Approaches to Knowledge Representation

An Introduction

Edited by
G. A. Ringland
and D. A. Duce

KNOWLEDGE-BASED AND EXPERT SYSTEMS SERIES

Series Editor: **Alex Goodall,** *Expert Systems International Ltd., Oxford, England*

1. Approaches to Knowledge Representation: An Introduction
   Edited by **G. A. Ringland** *and* **D. A. Duce**

Approaches to Knowledge Representation: An Introduction

Edited by
G. A. Ringland
and
D. A. Duce
Rutherford Appleton Laboratory, England

RESEARCH STUDIES PRESS LTD.
Letchworth, Hertfordshire, England

JOHN WILEY & SONS INC.
New York · Chichester · Toronto · Brisbane · Singapore

RESEARCH STUDIES PRESS LTD.
58B Station Road, Letchworth, Herts. SG6 3BE, England

Marketing and Distribution:

Australia, New Zealand, South-east Asia:
Jacaranda-Wiley Ltd., Jacaranda Press
JOHN WILEY & SONS INC.
GPO Box 859, Brisbane, Queensland 4001, Australia

Canada:
JOHN WILEY & SONS CANADA LIMITED
22 Worcester Road, Rexdale, Ontario, Canada

Europe, Africa:
JOHN WILEY & SONS LIMITED
Baffins Lane, Chichester, West Sussex, England

North and South America and the rest of the world:
JOHN WILEY & SONS INC.
605 Third Avenue, New York, NY 10158, USA

Library of Congress Cataloging in Publication Data

Approaches to knowledge representation: an introduction/edited by
G. A. Ringland and D. A. Duce.
p. cm. — (Knowledge-based and expert systems series; 1)
Bibliography: p.
Includes index.
ISBN 0 471 91785 0 (Wiley)
1. Expert systems (Computer science) 2. Artificial intelligence.
I. Ringland, G. A. (Gordon A.) II. Duce, David A. III. Series.
QA76.76.E95A669 1988
006.3′3—dc19 87-32331

British Library Cataloguing in Publication Data

Approaches to knowledge representation: an
introduction.—(Knowledge-based and
expert systems series).
1. Expert systems (Computer science)
I. Ringland, G.A. II. Duce, David A.
III. Series
006.3′3 QA76.76.E95

ISBN 0 86380 064 5
ISBN 0 471 91785 0 Wiley

ISBN 0 86380 064 5 (Research Studies Press Ltd.)
ISBN 0 471 91785 0 (John Wiley & Sons Inc.)

Printed in Great Britain by SRP Ltd., Exeter

Preface

Knowledge Representation is the keystone of the Artificial Intelligence enterprise, and systems utilizing AI techniques. Any project with a knowledge based content must choose some way of representing that knowledge, yet too rarely is this choice informed or even conscious.

This book originated from a series of lectures on Knowledge Representation given by the authors at Rutherford Appleton Laboratory. The aim is to explain and analyse a wide range of approaches to Knowledge Representation to assist in the process of rational design for knowledge based systems. The book is divided into three parts.

- The first is a discussion of the standard approaches to knowledge representation: logic, semantic networks, frames and rule based systems.
- The second is a discussion of how we, as humans, appear to represent knowledge.
- Finally a selection of more advanced topics is presented - the representation of time, meta-knowledge, conceptual graphs, issues of computational tractability, and functional approaches.

The intended audience is final year undergraduates, first year graduate students and computer professionals who are beginning to work in the areas of Knowledge Engineering and Artificial Intelligence.

Acknowledgements

Special thanks are due to Bob Hopgood for reading and commenting on the first draft of this volume and to Ronald Brachman of AT&T for his very perceptive comments on parts of the first draft.

We are particularly indebted to Amanda Hopgood and Charlie Kwong for producing a large number of the diagrams, to Alex Goodall of ESI Ltd. (Series Editor) and Veronica Wallace of Research Studies Press for their patience and help during the preparation of this book.

