

Questions for Inferencing

Although a pattern for the choice of concepts to be hypothesized has been discussed, what is more difficult to determine in most cases is what type label should be attached to the concepts. The discussion above specifies plausible structural choices for analogical inferencing but leaves open the semantic choices for concept labelling. This section provides guidance on what labels to choose but in no way proposes to be a complete solution to the problem.

If we are hypothesizing a concept which will be linked to another concept by a co-reference link, we know what the label on the new concept in the target is; it is the same as its co-referent. Otherwise we must take the source label of the concept which is providing the hypothesized concept. A better solution can be found if the inferencing is part of a system with input from a user. If a user is part of the system then it is possible to ask them if the source label is appropriate or to query them about other label possibilities obtained by climbing the type hierarchy. Because graphs are generalized using the type hierarchy, it is reasonable to suggest terms which are generalizations of the source label.

Whichever way a label is given to a concept, the continuation of inferencing from the original analogy may prove to resolve the uncertainty of the choice. This uncertainty will be resolved if a later hypothesis is part of a co-reference chain of concepts. Any concepts which would later be found to be part of a co-reference chain would automatically resolve the label of a concept.

Outside of the suggestions made so far, the only other way to choose the label given to a newly hypothesized target concept is to use further background knowledge. Because conceptual graphs contain more aspects than are currently required by the Analogical Tool, it is possible to address this. Chapter 7 contains enhancements to the tool which may be applicable.

4.4 Object-Oriented Formulation of the Algorithm

This section discusses the algorithms that implement the Analogical Tool. A description is given of the object-oriented aspects of the tool and then a general overview of how the tool works is given. The final part of this section presents pseudocode for the general search algorithm as well as for the evaluations.

4.4.1 An Object-Oriented Analogical Tool

The Analogical Tool is written in the objected-oriented language of Smalltalk [Goldberg and Robson, 1983]. Object-oriented languages encourage modularity and re-use of code and are based on the idea of objects, which correspond to objects in the real world. Objects define a unit of modularity which have their own state and their own behaviors defined as methods. Smalltalk supports inheritance and communication between objects is performed

by passing messages that correspond to certain methods attached to an object. The Analogical Tool implementation forms analogies given two analogues as input and supports analogical inferencing based on the tool, but does not perform this inferencing automatically.

The Analogical Tool has two parts. The first, allows for conceptual graphs to be visually displayed on a screen. The second, searches for common generalizations or analogies and orders them to derive minimal common generalizations. The first part of the tool is an extension of a program called NodeGraph [Adams, 1987]. NodeGraph allows a user to draw graphs made of nodes and arcs. The nodes and arcs can be labeled and graphs can be edited, scrolled on the screen and saved to a file. The Analogical Tool extension to NodeGraph allows for all elements of conceptual graphs to be drawn. Using this part of the tool, analogues represented as conceptual graphs can be drawn, edited and saved to a file. There are various classes within the Analogical Tool which support this visual display of conceptual graphs and give it a good user interface.

In the second part of the tool, an Analogy class exists which contains methods for searching for the analogies and methods which correspond to the evaluation criteria. Each analogy formed is an object which contains the concepts put into correspondence between the two analogues, and a measure determined by the ordering evaluations. This part of the tool also contains a TypeHierarchy class with subclasses that correspond to the concepts contained in the analogues. This TypeHierarchy class contains a method that corresponds to the comparable concepts evaluation criterion that determines if any two concepts are comparable. Because the concepts are actually subclasses in the TypeHierarchy, it is possible to enhance the Analogical Tool by adding type definitions, schemata and prototypes to the concepts in an object oriented manner.

4.4.2 About the Algorithm

The basic algorithm of the Analogical Tool 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 ordering measures.

This section describes the search algorithm and gives a description of the space the tool searches over. Because complexity is not felt to be a main issue of this thesis, there will be no attempt to quantify the exact nature of the search space. The purpose of this thesis is to determine fundamental properties of analogy and to design an Analogical Tool which embodies these properties.

As analogies or common generalizations are being formed, potential concept correspondences are considered. These must pass all formation evaluations before the concepts can be put into correspondence with each other. The common generalizations are then ordered using the *extent* and *semantic distance* measures. The *extent* measure is first applied to all analogies formed. The *semantic distance* measure is used to break any ties between analogies with equal extents. In this way the set of minimal common generalizations is determined. At this point if an analogy is extensible, analogical inferencing could take place.

The search algorithm which is a method of the class Analogy, potentially considers all possible correspondences for each target analogue concept with each source analogue concept. This search is over all possible common generalizations, which means over all possible extents of common generalizations. This space is bounded by the common generalization which contains no correspondences as the upper bound and by a lower bound which has all target concepts in correspondence. The potential search space that results is intractable but much of it is pruned using the formation evaluations, giving an algorithm which completes in several minutes for moderately sized examples.

The evaluation criteria used can be seen as fairly strong constraints and each failed potential correspondence prunes an entire branch of the search tree. This pruning is compounded by the fact that the search is not just locally controlled but globally controlled. As the search progresses, choices made for initial corresponding concepts further constrains what other concepts can be put into correspondence for each analogy. This globally constraining feature is a result of the way the formation evaluations carve up the analogues in order to specify parts in the common generalizations.

Although at present the tool forms all plausible analogies which conform to the formation evaluations and then orders these, it is fairly simple to decrease the search space

even further. This can be done by specifying that the tool search for the 5 or 10† most extensive analogies. In this way the first 5 or 10 analogies are derived and updated as the search continues. Searches for subsequent analogies can be halted if their extent measures are higher than these. Because the extent ordering evaluation actually measures how many of the target concepts are not in correspondence with source concepts, we can use this measure to terminate searching in less plausible, less maximal common generalizations. This addition to the searching algorithm corresponds to setting up a threshold level beyond which no extra search time is spent.

4.4.3 PseudoCode for the Analogical Tool

This section presents pseudocode for the major parts of the Analogical Tool. The code is included here to show how the pieces described previously, fit together. Presenting the actual pseudocode also helps to show the very modular, simple nature of the tool which is in part due to its object-oriented implementation. Each of the evaluations are clearly defined and addition of automatic analogical inferencing is seen as another module.

Use of the Analogical Tool starts with first drawing the analogues represented as conceptual graphs using the drawing element of the tool. Once the analogues are drawn a menu selection asks for the analogies to be formed. The tool then sorts the arcs belonging to the source and target analogues into separate lists. After this some initialization is performed to set up dictionaries for concept correspondences. Following, the main search method for analogies is performed. The code for this method which is called *findMore* is presented first, followed by code for the ordering and formation evaluations.

```
findMore:tArc Index: targetIndex  -  recursive search method
- if  a full analogy has been formed then evaluate using ordering evaluations and show
      the analogy
- else
      - get information about the targetArc - like nodes attached to it
      - until all possible matches for this targetArc found do:
            -until correspondence formed or we're at the end of the sourceArcs do:
                  -get information about the next sourceArc - like nodes attached to it
                  -if the sourceArc not already in correspondence then
                  -if same relation evaluation is true then
                  -if proposition evaluation is true then
```

† The choice of 5 or 10 here is arbitrary. In experiments, the size of the set of minimal common generalizations is typically less than 10 so this should ensure that the entire set will still be found.

-if co-referent links evaluation is true then
-if concepts comparable evaluation is true then
-update correspondence dictionaries
-findMore: next targetArc Index: index + 1
-get next sourceArc

Ordering and Formation Evaluations:

setConnectivity (extent ordering evaluation)
- count the number of targetArcs not in correspondence with any sourceArcs. - 0 reflects that all targetArcs are in correspondence.

semanticDistanceClimbed (semantic Distance ordering evaluation)
- count height of common generalization from lowest concept - for each correspondence formed and add these up.

*overlapCompatible:*targetRelation :sourceRelation (same relation formation evaluation)
-if this targetRelation is already in correspondence with a sourceRelation by virtue of the other arc attached to it then
-if I am re-enforcing the same correspondence then
return true
-else
return false

contextCompatible: targetConcept :sourceConcept (proposition formation evaluation)
-is targetConcept inside a proposition?
ifTrue: is the proposition node I am in in correspondence with anything?
ifTrue: is sourceConcept inside of that corresponding source Proposition?
ifTrue: return true
ifFalse: return false
ifFalse: is the sourceConcept in a proposition node?
ifTrue: is this proposition in correspondence with anything?
ifTrue: return false
ifFalse: return true
ifFalse: are any other targetConcepts in this proposition in correspondence with any sourceConcepts?
ifTrue: are these sourceConcepts in a proposition?
ifTrue: return false
ifFalse: return true
ifFalse: return true
ifFalse: is the sourceConcept inside of a proposition node?
ifTrue: are any other sourceConcepts in this proposition in correspondence with any targetConcepts?
ifTrue: are these targetConcepts in a proposition?
ifTrue: return false
ifFalse: return true
ifFalse: return true
ifFalse: return true

co-referentCompatible: targetConcept :sourceConcept (co-referent links formation evaluation)
-is the targetConcept in a co-referent chain?
ifTrue: are any of my co-referent partners in correspondence?
ifTrue: is sourceConcept in the corresponding co-referent chain?
ifTrue: return true
ifFalse: return false
ifFalse: is sourceConcept in a co-referent chain?
ifTrue: are any source co-referent partners in correspondence?
ifTrue: return false
ifFalse: return true
ifFalse: return true
ifFalse: is the sourceConcept in a co-referent chain?
ifTrue: are any source co-referent partners in correspondence?
ifTrue: return false
ifFalse: return true
ifFalse: return true

comparable: targetConcept :sourceConcept (concepts comparable formation evaluation)
-until generalization found or we're at the top of the typeHierarchy do:
is targetLabel = sourceLabel
ifTrue: return true
ifFalse: is targetLabel a generalization of sourceLabel?
ifTrue: return true
ifFalse: targetLabel becomes one step generalization of targetLabel

TypeHierarchy

All concepts used for the examples in chapters 5 and 6 are contained in one type hierarchy. Because this hierarchy is large, only a portion of it is shown in Figure 4.2. The beginnings of this type hierarchy are taken from Appendix B of "Conceptual Structures" [Sowa, 1984]. The remaining ordering of types in this hierarchy are my own and in no way purports to being the only or best way of ordering them. This goes along with Sowa's model which say that we each have our own type hierarchy which has been built up over the years and is constantly being added to, deleted from and re-organized.

In summary, this chapter has described the Analogical Tool in terms of how each of the notions, "part-of", "generalization" and a "generalization ordering" relation of the general framework of analogy correspond to parts of the conceptual graph model. These correspondences are used to develop a computational tool which forms analogies and supports analogical inferencing based on them. Four aspects of the tool have been described. First, it was shown that the tool forms minimal common generalizations as

preferred "stronger" analogies. Second, analogy formation was shown to be under the control of four formation evaluations and two ordering evaluations. The formation evaluations operate to determine how and what parts of the analogues are generalized, and the two ordering evaluations partially order the common generalizations. The third section showed that analogical inferencing based on the tool can proceed by extension of the target analogue and formation of a new minimal common generalization. Fourth, the Analogical Tool was described in its most specific form - as an object oriented implementation in the language of Smalltalk [Goldberg, Robson, 1983].

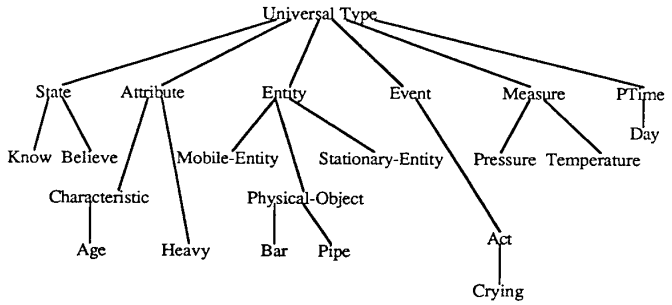


Figure 4.2 A Portion of the Type Hierarchy

Chapter 5
Testing the Analogical Tool

This chapter shows the Analogical Tool working on four examples of analogy taken from significant researchers in the field. The purpose of this chapter is to show how the Analogical Tool compares to work done by other researchers. Each example displays a different aspect of the tool and different forms of analogy formation and analogical inferencing.

For each example a description is first given, stating its significance and origin. A display of the screen from the Analogical Tool is shown next and the conceptual graphs of the analogues described. The output from the tool is then shown along with a discussion of the results. Each section concludes with a comparison of the other researcher's results with that of the Analogical Tool.

In most examples in this chapter there are two analogues represented by conceptual graphs in one figure. One analogue is shown in dashed nodes at the bottom of the figure. This is the target analogue, the analogue which is typically less well understood than the other source analogue. In all graphs co-reference links are shown by the angled links between concepts. Arcs which serve to connect relation nodes to concept nodes are shown as straight lines.

5.1 Example 1 - A Simple Teaching/Irrigating Analogy - CYC

This first very simple example is presented to show what the input and output of the Analogical Tool look like and how the evaluation criteria work. The example is significant because it comes from an important project in Artificial Intelligence today; the CYC project being conducted at MCC (Microelectronics and Computer Technology Corporation) [Lenat et al, 1986]. This project is building a large knowledge base of real world facts, heuristics and methods for reasoning over the knowledge base. The knowledge representation used in CYC is frame based and is similar to RLL [Greiner & Lenat, 1980] and KRL [Bobrow & Winograd, 1977]. The designers of CYC believe that common sense reasoning and analogy can widen the knowledge acquisition bottleneck of such a system. To this end the

authors plan to use analogy in several ways in CYC. Firstly, CYC uses analogy in a copy&edit format. As more knowledge is being added to CYC's knowledge base, it is easier to find a similar chunk, copy it, and then edit that copy than to enter that new chunk into the system from scratch. CYC also plans to use analogy to suggest or help the user choose particular existing frames from which to copy and also to do some of the editing itself.

Designers of CYC have defined four dimensions along which analogy can be classified as discussed in §2.1.3. The example shown in Figure 5.1 comes from the first dimension defined; variations in the level and degree of matching. In particular, two frames in the CYC knowledge base can have identically named slots whose values are not quite identical; the resultant mapping becomes part of the analogy. Figure 5.1 represents an analogy between Teaching and Irrigating. This example is described as follows by the CYC project team [Lenat et al., 1986, pp.69]:

> "Irrigating and Teaching are analogous in this fashion, since Teaching has a recipients slot filled with Students, and an aidsRecipientIn slot filled with Learning. This is the analogy that Teachers distribute knowledge to aid their students in learning, just as irrigation systems distribute water to aid plants in growing. This analogy will, in turn, either suggest the "Learning as Growing" analogy, or if it already exists, be strengthened by it."

5.1.1 Analogy Formation

The sentence above; Teachers distribute knowledge to aid their students in learning, is represented in Figure 5.1 by the top conceptual graph. Here we see Distribute having an agent Teacher, an object Knowledge, a recipient Students and a purpose being the proposition containing: Aiding Students in Learning. The second sentence above; Irrigation systems distribute water to aid plants in growing is represented in Figure 5.1 by the bottom dashed conceptual graph. This graph is very similar to the top graph.

The contents of Figure 5.1 are given to the Analogical Tool which produces the output in Figure 5.2. For this example there is only one minimal common generalization formed with an extent measure of 0†. Because only one minimal common generalization is formed, there is no need to use the Semantic Distance measure. Figure 5.2 shows how the

† a measure of 0 here means that all concepts in the target analogue were matched with source concepts.

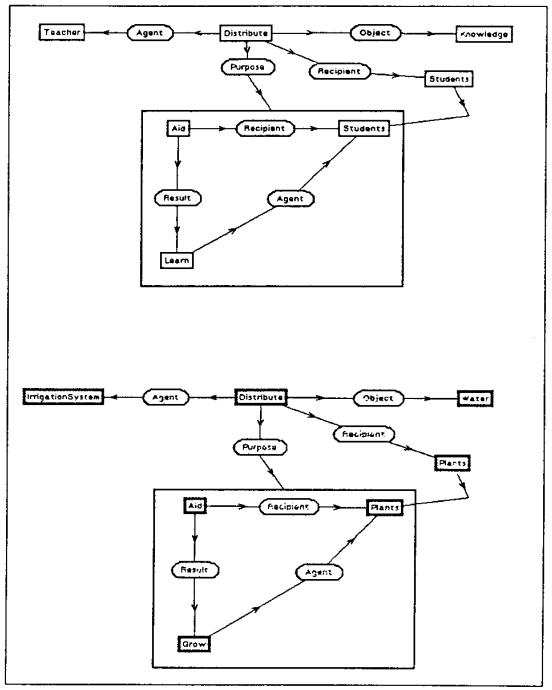


Figure 5.1 Input Analogues to the CYC Example

target analogue concepts correspond to source analogue concepts. Although this is a simple example, it contains enough structure to require use of all four formation evaluation criteria. There is the requirement of maintaining same relations in matches, maintaining co-reference links (Students-Students and Plants-Plants), maintaining the *contains* relation for proposition nodes and maintaining comparable concepts according to the type hierarchy. Other matches are formed in other common generalizations such as [Aid] -> (Recipient) -> [Plants] matching to the chain: [Distribute] -> (Recipient) -> [Students], but the resulting analogy with this match in it has a much higher extent measure (i.e. fewer matched target nodes) than the minimal common generalization formed. Again, the minimal common generalization with extent 0 from this example represents the analogy which expresses the most similarity between the analogues.

```
ANALOGY CONCEPT CORRESPONDENCES
Plants  -->  Students
Proposition  -->  Proposition
Plants  -->  Students
Distribute  -->  Distribute
Water  -->  Knowledge
Grow  -->  Learn
Aid  -->  Aid
IrrigationSystem  -->  Teacher
EXTENT MEASURE  --  0
SEMANTIC DISTANCE MEASURE  --  8
```

Figure 5.2 Output from the CYC Example

The minimal common generalization formed in this example totally matches the target concepts with source concepts. Because of this, no further reasoning takes place. Further analogical inferencing would occur if the source analogue contained further information which could be hypothesized as existing in the target. Examples to follow contain this additional information, and show how inferencing can occur.

5.1.2 Comparison of CYC to the Analogical Tool

This first simple example forms the expected analogy, shows how the Analogical tool operates and gives its input and output. The "Learning as Growing" analogy described by CYC researchers is formed and shown by the correspondence of Learn to Grow in the tool's output. The analogy formed by the tool actually gives a more complete description of the correspondences and hence the analogy, than the quote taken from the CYC paper. Here, we not only see that Learning is like Growing but that Teachers are like the IrrigatingSystems, Knowledge is like Water and Students are like Plants.

5.2 Example 2 - Schema Abstraction - Gick and Holyoak

The second example is taken from work on schema abstraction by Gick and Holyoak [1983]. This example is significant because it shows the ability of the Analogical Tool to form general problem solving schemas and shows how analogical inferencing can proceed based on the Analogical Tool.

In their work on schema abstraction, Gick and Holyoak study the process of analogical thinking in humans. They predict that "induction of a general schema from concrete analogues will facilitate analogical transfer". To test this prediction subjects first read one or more stories illustrating problems and their solutions and then attempt to solve a disparate but analogous transfer problem. It was found that if two prior analogues are given, subjects often derived a general problem solving schema as an incidental product of describing similarities of the analogues. The quality of the induced schema is highly predictive of subsequent solving of other analogous problems.

The hypothesis that Gick and Holyoak propose in this work is the importance of forming general problem solving schemas and their later use for solving similar problems in possibly disparate domains. The analogues used in this example are a military problem and an analogous radiation problem. This example is used in three ways by the Analogical Tool: first, to form a minimal common generalization and the resulting concept correspondences; second, to form a general problem solving schema and third to show how analogical inferencing based on the tool can proceed using previous output of the tool.

5.2.1 Analogy Formation

Figure 5.3 shows the two analogues represented as conceptual graphs. The top conceptual graph corresponds to the following four sentences in the military problem:

> A Commander wants to Capture a Fortress.
>
> The Commander Possesses a Large Army.
>
> The Army cannot Go Down One Road (because they would be noticed).
>
> A Solution is to send Parts of the Army along Many Roads Simultaneously.

The analogous radiation problem is shown by the dashed conceptual graph at the bottom of Figure 5.3. Here, 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.

The angled arcs represent co-reference links and show that the mention of Army in the top graph refers to the same army in each sentence. Also, the first time Commander is mentioned he is "A" Commander and the second time by use of the co-reference link he is "The" Commander. These links not only form one of the formation evaluation criteria but also serve to tie the analogue together. The output from Figure 5.3 is shown in Figure 5.4. Here we see what concepts have been put into correspondence in the single minimal common generalization formed. This analogy has an extent of 0 (i.e. all target concepts have been matched) and a semantic distance measure of 14. Because there is only one minimal common generalization formed, the semantic distance measure is not needed. Formation of this stronger analogy required the use of all four formation evaluation criteria, same relations, co-reference links, propositions and comparable concepts. It should also be noted that although the analogous sentences in the two analogues have been presented here in the same order, it makes no difference to the tool what order they are presented in. The tool finds the same minimal common generalization independent of the sentence ordering.

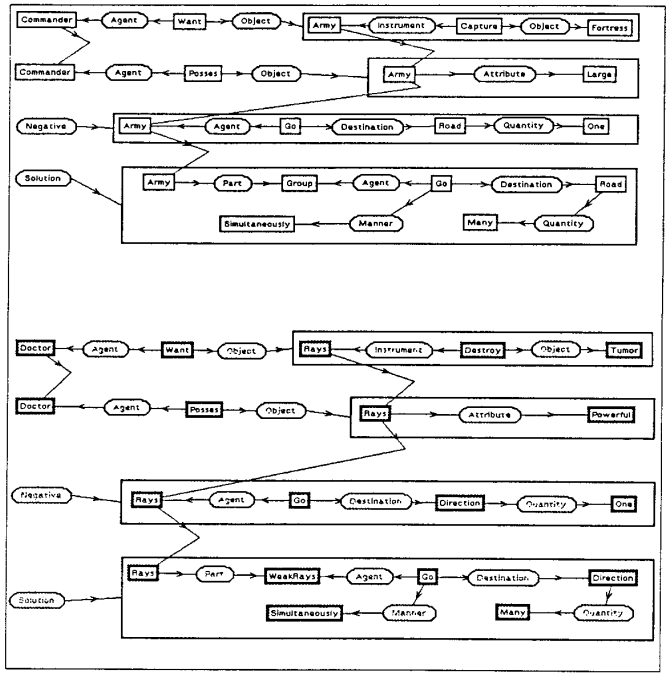


Figure 5.3 Input Analogues to the Gick and Holyoak Example

ANALOGY CONCEPT CORRESPONDENCES
Simultaneously --> Simultaneously
Proposition --> Proposition
Rays --> Army
Many --> Many
Doctor --> Commander
Rays --> Army
Want --> Want
Proposition --> Proposition
Rays --> Army
Doctor --> Commander
Proposition --> Proposition
Direction --> Road
Go --> Go
One --> One
Tumor --> Fortress
Rays --> Army
Direction --> Road
Possesses --> Possesses
Proposition --> Proposition
Destroy --> Capture
Powerful --> Large
WeakRays --> Group
Go --> Go
EXTENT MEASURE -- 0
SEMANTIC DISTANCE MEASURE -- 14

Figure 5.4 Output from the Gick and Holyoak Example

5.2.2 Schema Abstraction

In addition to the Analogical Tool forming the concept correspondences and ordering them, it also forms the kind of general problem solving principle or schema abstraction Gick and Holyoak describe as discussed in §1.1.2. Through the use of the type hierarchy and the comparable concepts evaluation criterion, a minimal common generalization of each pair of corresponding concepts is found. As the comparable concepts evaluation is deciding if two concepts are comparable it climbs the type hierarchy until it reaches either a common

generalization of the two concepts or the universal type. If the universal type is reached the concepts are not comparable, but if a minimal generalization of the two is found it can be used for forming a schema abstraction or general problem solving principle. When these generalized concepts are added to the structural aspect of the minimal common generalization, a general problem solving principle is formed. For this example the principle 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.
>
> A Solution is for Parts of the Mobile Entity to Go along Many Paths Simultaneously.

Person is the common generalization of Commander and Doctor; Mobile Entity is the generalization of Army and Rays; Violent act is the generalization of Capture and Destroy, etc. This principle is very similar to the one generated by Gick and Holyoak:

> "If the goal is to use force to overcome a central target and you have a sufficiently great force, and you are unable to apply full force along one path, then apply weak forces along multiple paths simultaneously."

The Analogical Tool has thus succeeded in automating the kind of process that Gick and Holyoak describe as being an important part of analogical transfer. This kind of transfer is especially important in teaching and learning situations. When a teacher is trying to teach a general principle to a student he or she often gives several examples of the principle with the hope that the student will find the analogy (or the teacher may make it explicit), abstract the principle and then be able to use it in other similar situations.

5.2.3 Analogical Inferencing

This same example above can be used to show how analogical inferencing can proceed based on the tool. As shown in Figure 5.5 the analogues are the same with the exception that the solution to the radiation problem is unknown. This is represented as an empty Proposition node attached to the Solution relation and can be thought of as a query to the source analogue (military problem) about the target analogue (radiation problem). Correspondences resulting from the strongest analogy formed are the same as the last

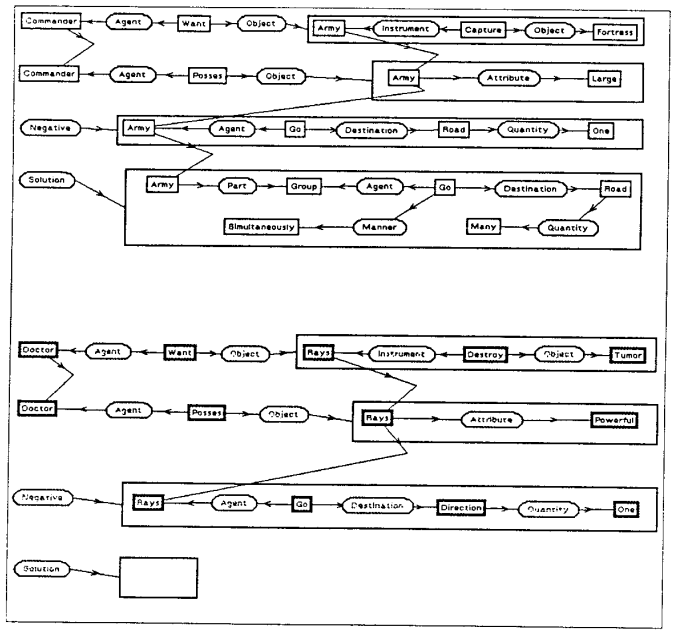


Figure 5.5 Input Analogues to Gick and Holyoak Example Showing Inferencing

output with the difference that concepts inside of the source solution proposition have no correspondences in the target solution proposition. In this case the empty proposition node from the target analogue is matched to the solution proposition in the military problem. With these two propositions matched, a solution to the radiation problem can be hypothesized based on the solution to the military problem. Although the tool at present does not automatically perform this inferencing process, the following section discusses how it would occur.

Analogical inferencing in this example proceeds as discussed in §4.3. There it was shown that information is hypothesized as existing in the target analogue based on its existence in the source analogue. Connected concepts and relations are the preferred hypotheses and within these an ordering was given. In general, the entire structure of the solution to the military problem is hypothesized as being a solution to the radiation problem. Still to be decided is the semantic aspect of the solution; that is what are the labels of the concepts in the solution to the radiation problem. To answer this we follow the inferencing steps for hypothesizing concepts and relations.

The first possible nodes to be hypothesized are those attached to co-referent chains. Because the military solution contains an Army concept which is part of a co-referent chain, and because Army has been put into correspondence with Rays in the minimal common generalization, we can hypothesize Rays as existing in the solution to the radiation problem. Once the Rays concept is hypothesized it needs to be checked against the evaluations. There is no problem so a new minimal common generalization is formed which contains a generalization of this new correspondence between Army and Rays, i.e. Mobile-Entity. There are no more co-referent concepts which can be hypothesized so the next possible hypotheses are relations which exist in the source analogue between two concepts in correspondence with target concepts. There are no relations like this so the the relation attached to the Army concept - the Part relation, is hypothesized as existing in the target. Any attached relation can be hypothesized and labeled because of the same relations evaluation criterion. There are no possible propositions to hypothesize so from here we can only hypothesize concept nodes and relation nodes. The next node to be hypothesized is the Group concept node from the source analogue. This is the next to be hypothesized because it is connected to the Part relation. Because there is no co-referent connection to the Group concept in the source analogue we are not sure how to label this concept in the

target analogue. At this point we have hypothesized the following new parts as existing in the target analogue: [Rays] -> (Part) -> [Group?]. The question mark is inserted to emphasize that we have no certainty about the label for this concept. Here we turn to the questions for inferencing in §4.3.2 for possible answers.

We have three choices for labeling this [Group?] concept which fit within the structure of our instantiated general framework. First, we can just use the label from the source analogue. This technique has been used by other researchers such as Gentner [1988] and Indurkhya [1985]. Our second choice is to use our type hierarchy to suggest a label. If we label the concept with a generalization of Group then we are obeying the comparable concepts evaluation criteria, so this is a valid label to suggest. Our third choice for a label is to use a type label which is a specialization of a generalization of Group. These choices use the type hierarchy are shown in Figure 5.6. In the third choice, Entity is a Generalization of Group and has many specializations which could all be possible labels. We could hypothesize Mobile Entity, Stationary Entity, or go further down in the hierarchy. If our system has input from a user we can rely on them to pick from these possible valid choices. With no input from a user we probably want to use the source label or use the maximal generalization of the source label that is not the universal type. This ensures a valid choice of label but may not be very close to the correct label. Again, it should be mentioned that these are suggestions for what is a very difficult problem with no clear solution. Because we are in the area of plausible reasoning there is very little evidence to support choices for concept labels. We continue hypothesizing the Go concept, the Road concept, the Simultaneously concept, the Many concept and all the relations between them. With each of these concepts the same question of labeling exists. Fortunately, this labeling question may not be as bad as it first appears. If you told a person that "the solution to the radiation problem" is "to send Groups of the Rays along Many Roads Simultaneously," they would probably know what you meant. So, for any use of this inferencing that has some interaction with a human the results are probably understandable. Only if inferencing is done in a totally automated environment with no input from a user will this be a major problem.

5.2.4 Comparison of Gick and Holyoak With the Analogical Tool

As stated in the section on forming general problem solving principles, the Analogical Tool seems to capture what Gick and Holyoak describe. The tool forms the same strongest

analogy as Gick and Holyoak and also forms a similar general principle with some difference of terms. The tool can also form further analogies between the general principle and other analogous problems and reason to solve these new similar problems. In particular, the labeling problem would all but disappear when using the general principle because we can use the very general labels from it and directly carry them over to a target analogue. Gick and Holyoak also discuss the importance of analogical transfer in teaching and learning situations and as shown, the Analogical Tool can be useful in these situations. In addition, this example has shown how analogical inferencing can proceed from an initial analogy. A solution for the radiation analogue was hypothesized based on information from the military analogue.

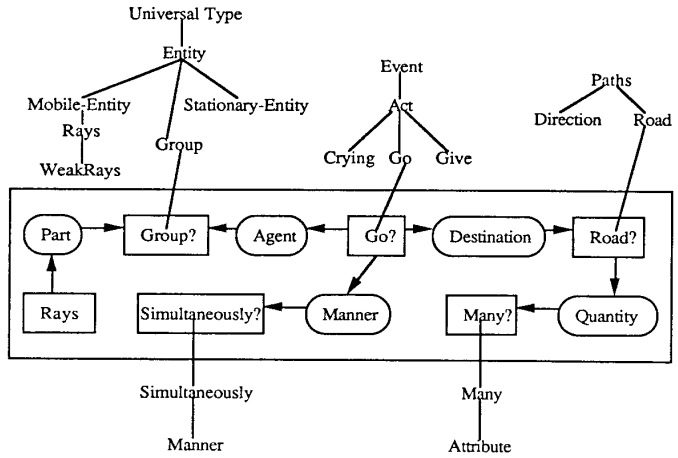


Figure 5.6 Use of the Type Hierarchy for Labelling of Concepts

5.3 Example 3 - The Heat/Water Analogy - Dedre Gentner

The third example comes from a paper by Gentner and Seziorski on uses of analogy in science [Gentner and Seziorski, 1986]. The example is significant because it represents a real life scientific use of analogy, and presents us with a more complex representation than our earlier examples. This example differs from the others because it requires the use of nested propositions; i.e. stating propositions within propositions.

Analogy has been an important part of scientific thinking and Gentner and Seziorski use this paper to look at the styles of analogizing practiced by scientists at different points throughout history. The most modern example in the paper is the heat/water analogy used by the French thermodynamicist Carnot (1796-1832). The explanation given in this example shows how Carnot actually used the analogy.

Carnot is responsible for the Carnot cycle for heat engines which is still taught as an ideal energy conversion system. In his description of work surrounding this system, Carnot presented a powerful analogy between heat and water that clarified his positions and generated new testable questions. Gentner and Seziorski describes Carnot's use of analogy as follows [Gentner and Seziorski, 1986 pp.10-12]:

> "Early in his Reflexions [Carnot, 1977], Carnot introduces the analogy between water falling through a waterfall and caloric (heat) falling through a heat engine. The basic notion of an analogy between heat and fluid was not new. Indeed, the dominant theory of heat at the time was the caloric theory, which defined heat as a weightless fluid that shared certain properties of ordinary matter. Like other matter, caloric was a conserved quantity, incapable of being created or destroyed. Thus the idea of some commonality between heat and water was not new with Carnot, since both are instantiations of the common abstraction that both are fluids. What was new was the thoroughness of his development of the analogy -- the extent to which explicit causal structures from the water domain were applied in the heat domain."

Carnot uses the analogy to set forth the principles of a heat engine, and then derives further insights about the motive power of the engine by analyzing the system of relations in a water engine.

> [1] According to established principles at the present time, we can compare with sufficient accuracy the motive power of heat to that of a waterfall. Each has a maximum that we cannot exceed, whatever may be, in the one hand, the machine which is acted upon by the water, and whatever, on the other hand, the substance acted upon by the heat.
>
> [2] The motive power of a waterfall depends on its height and on the quantity of the liquid; the motive power of heat depends also on the quantity of caloric used, and on what may be termed, on what in fact we will call, the height of its fall, that is to say, the difference of temperature of the bodies between the higher and lower reservoirs.
>
> [3] In the waterfall the motive power is exactly proportional to the difference of level between the higher and lower reservoirs. In the fall of caloric the motive power undoubtedly increases with the difference of temperature between the warm and the cold bodies; but we do not know whether it is proportional to this difference. We do not know, for example, whether the fall of caloric from 100 to 50 degrees furnishes more or less motive power than the fall of this same caloric from 50 to zero. It is a question which we propose to examine hereafter."

This example, shown in Figure 5.7 describes the knowledge Carnot had of both the heat flow and water flow situations.

5.3.1 Analogy Formation

The target analogue corresponding to the less well understood heat situation is represented by the dashed conceptual graph at the bottom of Figure 5.7. The following sentences describe this analogue:

> A source contains water which has a characteristic height.
>
> This water flows from the source to the destination which also has a characteristic height.
>
> The height of the source is higher than the height of the destination and the difference in this height is a positive value.
>
> The power produced from the water flowing depends on the amount of water.
>
> The power produced also depends on the difference in heights between the source and the destination.
>
> There is a maximum amount of power that is produced from this method.

The conceptual graph at the top of Figure 5.7 represents the water situation and is identical to the above sentences with the addition of two "ProportionalTo" relations. In the

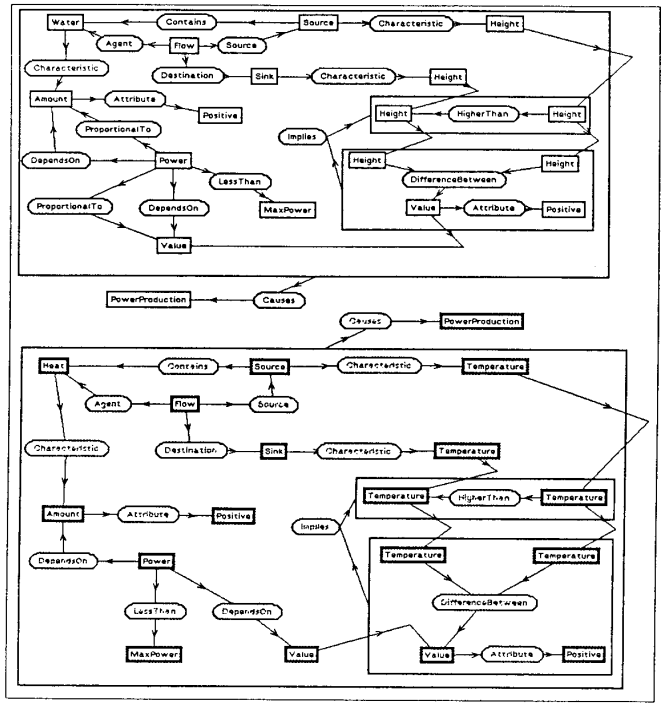


Figure 5.7 Input Analogues to the Gentner Example

water situation we also know that the Power is ProportionalTo the Amount of Water and the Power is ProportionalTo the Difference between the two Heights. This is information we do not know about the heat situation. When the Analogical Tool is presented with this input it derives one minimal common generalization as shown in Figure 5.8. Here we see that all of the target heat concepts have been matched with source concepts. All four of the formation evaluations were used to derive this analogy and the existence of nested propositions has been no trouble for the Analogical Tool.

ANALOGY CONCEPT CORRESPONDENCES
Source --> Source
Positive --> Positive
Temperature --> Height
Flow --> Flow
Sink --> Sink
Proposition --> Proposition
Power --> Power
Positive --> Positive
Temperature --> Height
Temperature --> Height
Proposition --> Proposition
Amount --> Amount
Proposition --> Proposition
Temperature --> Height
MaxPower --> MaxPower
Temperature --> Height
Heat --> Water
Temperature --> Height
Value --> Value
PowerProduction --> PowerProduction
Value --> Value
EXTENT MEASURE -- 0
SEMANTIC DISTANCE MEASURE -- 9

Figure 5.8 Output from the Gentner Example

This formation of the original analogy exactly corresponds to the description of the analogy formation used by Carnot and described in paragraphs numbered 1 and 2 above. Carnot first establishes a correspondence between the fact that power produced in both the heat and water situations have a maximum level. Secondly, Carnot realizes that the motive power of a waterfall depends on its height and quantity of liquid just as the motive power of heat depends on the quantity of caloric used and the temperature difference between two bodies. With this analogy formed, Carnot recognizes that there is more related information he knows about the water situation that does not correspond to the heat situation. Because there is information in the source analogue which is not in the target analogue, it is possible for analogical inferencing to take place.

5.3.2 Analogical Inferencing

The analogical inferencing which can proceed from the minimal common generalization formed is exactly the reasoning Carnot used. In paragraph number 3 above Carnot notices that the information missing in the target analogue is the "ProportionalTo" relations. Carnot hypothesizes their existence and plans to test these hypotheses. Inferencing based on the tool can hypothesize the same two ProportionalTo relations quite easily. These two hypotheses are valid because they are relations between concepts already in correspondence. In the target analogue, Power has been put in correspondence with Power in the source analogue, Amount in the target has been put into correspondence with Amount in the source and Value in the target has been put in correspondence with Value in the source. The ProportionalTo relations being hypothesized from the source to the target are just relations between these three concepts. Thus output of the Analogical Tool can be used to extend the original analogy in the same way as Carnot and as Gentner's structure-mapping program.

5.3.3 Comparison of Gentner's Theory With the Analogical Tool

The Analogical Tool forms the same initial analogy as Carnot and Gentner. The analogical inferencing which can be derived is again the same as Carnot and Gentner. This example has been included for two purposes, one to present a real life analogy and show how the reasoning from it took place, and two, to present a more complex representation of

analogues. Both the inferencing based on the Analogical Tool and Gentner's system perform well for this example and lead to hypothesis of the correct missing relations.

5.4 Example 4 - The Sky is Crying Metaphor - Bipin Indurkhya

The final example of this chapter comes form Indurkhya's Phd thesis [1985] on metaphor comprehension and analogical reasoning. This example shows the communication aspect of metaphor which asks us to explore the many possible meanings of it. For this it is not only the minimal common generalizations we seek. We may also want to consider non-minimal common generalizations that allow us to understand the metaphor more fully.

In his thesis, Indurkhya proposes AST (Approximate Semantic Transference) as a theory for metaphor comprehension. This theory uses Schema Language to define source and target domains. With these domains defined, AST forms AT-MAPS. Given an AT-MAP and a partial structure of a source domain we can think of transferring this structure from the source to the target by replacing all the source domain tokens by their corresponding target tokens as given by the mapping. In this transformation the structural relationships among the source domain tokens are preserved but the tokens themselves are replaced. Indurkhya states that "a mapping is coherent with respect to a partial structure of the source domain if the structure of the target domain is consistent with the transformed partial structure of the source domain" [Indurkhya, 1985, p.8]. Two operators, Augmentation and Positing structure in AST, allow for addition of structure to the target domain that is constrained by the structure of the source domain.

The example metaphor used here is: "the sky is crying". As with many metaphors, this literally false statement is asking us to try and understand what it means. As stated by Lakoff and Johnson [1980], "there is a coherent structural analogy lying at the heart of most analogies". To find it we must express what we know about the source domain sky and the target domain crying and then search for the coherent structural analogy.

5.4.1 Analogy Formation

Figures 5.9 and 5.10 give representations of the target and source domains in conceptual graphs. At the bottom of Figure 5.9 is a representation of the target crying domain which represents the following information:

The result of a person crying is that teardrops fall down.

The teardrops are liquid and heavy.

Eyes which are part of a person can also cry.

Crying implies that the person who is crying is either happy or sad.

Being happy implies that the person is not sad.

At the top of Figure 5.9 is a representation of part of the source domain sky. This part has to do with raining. This graph represents the following information:

The result of the sky raining is that rain drops fall down.

The raindrops are liquid and heavy.

Raining implies that the sky is cloudy.

Raining implies that it is daytime and the sky is gray or it is nighttime and the sky is dark.

Figure 5.10 shows the same target crying domain at the bottom and another part of the sky domain, snowing, as the top source analogue. The snowing analogue is the same as the raining analogue with the exception that snowflakes that fall down are solid and light instead of liquid and heavy as are raindrops. The following paragraphs discuss two different aspects of the tool's output given these graphs as input.

Figure 5.11 shows the output from Figure 5.9 which compares the crying and raining analogues on the right and from Figure 5.10 which compares the crying and snowing analogues on the left. Comparing these two outputs we see that both match Person to Sky, but the target analogue about crying matches better to raining than it does to snowing, i.e. crying is more like the raining part of the sky domain than it is the snowing part of the sky domain. Although the extent measures are the same between the two, the semantic distance measure is lower for the raining analogue than for the snowing analogue. This is because raindrops are liquid and heavy like teardrops, whereas snowflakes are solid and light. It is interesting to note that if Gentner were to use her *systematicity principle* on this example, she would ignore these two attributes and thus miss the closer analogy.

The Analogical Tool in fact derives six minimal common generalizations for this example. All six are not shown here because they look exactly the same as the output on the left in Figure 5.11. There are six of them because there is one proposition node in the

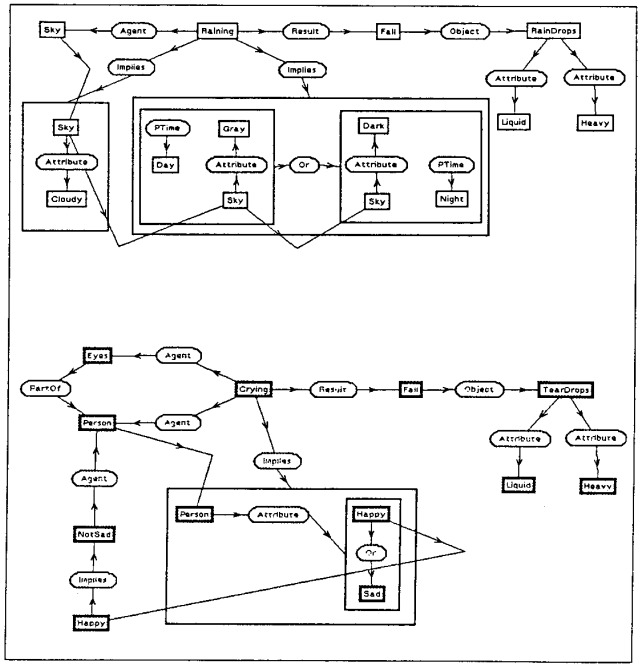


Figure 5.9 Input Analogues to the Indurkhya Example

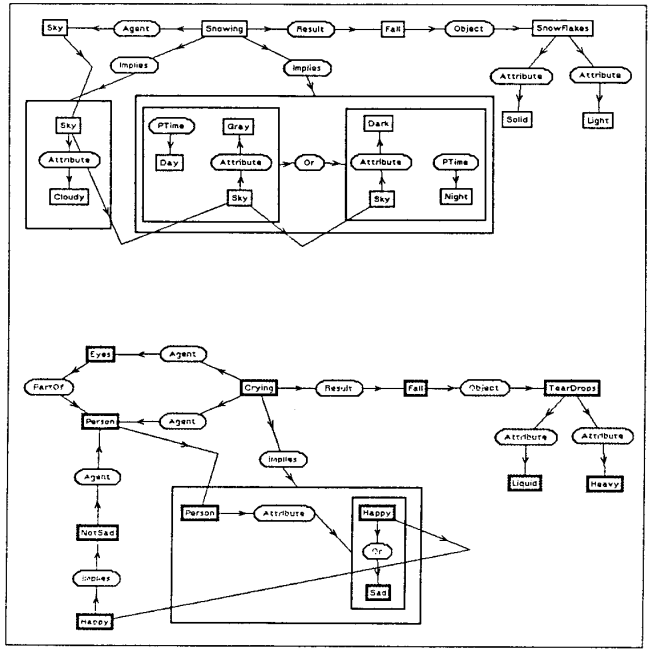


Figure 5.10 Input Analogues to the Indurkhya Example

ANALOGY CONCEPT CORRESPONDENCES

Proposition --> Proposition	Person --> Sky
Person --> Sky	Heavy --> Light
Heavy --> Heavy	Proposition --> Proposition
Crying --> Raining	Crying --> Snowing
Fall --> Fall	TearDrops --> SnowFlakes
TearDrops --> RainDrops	Fall --> Fall
Liquid --> Liquid	Person --> Sky
Person --> Sky	Liquid --> Solid
EXTENT MEASURE -11	EXTENT MEASURE -11
SEMANTIC DISTANCE MEASURE - 9	SEMANTIC DISTANCE MEASURE - 11

Figure 5.11 Output from the Indurkhya Example

target crying analogue which can match to two proposition nodes in the source raining analogue. For each of these proposition matches there are three [Sky] -> (Attribute) parts that the [Person] -> (Attribute) part can match to while still maintaining the proposition formation evaluation criterion. This results in six (2*3) equal minimal common generalizations. Another six common generalizations are formed which represent the same matches but with a semantic distance that is higher because the liquid attribute of teardrops matches to the heavy attribute of raindrops and vice versa. These are also not shown because they are not minimal common generalizations.

The last paragraphs showed that the minimal common generalizations formed for the "stronger" analogies matched person to sky. Because metaphors usually ask us to look deeper into the meaning of the metaphor, we also look at non-minimal common generalizations for other possible interpretations of the metaphor. Doing this we see in Figure 5.12 that the next most minimal generalizations formed for the crying-raining analogy are very similar except that they match eyes to sky this time. Now we see eyes crying being like the sky raining instead of a person crying being like the sky raining. This gives us a slightly different view of the metaphor - but one which is still interesting. The reason that the person-sky match shown in Figure 5.11 is "stronger", is because of the co-referent link from person as the agent of crying to person which has an attribute of being happy or sad. This co-referent link allows for matches of both of these person concepts to sky concepts in the raining analogue. In contrast, there is only one eyes concept, thus limiting the number of possible matches which can take place.

5.4.2 Comparing Indurkhya's AST Theory With the Analogical Tool

The first comparison of the Analogical Tool to Indurkhya's AST theory is that the two come up with different "stronger" analogies. The Analogical Tool found the "stronger" analogy to be the minimal common generalizations which match person to sky, and found slightly weaker but still interesting analogies which matched eyes to sky. Indurkhya's AST theory finds only the weaker analogy which matches eyes to sky; he does not form the analogy which matches person to sky. The reason is that Indurkhya needs to use template matching of structural constraints to help him decide which tokens to match. In this example the only template he can use from the crying domain is the Schema Language structural constraint: cry(eyes) -> fall-down(tear-drops, eyes). This template matches to the similar constraint in the sky domain: raining(sky) -> fall-down(rain-drops, sky). This

```
ANALOGY CONCEPT CORRESPONDENCES
Heavy --> Heavy
Eyes --> Sky
Crying --> Raining
Fall --> Fall
TearDrops --> RainDrops
Liquid --> Liquid
EXTENT MEASURE -- 13
SEMANTIC DISTANCE MEASURE -- 4
```

Figure 5.12 Output from the Indurkhya Example

causes the following tokens to be matched together: (crying -> raining) (eyes -> sky) and (tear-drops -> rain-drops). Because there is no template which allows for person to be matched to sky, this analogy is never formed. This seems to be a limitation of Indurkhya's theory because a match of person to sky actually does seem to be the stronger correspondence to form. It is stronger because there is much connected information about person such as person having the attribute of being happy or sad which can be inferred from this match. This connected information does not even get inferred by Indurkhya as existing in the map which puts eyes into correspondence with sky; but it does exist as part of the minimal common generalization of the Analogical Tool.

Indurkhya goes on to do further inferencing from the basis AT-MAP which puts eyes and sky in correspondence. From Figure 5.9 and from Indurkhya's structural constraints there is the statement about eyes being part-of a person. Indurkhya uses Positing structure and this statement to hypothesize the existence of this same relationship in the sky domain. In doing this, he hypothesizes the existence of a concept he calls sky-person for the sky domain. Following this, the part-of relationship is hypothesized; sky is now part-of sky-person. So we have in effect the statement "the sky is crying" to mean that the sky is raining just like eyes crying and that the sky raining is part-of a bigger system, sky-person (which might mean god possibly) which can also rain.

Inferencing based on the Analogical Tool can do similar hypothesizing from its weaker common generalization which matches eye to sky. We would first hypothesize a part-of relation attached to sky in the target analogue based on the source. Following this, we would hypothesize a concept in the target analogue which corresponds to the person concept in the source domain of crying.

Performing inferencing from the "stronger" analogy that matches person to sky, we would hypothesizes the missing part-of relationship. This time we are hypothesizing something which is a part-of the sky (say possibly clouds) which can also rain. This metaphor is now interpreted as the sky is raining just like a person is crying and that there is a part-of the sky, say clouds which can also rain, just like eyes which are part-of a person. Again, Indurkhya's program does not form this extended analogy because it never forms the person-sky correspondence.

Another difference between the two systems is in their various stages. The Analogical Tool first derives a set of minimal common generalizations using formation and ordering evaluations. Following this, if analogical inferencing is possible hypothesis of new information is constrained by the evaluation criteria. Indurkhya's AST theory first forms a minimal AT-MAP and then extends it. This extension is done by making further incremental coherent extensions either by more matching of tokens or using one of the two operators; Augmentation or Positing structure. Indurkhya offers little advice about what order these various operations should be carried out in.

A further difference between the systems is that the Analogical Tool allows for matches which put the attributes Liquid and Heavy into correspondence with Heavy and Liquid

respectively. This is possible because of the type hierarchy and the comparable concepts evaluation criterion. Indurkhya's AST theory only allows for Liquid to match to Liquid and Heavy to match to Heavy even though all four are attributes of rain-drops and tear-drops. The reason for this is the Schema Language representation used which is very close to predicate calculus.

All of the above differences between the two systems show the Analogical Tool to be more general than Indurkhya's AST theory by allowing more metaphor interpretations and by allowing matches using the type hierarchy to take place. It should be noted that Indurkhya could include the same kind of information contained in the type hierarchy in his structural constraints and achieve the same results. The main comment about Indurkhya's theory is that in this example it forms only the weaker analogy and this seems to be a limitation.

5.5 Summary

This chapter has shown the Analogical Tool working on four examples of analogy taken from significant researchers in the field. The purpose of the chapter has been to show how the Analogical Tool compares to work done by other researchers. Each example displayed a different aspect of the Analogical Tool and different forms of analogy formation and analogical inferencing.

The first simple example from the CYC project [Lenat et al, 1986] showed what the input and output of the Analogical Tool look like and how the evaluation criteria work. This example was shown to form the expected "Learning as Growing" analogy and is significant because it comes from an important project going on in Artificial Intelligence today, the CYC project at MCC [Lenat et al, 1986].

The second example from Gick and Holyoak [1983] showed the tool in use on a more complex example where it formed the expected analogy. The ability of the Analogical Tool to form general problem solving schemas was also shown in this example. This ability is a direct result of using the type hierarchy when forming minimal common generalizations. This example also showed how analogical inferencing can proceed based on the Analogical Tool.

The third example from Gentner [1986] showed a real life scientific use of analogy, and also presented a more complex representation. Performance of the tool in this example shows human analogical reasoning being emulated. It also differed from the others by requiring the use of nested propositions.

The last example from Indurkhya [1985] showed the communication aspect of metaphor which asks us to explore the many possible meanings. For metaphor, it is not only the minimal common generalizations we seek. We may want to also consider non-minimal common generalizations which allow us to understand the metaphor more fully.

These four examples have shown the Analogical Tool to perform well when compared with work from prominent researchers in the field. The examples define many of the different uses of analogy given in §1.1.2. In each case the Analogical tool has been shown to be effective. Further investigation of the capabilities of the tool show limitations in the area of relevancy issues. These limitations result because the tool is a similarity based system. Chapter 7 presents two examples which show these limitations and provide solutions.

Chapter 6
Relevancy Limitations and Solutions

This chapter shows that limitations of similarity based systems such as the Analogical Tool are mainly issues of relevancy. Two examples, one from Gentner and one from Russell exemplify these limitations and give solutions. These examples show where analogy as similarity pattern matching can be problematic. Drawing inferences from minimal analogies is an inductive process and thus potentially false. Both examples show that in some cases, "stronger" analogies cannot be detected without relevancy information. Gentner uses the *systematicity principle* to deal with relevancy, while inferencing based on the Analogical Tool uses the *extensibility principle*. Further relevancy issues in the Russell example require use of *determination rules*.

6.1 The Heat/Water Analogy Revisited - Dedre Gentner

The first example in this chapter comes from Gentner's paper entitled "Mechanisms of Analogical Learning" [1988]. In this paper Gentner describes the structure-mapping theory of analogical mapping and offers psychological evidence for it. This example is significant because it contains a pitfall that many overly simplistic analogy systems fall into. The problem arises when two analogies of equal strength are formed. For Gentner, choice of the more relevant one results from use of the *systematicity principle*. This principle can be seen as a structural relevancy criterion that prefers maintaining relational chains. The more relevant analogy is the one that hypothesizes the longest relational chains. Inferencing based on the Analogical Tool, chooses the more relevant analogy using the *extensibility principle* discussed in §4.3.2.

Gentner describes her structure mapping theory as follows [1987, pp.3-4]:

> "an analogy is a mapping of knowledge from one domain (the base) into another (the target) which conveys that a system of relations that holds among the base objects also holds among the target objects. Thus an analogy is a way of focusing on relational commonalties independently of the objects in which those relations are embedded. In interpreting an analogy, people seek to put the objects of the base in 1-1 correspondence with the objects in the target so as to obtain the maximum structural match. Objects are placed in correspondence by virtue of their like roles in the common relational structure; there does not need to be any resemblance between the target objects and their corresponding base objects. Central to the mapping process is the principle of *systematicity*: people prefer to map connected systems of relations governed by higher-order relations with inferential import, rather than isolated predicates. Analogical mapping is in general a combination of matching existing predicate structures and importing new predicates."

To illustrate the structure mapping rules, Figure 6.1 shows the heat/water analogy input to Gentner's analogy system. Here, the learner has a good representation of the water domain (source) and only a partial representation of the heat domain (target). Gentner's system forms two equal analogies, one with temperature matched to pressure and the other with temperature matched to diameter. Systematicity is used to choose the more relevant match of temperature to pressure because the connected Cause relation which takes Greater and Flow as arguments can be hypothesized. The match of temperature to diameter leaves nothing to be hypothesized and is the less relevant analogy.

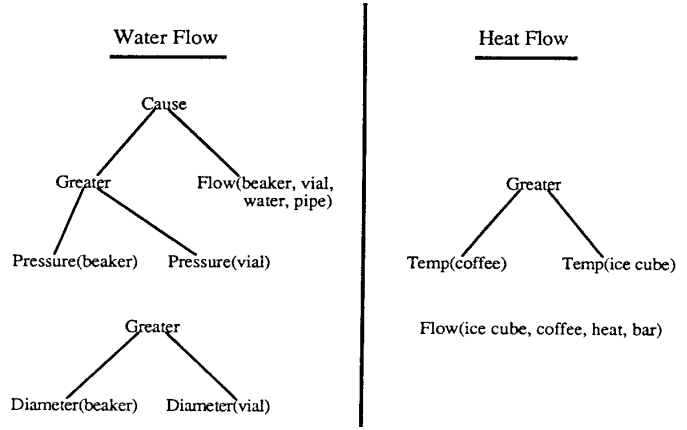


Figure 6.1 The Heat/Water Analogy from Gentner

6.1.1 Analogy Formation

Figure 6.2 shows the representations of both the water flow and heat flow situations used as input to the Analogical Tool. The representation in this example is less detailed than in heat/water analogy presented in §5.3. At the top of the figure is a representation a person might have of the 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 results in a flow of water through the pipe from the beaker to the vial. The person might 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 6.2 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 flows from the coffee to the ice-cube through a bar that connects the two. What the person does not know is that the heat flow is a result of the difference in heat between the ice-cube and the coffee; that is they do not know the causal relationship.

Figure 6.3 shows the set of minimal common generalizations formed from these two analogues. Here we see there are two members in this set, both with the same extent and semantic distance measures. All four formation evaluation criteria are used to form these generalizations. This example is different from previous examples in chapter 5 first, because it shows the forming of two equal analogies and second, because not all of the target analogue (the heat flow analogue) has matched with the source analogue. There are six arcs and four nodes from the target analogue which do not match with the source analogue. Neither of the Attribute relations in the target analogue nor the arcs connected to them match to the source analogue. The chain about the coffee being in the cup does not match either. The only difference between the two members of the set of minimal common generalizations is the concepts put into correspondence with Temperature. In one analogy the Temperature concepts are matched with the Pressure concepts. In the other equal analogy, the Temperature concepts are matched with the Diameter concepts. In order to decide which of these analogies is the more relevant, the *extensibility principle* is used.

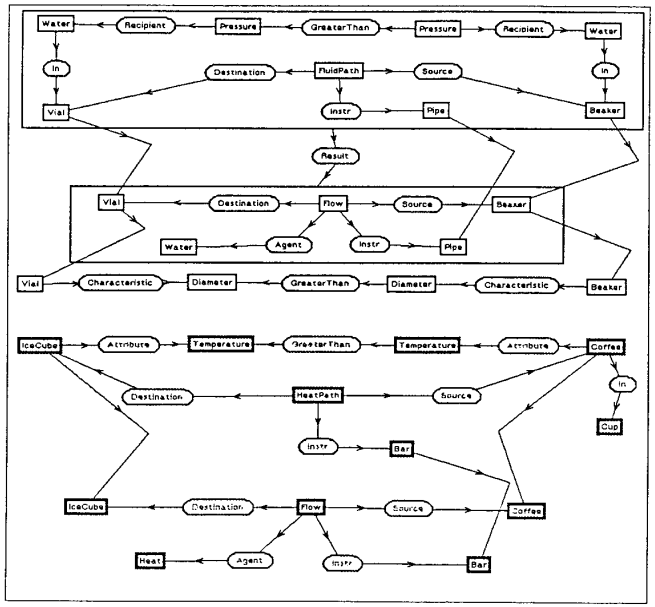


Figure 6.2 Input Analogues for the Gentner Example.

ANALOGY CONCEPT CORRESPONDENCES

IceCube --> Vial	Bar --> Pipe
Coffee --> Beaker	Coffee --> Beaker
Heat --> Water	Heat --> Water
Temperature --> Pressure	HeatPath --> FluidPath
Flow --> Flow	Temperature --> Diameter
Temperature --> Pressure	Flow --> Flow
IceCube --> Vial	Temperature --> Diameter
Coffee --> Beaker	Icecube --> Vial
Bar --> Pipe	Coffee --> Beaker
Bar --> Pipe	Bar --> Pipe
HeatPath --> FluidPath	IceCube --> Vial
EXTENT MEASURE - 6	EXTENT MEASURE - 6
SEMANTIC DISTANCE MEASURE - 18	SEMANTIC DISTANCE MEASURE - 18

Figure 6.3 Output from the Gentner Example

6.1.2 Analogical Inferencing

In §4.3.2 the *extensibility principle* was described as follows: when a choice must be made between two minimal common generalizations as to what inferencing is more plausible, the *extensibility principle* suggests a resolution. If for instance one of the generalizations is not extensible at all and the other is, then 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 will be the new minimal common generalization of the extended structure. Use of this *extensibility principle* correctly picks the analogy which puts the Temperature concepts in correspondence with the Pressure concepts.

If we examine the two analogies for what hypotheses can be made from them we see that the analogy which puts Temperature in correspondence with Pressure allows for more inferencing. In this analogy we can essentially overlay the target analogue on top of the source analogue. This leaves the bottom line of the source graph which talks about Diameters unmatched (the unmatched parts mentioned above are still unmatched here also). Now in order of possible hypotheses we are first allowed to hypothesize one of the

proposition nodes from the source analogue to the target analogue. Following this we can then hypothesize the other proposition node from the source to the target. With these two proposition nodes hypothesized we can hypothesize the Result relation between them. We now have a causal structure in the target analogue based on information from the source analogue. All of these hypotheses satisfy the evaluation criteria so we can form a new minimal common generalization which contains generalizations of this newly hypothesized structure.

In contrast, the analogy that matches the Temperature concepts with Diameter concepts only allows for hypothesis of the proposition node that surrounds the flow part of the source analogue. The proposition in the source analogue which surrounds the information about the Pressure being GreaterThan etc. cannot be hypothesized. The reason is, this hypothesis does not conform to the proposition evaluation. If we were to hypothesize this proposition node, we would have parts of the graphs contained within it matching with parts in the corresponding source proposition node and other parts contained within it in correspondence with parts of the source analogue which are not inside of the corresponding source proposition node. This second analogy is less extensible and is thus not preferred. From this example we see that one analogy is preferred to the other because it allows us to do more inferencing and this fact is detected using the *extensibility principle*.

6.1.3 Comparison of Gentner's Theory With the Analogical Tool.

The driving force behind Gentner's structure-mapping theory is use of the *systematicity principle*. This principle can be seen as a structural relevancy criteria where the more relevant structure maintains relational chains. In contrast, inferencing based on the Analogical Tool, uses a more general approach which results in the emergence of many evaluation criteria that work together to choose the more relevant analogy. These criteria are semantic as well as structural. Thus, rather than using an explicit driving principle, the Analogical Tool and analogical inferencing based on it derives analogies based on emergent properties. In addition, the Analogical Tool does not ignore attributes which can often add extra information, as the structure-mapping theory does. This ignoring of attributes is limiting as noted by Indurkhya [1985] when he says "though we agree in spirit with Gentner's theory we believe that by requiring that the relations be preserved only a certain class of analogies are captured." Indurkhya [1985] says further, objects, attributes and relations can all participate in the mapping process. The Analogical Tool, and the

inferencing based on it, like Indurkhya's AST theory are more general than the structure-mapping theory and rely on emergent evaluations to produce relevant analogies.

6.2 Determinations - Stuart Russell

This example comes from Stuart Russell's PhD thesis [1987] and from a paper by Russell and Todd Davies [1986]. It shows where analogy as similarity pattern matching can be problematic. The Analogical Tool can be described as a similarity based system which derives minimal common generalizations. As shown by this example, inferencing from minimal analogies is an inductive process and thus potentially false. Analogical inferencing based on correspondence alone can be misleading if the major correspondences are not relevant to the objective of the inferencing. The previous example showed some relevancy issues can be dealt with using criteria such as the *systematicity principle* or the *extensibility principle*. Because both Gentner's system and the Analogical Tool are similarity based, there are other relevancy issues that are problematic for them and these can be seen as limitations.

Russell's thesis defines one method of dealing with this problem using *determinations*. As Russell states [1986, pp.2]:

> "Our project is to provide an underlying justification for the plausibility of analogy from a logical perspective, and in so doing to provide a way to specify background knowledge that is sufficient for drawing reliable analogical inferences. The approach is intended to complement, rather than to compete with, more heuristic methods."

Determination rules are similar to dependency relations in database theories and are of the form: P determines Q. These rules are weaker than implication but serve as background knowledge which when added to the premises of an analogy, makes the conclusion follow soundly.

The following simple example illustrates problems similarity based systems have due to their inductive nature and shows two ways Russell's *determinations* can be incorporated into the tool to overcome them.

6.2.1 Problems in the Correspondence Base of the Analogical Tool

Figure 6.4 shows the example taken from Russell. Here, we have the target analogue represented by the dashed conceptual graph at the bottom of the figure. This graph represents:

Jack's Nationality is UK.

Jack's Native Language is unknown. (we are trying to find it)

Jack is male.

Jack is six feet tall.

The two conceptual graphs at the top of the figure represent similar information we have about Jill from the UK and Guiseppe from Italy. This example is posed as an analogical query; we are trying to find out what Jack's NativeLanguage is, based on his similarity to someone else in the database. There are no propositions in this example so it is very simple for the Analogical Tool and only requires use of three evaluations when forming the minimal common generalization. The output in Figure 6.5 shows that there are four generalizations formed with an extend of 0 (total match of concepts). In this example the semantic distance measure is necessary to give the single minimal common generalization. This generalization finds Jack to be most similar to Guiseppe and thus matches Jack's Unknown concept for NativeLanguage with Guiseppe's NativeLanguage of Italian. Jack is most similar to Guiseppe because he is a male and six feet tall. This more similar analogy answers the query of what is Jack's language, with Italian. In fact, this is the incorrect answer. What we really wanted to happen was for the system to know that one's Nationality actually determines one's NativeLanguage. From this we arrive at the correct answer that Jack's NativeLanguage is English because Jill has the same Nationality as Jack and her NativeLanguage is English. It should be noted that for this example the Analogical Tool does pick the correspondence of Jack to Jill as the next strongest analogy with a semantic distance measure difference of only one.

6.2.2 Using Russell's Determinations in the Tool

In Russell's implementation based on the logic programming system MRS [Russell, 1985], *determination rules* are added to the database, and the system uses them whenever possible

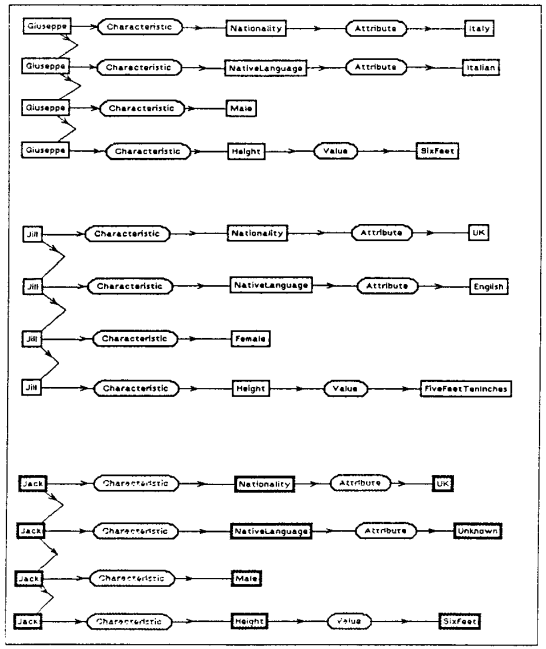


Figure 6.4 Input Analogues for the Russell Example.

ANALOGY CONCEPT CORRESPONDENCES

Nationality --> Nationality	Nationality --> Nationality
Jack --> Guiseppe	Jack --> Jill
Jack --> Guiseppe	Jack --> Jill
UK --> Italy	UK --> UK
Height --> Height	Height --> Height
Unknown --> Italian	Unknown --> English
Jack --> Guiseppe	Jack --> Jill
Jack --> Guiseppe	Jack --> Jill
SixFeet --> SixFeet	SixFeet --> FiveFeetTenInches
Male --> Male	Male --> Female
NativeLanguage --> NativeLanguage	NativeLanguage --> NativeLanguage
EXTENT MEASURE - 0	EXTENT MEASURE - 0
SEMANTIC DISTANCE MEASURE - 5	SEMANTIC DISTANCE MEASURE - 6

ANALOGY CONCEPT CORRESPONDENCES

Nationality --> NativeLanguage	Nationality --> NativeLanguage
Jack --> Guiseppe	Jack --> Jill
Jack --> Guiseppe	Jack --> Jill
UK --> Italian	UK --> English
Height --> Height	Height --> Height
Unknown --> Italy	Unknown --> UK
Jack --> Guiseppe	Jack --> Jill
Jack --> Guiseppe	Jack --> Jill
SixFeet --> SixFeet	SixFeet --> FiveFeetTenInches
Male --> Male	Male --> Female
NativeLanguage --> Nationality	NativeLanguage --> Nationality
EXTENT MEASURE - 0	EXTENT MEASURE - 0
SEMANTIC DISTANCE MEASURE - 8	SEMANTIC DISTANCE MEASURE - 10

Figure 6.5 Output from the Russell Example

to perform analogical inferencing. In the above example the rule Nationality(x,n) determines NativeLanguage(x,l) is added to the database which causes this rule to be used first. Using this rule in the above example, Russell's system completely ignores

information about Guiseppe and looks at only the relevant information about Jill; because Jill has the same Nationality as Jack, his NativeLanguage must be the same as Jill's. The *determination rules* not only give the correct answer, they also cut down on the amount of searching and matching which must be done. If the system has these kind of rules it can ignore irrelevant information and use only information relevant to the problem. The Analogical Tool can incorporate these *determination rules* in two ways.

The first way the determination rules can be incorporated is by using an aspect of the conceptual graph model called *focus of assertion*. Implementation of determination rules in this manner is shown in Figure 6.6. Here we see that the second line of each of our original graphs has had the determination added to it. For each of Jack, Jill and Guiseppe we now state that [Nationality!] -> (Determines) -> [NativeLanguage!]. The exclamation marks show the focus of assertion and specifies that within this graph this is the most important information; the information to consider first when forming correspondences. In this way when Jack matches against Jill and Guiseppe, only this information is used, and the correct answer is given; Jack's NativeLanguage is English because this more important part of the graph matches exactly to Jill's and she has a NativeLanguage of English. Although the determination rule has been added to each of the three graphs in Figure 6.6 it need not be. It is possible to have the *determination rules* as general background knowledge in the same way as Russell. When analogies are being formed, these general determination rules can be joined into the graphs using the conceptual graph join operation described in §3.3.1. In this way the graphs remain small and the *determination rules* can be implemented as general background information and can be acquired separately.

The second way in which Russell's *determination rules* can be incorporated into the Analogical Tool is closer to the general framework we have at present, and uses the type hierarchy effectively. First, for this method we need to add the referent field to concepts. In this implementation of the Analogical Tool concepts do not contain a referent field, for simplicity. In this example, [Nationality] -> (Attribute) -> [Italy] would be replaced by the more concise representation: [Nationality : Italy] and [NativeLanguage] -> (Attribute) -> [Italian] would be replaced by [NativeLanguage: Italian]. Second, we notice where in our type hierarchy these concepts appear. Figure 6.7 shows part of the type hierarchy in which many of the concepts from this example appear. Both Nationality and NativeLanguage are

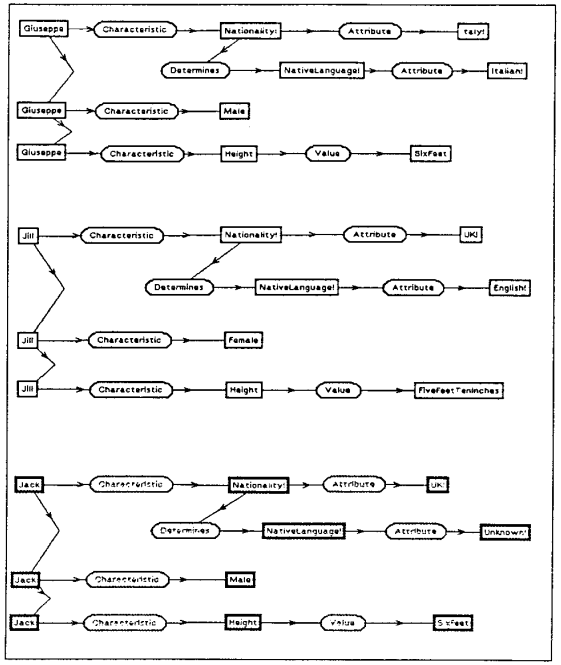


Figure 6.6 Determinations as Focus of Assertion in the Russell Example.

cultural characteristics, whereas concepts like Sex, Height and hair-color are physical characteristics. Our type hierarchy thus includes much of the same information contained in Russell's *determination rules*. He says that Nationality determines NativeLanguage and so we should ignore characteristics like Sex or Height when forming the analogy.This corresponds with our semantic distance measure over the type hierarchy. Here we see that Nationality is closer to NativeLanguage in our hierarchy than sex or height. What we need to do when forming correspondences is to prefer, and look for, correspondences which are semantically closer. Now with our new representation in place, when we look for Jack's NativeLanguage we notice that of the other matches we make, Nationality is much closer in the hierarchy than sex or height and we determine that an exact match with Nationality to Jill is more relevant information than an unequal match to Guiseppe. We then match the Unknown referent of NativeLanguage for Jack with the referent field of NativeLanguage for Jill which says that Jack's NativeLanguage is English just like Jill's.

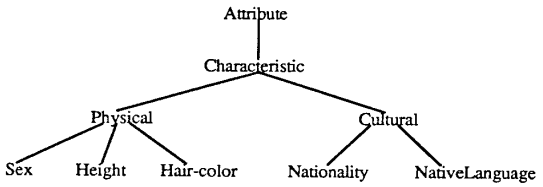


Figure 6.7 A Portion of the Type Hierarchy

This explanation is not just a single incident which happens to work for one example. In looking at other examples given by Russell stating other uses for *determination rules*, we find that the same kind of information is present in the type hierarchy. All that needs to be done is to make use of this information. It is really no surprise that this kind of knowledge is contained in the type hierarchy because it is just a modular way of representing background knowledge, i.e. knowledge about generalizations and specializations. It appears that much of this generalization/specialization information is similar to Russell's *determinations*. This has not been completely studied or tested but it does appear to be a fruitful question to pursue.

This section has presented a problematic example to show limitations of similarity based systems, such as the Analogical Tool. Although this is a problematic case, there are at least two ways to implement Russell's *determination rules* to deal with them. One extension expands the tool by using more of the conceptual graph model and the other attempts to stay within the bounds of the tool and use information already available.

6.3 Summary

This chapter has described limitations of similarity based systems as being mainly issues of relevancy. Two examples, one from Gentner and one from Russell exemplified these limitations and gave solutions. These examples showed where analogy as similarity pattern matching can be problematic. Drawing inferences from minimal analogies is an inductive process and thus potentially false. Both examples showed that in some cases, "stronger" analogies cannot be detected without relevancy information. Gentner uses the *systematicity principle* to deal with relevancy, while inferencing based on the Analogical Tool uses the *extensibility principle*. Further relevancy issues in the Russell example required use of *determination rules* and these rules can be implemented in the Analogical Tool in two ways.

Chapter 7
Conclusions and Further Work

This chapter presents a summary of the major points and accomplishments of the thesis. They are presented briefly in the summary and achievements section and expanded upon in the discussion sections. The final portion of the chapter presents further enhancements to the Analogical Tool.

7.1 Summary and Achievements

The overall aim of this research has been to investigate the fundamental properties of computer based analogy and to use these properties as a basis for analogy formation and analogical inferencing. The specific objectives of the thesis were:

1) To analyze the processes of analogy as studied in human psychology.

2) To analyze mathematical theories of analogy.

3) To analyze the constraints on analogy formation to determine how plausible analogies are formed and ordered.

4) To formulate a general framework for analogy which incorporates results from the above analyses.

5) To design and implement an Analogical Tool for analogy formation based on the general framework for analogy.

6) To apply the Analogical Tool to published examples of analogy to demonstrate the abilities of the tool.

7) To investigate the extension of the Analogical Tool into the area of analogical inferencing based on results from the tool's performance in analogy formation.

8) To determine limitations of the Analogical Tool.

Objectives 1 through 3 above have been satisfied by analyzing works in the areas of Psychological models, Mathematical theories and Evaluation criteria for analogy. Objective

4 has been met by description of a general framework for analogy which embodies critical functions realized from the analysis. The design of an Analogical Tool as an instantiation of the general framework using conceptual graphs for the knowledge representation scheme has met the fifth objective. Objective 6 has been satisfied by successfully testing the Analogical Tool on critical examples of analogy from the literature and comparing results to the original researchers. Using these results, it has been shown how the Analogical Tool supports analogical inferencing. Not only has this inferencing been described, but has been shown to also fit within the general framework. In satisfying the final objective, relevancy limitations to the tool were found. These limitations were shown to be characteristic of similarity based systems and to require use of the *extensibility principle* and Russell's *determination rules* [1987]. Figure 7.1 gives an overview of the thesis, showing the various stages of meeting the objectives and pointing the way to further enhancements of the system.

7.2 Discussion: The General Framework and the Analogical Tool

This section discusses the significance of the thesis by expanding on the points presented above. The discussion here is restricted to the general framework and the Analogical Tool. These represent the first two boxes in the overview of the thesis shown in Figure 7.1.

A major thrust of the thesis has been to base formulation of the Analogical Tool on an analysis of work done by others in the field of analogy. For this reason work of other researchers are categorized and analyzed for their similarities. From these similarities, critical functions of analogy are realized. Most of the works analyzed are recent and deal with attempts to formulate computational emulations of analogy. These are chosen mainly because the Analogical Tool is also a computational emulation of analogy.

Work on analogy and induction in the area of mathematics by Polya [1954] leads him to describe analogy as a common generalization. Clarification of this description leads to a general framework which supports Polya and embodies realized critical functions. The framework presents a model of analogy which just describes what analogy is. This framework is characterized by three important elements; "part-of", "generalization" and a "generalization ordering" relation. The general framework, formally defined in §2.3, requires few primitives and can be seen as a framework for plausible reasoning in general

as well as specifically for analogy. We can also imagine different instantiations of the framework which use different knowledge representations.

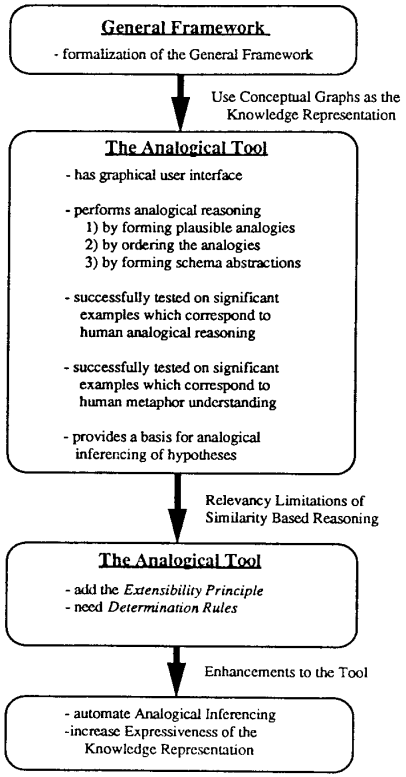


Figure 7.1 An Overview of the Thesis

Although this thesis uses conceptual graphs for the knowledge representation scheme, it is possible to use a different representation and arrive at a different Analogical Tool based on the same general framework.

Conceptual graphs were chosen as the knowledge representation scheme 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 expressive enough to be able to represent examples of analogy taken from the literature.

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

5) 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 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 results in a computational tool which can be seen to both form analogies and show how analogical inferencing proceeds. The inferencing is not an add-on explicit part of the system, rather it is an integrated extension of the existing framework.

The four formation evaluations of the Analogical Tool define what parts of the analogues are put into correspondence when forming common generalizations, and the two ordering evaluations are used to derive the "stronger", more plausible, minimal common generalizations.

Analogical inferencing based on the tool is seen as formation of a new minimal common generalization which is an extended specialization of an original minimal common generalization. This new specialized minimal common generalization is the result of extending one of the analogues, typically the target. Because of the original similarity between the two analogues, information in the source analogue is hypothesized as existing in the target analogue by analogical inferencing. The choice of what hypothesis to make are again controlled by the same evaluations used in analogy formation. This means that work done in the study of analogy formation in the Analogical Tool is directly applicable to analogical inferencing.

The examples of analogy on which the tool is tested are taken from prominent researchers in the field of analogy. Each of the examples is very different and shows the various uses of analogy described in §1.2.2. The power of the Analogical Tool is its ability to handle these many varied examples and also be extendable into areas which are problematic. The examples range from a simple example from CYC [Lenat et. al, 1986] to metaphor understanding. The performance abilities of the tool show it to be the kind of general non purpose-directed tool described in §1.3; that is being general enough to be used for many purposes. This tool, which results from instantiating the general framework based on past work on analogy, derives analogies using implicit evaluation criteria. The tool and the analogical inferencing based on it have been shown to be more general than systems of other researches such as Gentner [1983] and Indurkhya [1985] with no loss of functionality.

7.3 Discussion: Addressing Relevancy Limitations

The overview of the thesis displayed in Figure 7.1 shows determination of limitations of the Analogical Tool as a similarity based system, to be restricted to issues of relevancy. Two examples, one from Gentner and one from Russell exemplified these limitations and gave solutions. The Gentner example in §6.1 and the Russell example in §6.2 showed

where analogy as similarity pattern matching can be problematic. As described there, drawing inferences from minimal analogies is an inductive process and thus potentially false. Both examples show that in some cases, the "stronger" analogies cannot be detected without relevancy information. Gentner uses the *systematicity principle* to deal with the relevancy problem in §6.1 and inferencing based on the tool uses the *extensibility principle*. Further relevancy issues in the Russell example require use of *determination rules*.

The Extensibility Principle. Gentner's *systematicity principle* is a structural criterion that deals with some relevancy requirements. By requiring maintenance of relational chains across analogues, Gentner shows that for the particular example given in §6.1 the more relevant analogy is chosen. Inferencing based on the Analogical Tool uses the *extensibility principle* to choose the same, more relevant analogy. The *extensibility principle* is structural as well as semantic since it involves use of all formation evaluations. The three formation evaluations of same relations, co-reference links and propositions are structural while the comparable concepts evaluation is semantic. Thus the *extensibility principle* goes beyond the *systematicity principle* by using some semantics for dealing with relevancy issues. As shown by Russell [1987], similarity based systems such as Gentner's and the Analogical Tool have problems dealing with other relevancy issues and require use of background knowledge.

Determinations. Background knowledge in the form of determination rules deals with relevancy issues beyond the capabilities of most similarity based analogy systems. As discussed in §6.2 the addition of Russell's determinations to the Analogical Tool would give it the ability to deal with these issues. In that discussion, two ways of implementing the determination rules was presented. The first way is using focus of attention from conceptual graphs and the second, using information in the type hierarchy. Knowledge in the type hierarchy appears to be similar to Russell's determination rules and might be used instead of them. Further investigation needs to be completed to determine which implementation is the more appropriate.

7.4 Further Enhancements to the Analogical Tool

Discussion of the major achievements of the Analogical Tool show it to be an effective system for performing analogical reasoning. As shown in Figure 7.1, this reasoning

comes in many forms; formation of plausible analogies, ordering of the analogies, formation of schema abstractions, use of the extensibility principle, as well as analogical inferencing. Further enhancements of the tool are possible by automating the analogical inferencing process and by increasing the expressiveness of the knowledge representation of the tool.

Automation of Analogical Inferencing in the Tool. In §1.1 we defined analogical reasoning as being comprised of two elements. The first, analogy formation, forms and orders analogies, and specific to the Analogical Tool, forms schema abstractions and aids in metaphor understanding. The second, analogical inferencing, generates hypotheses in a target analogue based on a source analogue. The tool supports this analogical inferencing and automation of the process would enhance the Analogical tool.

Inferencing would proceed as discussed in §4.3. For each minimal common generalization formed, a list of concepts and relations not yet in correspondence would be produced. The system would then proceed through the following questions until no more hypothesizing is possible for each minimal common generalization:

- can I hypothesize a co-reference link?
- if no then can I hypothesize a connected relation?
- if no then can I hypothesize a proposition box?
- if no then can I hypothesize a concept inside of a proposition box?
- if no then can I hypothesize a connected concept?
- if no then can I hypothesize a concept with a connected relation?
- if no then can I hypothesize an unconnected concept?

For each time through this loop that a hypothesis is made the system would check the formation evaluations. If the evaluations evaluate to true the minimal common generalization would be updated. When all hypotheses for all minimal common generalizations are made the *extensibility principle* would be used to report the analogical inferencing performed; that is the most extensible analogy would be reported. For the first implementation of analogical inferencing in the tool, the newly hypothesized target labels would take on the labels of their corresponding source concepts.

Enhancing the Expressiveness of the Knowledge Representation. Empirical testing of the Analogical Tool has shown the knowledge representation to be expressive enough for analogue description and to allow for powerful analogical reasoning. This expressiveness has resulted from use of only a subset of the conceptual graph model. It is feasible to consider increasing the expressiveness of the knowledge representation and possibly the power of the tool by adding more elements of the conceptual graph model.

The first addition is adding the referent field to all concepts. This will provide a more concise representation. At present all concepts are generic, but with this addition we can represent for example [Person:Sam]. This addition is also necessary for implementation of Russell's *determination rules*.

The next extension to be made to the tool is attaching type definitions, schemata, prototypes and canonical graphs to the type hierarchy. These additions would provide background knowledge and add more meaning to the graphs. For instance, prototypes provide default knowledge which could be useful in determining concept labels in analogical inferencing. Canonical graphs and schemata provide expectations by specifying the kinds of relations and concepts expected to be attached to certain other concepts. As stated earlier, addition of these structures to the Analogical Tool would be done in an object oriented manner because each of the concepts in the type hierarchy is a Smalltalk class. This means that each concept has its own variables and methods and thus its own structure and state.

Another conceptual graph extension could be addition of a hierarchy of relation types. This hierarchy would be implemented in the same fashion as the concept type hierarchy. This addition would allow the kind of creative slippages described by Hofstadter and Mitchell [1988] to occur.

Each of the extensions mentioned here can be easily implemented but decisions are still necessary to determine just how each of them would be used, and when. There is the possible disadvantage of an increase in complexity associated with each of these extensions that must be weighed against added expressiveness.

Further Expressiveness through Contraction and Expansion of Graphs. Expansion of conceptual graphs refers to replacing a concept in a conceptual graph with its type definition if the concept is not primitive. Contraction of conceptual graphs refers to

finding a subgraph of a graph which exactly matches a type definition in the type hierarchy. Once this subgraph is found, it can be replaced by the corresponding concept label from the type hierarchy. In this way conceptual graphs can be expanded and contracted and reasoning can occur at different levels of detail. This expansion and contraction of graphs could be used in a way similar to that of the CYC project team [Lenat et al, 1986]. In CYC, frames are matched and slots put into correspondence. Frames corresponding to these slots can then be retrieved and further matches found at a more detailed level. This same kind of process can happen in the Analogical Tool if concepts in the type hierarchy have type definitions associated with them. Further analogies can be found at a more detailed level by retrieving the type definitions of corresponding concepts between the analogues. It is also conceivable to take each of the analogues and expand one or several concepts within them before forming the analogies. This would allow formation of analogies at various levels of detail.

This thesis has described the design, implementation, testing, critical evaluation and enhancement of an Analogical Tool. This tool is based on a principled approach that uses work of past researchers in analogy. Design of the tool has helped bring about formulation of a general framework for analogy and a clarification of the terms *analogy* and *analogical reasoning*.

A general framework has been defined in terms of a limited number of very general inferencing procedures and yet provides sufficient constraints for plausible hypotheses which are independent of the knowledge representation used. Use of conceptual graphs for the framework has provided a logically based yet semantically rich representation that allows for human analogical reasoning to be emulated. The truth of this reasoning depends as it should, on the correspondence between the knowledge representation used and the world it represents.

References

[Adams, 1987]
Sam S. Adams. NodeGraph 80. Knowledge Systems Corporation. 1987.

[Bobrow and Winograd, 1977]
D. Bobrow and T. Winograd. An Overview of KRL, A Knowledge Representation Language. Cognitive Science 1 (1).

[Burstein and Adelson, 1987]
Mark Burstein and Beth Adelson. Analogical Learning: Mapping and Integrating Partial Mental Models. Proceedings of the Cognitive Science Society Conference. pp. 11-22. 1987.

[Carbonell, 1986]
Jaime Carbonell. Derivational Analogy. Chapter 14. Machine Learning Vol. II. Ryszard Michalski, Jaime Carbonell and Tom Mitchell, Editors. Morgan Kaufmann Publishers, Inc. 1986.

[Carnot, 1977]
S. Carnot. Reflections on the Motive Power of Fire. Gloucester, Ma: Peter Smith. (Original work published 1824).

[Evans, 1968]
Thomas Evans. A Heuristic Program to Solve Geometric Analogy Problems. In Semantic Information Processing, Edited by Marvin Minsky. MIT Press. Cambridge, Ma. 1968.

[Fargues et. al, 1986]
Jean Fargues, Marie-Claude Landau, Anne Dugourd, Laurent Catach. Conceptual Graphs for Semantics and Knowledge Processing. IBM Journal of Research and Development. Vol. 30. No. 1. 1986.

[Fillmore, 1968]
C.J. Fillmore. The Case for Case, in Universals in Linguistic Theory, Edited by E. Bach and R. Harms. Harms, Holt, Rinehart and Winston. New York. 1968.

[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. pp. 155-170. 1983.

[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.

[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. pp. 1-38. 1983.

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

[Greiner, 1985]
Russell Greiner. Learning by Understanding Analogies. PhD Thesis. Stanford Unviersity. 1985.

[Greiner and Lenat, 1980]
R. Greiner and D. Lenat. RLL: A Representation Language Language. AAAI-1. pp.165-169. 1980.

[Haraguchi, 1986]
Makoto Haraguchi. Analogical Reasoning Using Transformations of Rules. Bulletin of Informatics and Cybernetics. Vol. 22. No. 1-2. 1986.

[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.

[Indurkhya, 1988]
Bipin Indurkhya. Constrained Semantic Transference: A Formal Theory of Metaphors. Chapter 6. Armand Prieditis, Editor. Analogica. Morgan Kaufmann Publishers, Inc. Los Altos, California. 1988.

[Kedar-Cabelli, 1988]
Smadar Kedar-Cabelli. Toward a Computational Model of Purpose-Directed Analogy. Chapter 4. Analogica. Armand Prieditis Editor. Morgan Kaufmann Publishers. Los Altos, California. 1988.

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

[Lakoff and Johnson, 1980b]
George Lakoff and Mark Johnson. Metaphors We Live By. University of Chicago Press. Chicago. 1980.

[Leatherdale, 1974]
W. H. Leatherdale. The Role of Analogy, Model and Metaphor in Science. North-Holland Publishing Co. Amsterdam. 1974.

[Leishman, 1988a]
Debbie Leishman. An Annotated Bibliography of Works on Analogy. The International Journal of Intelligent Systems. To appear.

[Leishman, 1988b]
Debbie Leishman. Analogy as a Partial Correspondence Over Conceptual Graphs. The Proceedings of Principles of Knowledge Representation and Reasoning Conference. May 1989. To appear.

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

[Lenat and Guha, 1988]
Doug Lenat and R.V. Guha. The World According to CYC. MCC Technical Report No. ACA-AI-300-88. 1988.

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

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

[Rattermann and Gentner, 1987]
M.J. Rattermann and Dedre Gentner. Analogy and Similarity: Determinants of Accessibility and Inferential Soundness. Proceedings of the Cognitive Science Society Conference. pp. 609-620. 1987.

[Russell, 1985]
Stuart Russell. The Complete Guide to MRS. Technical Report No. STAN-CS-85-1080. Department of Computer Science. Stanford University. 1985.

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

[Russell and Davies, 1987]
Stuart Russell and Todd Davies. A Logical Approach to Reasoning by Analogy. Proceedings of AAAI-87. pp. 264-270. 1987.

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

[Sowa, 1988]
John Sowa. Conceptual Graph Notation. Proceedings Third Annual Workshop on Conceptual Graphs. Minnesota. pp. 1-22. 1988.

[Sowa and Way, 1986]
John Sowa and Eileen Way. Implementing a Semantic Interpreter using Conceptual Graphs. IBM Journal of Research Development. Vol. 30. No. 1. pp. 57-69. 1986.

[Sridharan and McCarty, 1981]
N.S. Sridharan and L.T. McCarty. The Representation of an Evolving System of Legal Concepts. IJCAI-81. Vol. I. pp. 246-253. 1981.

[Stefanescu, 1985]
Dan Catalin Stefanescu. Case Studies in Computation by Analogy. PhD Thesis. Harvard University. Cambridge. 1985.

[Sternberg, 1977]
Robert Sternberg. Intelligence, Information Processing, and Analogical Reasoning. Lawrence Erlbaum Associates. Hillsdale. 1977.

[The American Heritage Dictionary, 1983]
The American Heritage Dictionary. Dell Publishing Company. New York. 1983.

[Trenouth, 1988]
Jason Trenouth. Reasoning by Analogy in Prolog. AISB, Spring 88. No. 64. 1988.

[Way, 1987]
Eileen Way. Dynamic Type Hierarchies: An Approach to Knowledge Representation Through Metaphor. PhD Thesis. Department of Philosophy. State University of New York at Birghamton. 1987.

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