Credits

Chapter 3. Figure 1 adopted from M.R. Quillian "Semantic Memory" in M. Minsky (Editor) *Semantic Information Processing* (1968) with permission from MIT Press. Figure 2 reproduced with permission from McGraw-Hill Book Company from P.H. Winston *The Psychology of Computer Vision* (1975). Figure 3 adapted with permission from *Computer Models of Thought and Language* by R.C. Schank and K.M. Colby, copyright 1973 W.H. Freeman and Company. Figure 7 reproduced with permission from Elsevier Science Publishers B.V. from L.K. Schubert "Extending the expressive power of semantic networks", *Artificial Intelligence* **7**(2), pp.163-198, 1976. Figure 8 is used by permission of the International Joint Conference on Artificial Intelligence, Inc. from P.J. Hayes, "On Semantic Nets, Frames and Associations", *Proc. 5th IJCAI* 1977. Copies of the Proceedings are available from Morgan Kaufmann Publishers, Inc., 95 First Street, Los Altos, CA 94022, U.S.A. Figure 9 is reproduced with permission from R.J. Brachman and J.G. Schmolze "An overview of the KL-ONE knowledge representation scheme", *Cognitive Science* **9**(2), pp.171-216, 1985.

Chapter 6. Figure 1 is reproduced with permission from the American Association for the Advancement of Science from R.N. Shepard and J. Metzler "Mental rotation of three-dimensional objects", *Science* **171**, pp.701-3, 1971. Figures 2 and 4 are reproduced with permission from Academic Press from L.A. Cooper and R.N. Shepard in *Visual Information Processing*, ed. W.G. Chase (1973) and R.N. Shepard and C. Feng "A chronometric study of mental paper folding", *Cognitive Psychology*, **3**, pp.228-243, 1972, respectively. Figure 5 is reproduced with permission from L.R. Brooks "Spatial and verbal components of the act of recall", *Canadian J. Psychology*, **22**, pp.349-368, 1968.

Chapter 7. Figures 10 and 11 are reproduced from J.F. Sowa, *Conceptual Structures: Information Processing in Mind and Machine* (1983), with permission from Addison-Wesley Publishing Co. Inc.

Chapter 10. Figure 4 is reproduced with permission of the Institute of Electrical and Electronic Engineers from J. Mylopoulos, T. Shibahara and J.K. Tsotsos, "Building Knowledge-based Systems: the PSN experience", *IEEE Computer*, **16**(10), copyright © IEEE 1983.

Gordon A. Ringland
David A. Duce

14 October 1987

Table of Contents

All authors are located at the Rutherford Appleton Laboratory, Chilton, Didcot, OXON OX11 0QX, U.K.

1 Background and Introduction

David Duce and Gordon Ringland

1.1 Background

There is a sense in which every computer program contains knowledge about the problem it is solving. A program for solving differential equations, for example, certainly contains knowledge about that particular problem domain. The knowledge is in the particular algorithms the program employs and the decision procedure which determines which algorithm to employ in a particular set of circumstances. However, it is a characteristic of most computer programs that the knowledge they contain is not represented explicitly and cannot be readily expanded or manipulated. Knowledge is in a sense projected onto the program, like a 3-Dimensional image being projected onto a 2-Dimensional surface, and cannot be reconstructed. Given a "traditional " payroll program it would be only possible to make fragmentary deductions about, say, statutory sick pay legislation, yet this is a part of the knowledge on which the program is based and which was used in the construction of the program.

This scenario is to be contrasted with the field of Artificial Intelligence (AI) where the concern is to "write down descriptions of the world in such a way that an intelligent machine can come to new conclusions about its environment by formally manipulating these descriptions" (Brachman and Levesque, 1985a). As Sloman (1979) remarks, "work in Artificial Intelligence, whether aimed at modelling human minds or designing smart machines, necessarily includes a study of knowledge. General knowledge about how knowledge is acquired, represented and used, has to be embodied in flexible systems which can be extended, or which can explain their actions. A machine which communicates effectively with a variety of humans will have to use information about what people can be expected to know in various circumstances".

Jackson (1986) in his excellent book *Introduction to Expert Systems* gives a very succinct overview of AI. He identifies three periods in the development of AI, the Classical Period, the Romantic Period and the Modern Period. He identifies the Classical Period with the game playing and theorem proving programs that were written soon after the advent of digital computers. The game playing (for example, chess) programs of this era were based on the notion of searching a state space. Problems were formulated in terms of a starting state (e.g. the initial state of a chess board), a test for detecting final states or solutions (e.g. the rules for checkmate in chess), and a set of operations that can be applied to change the current state (for example, the legal moves in chess). In any but the simplest of cases, an exhaustive search of the state spaces was infeasible and the trick then was to find some means of guiding the search. This led to the use of rules of thumb or heuristics, that could be used to guide the search in specific domains. Chess-playing programs constructed according to this paradigm cannot be said to explicitly represent the knowledge the chessmaster has about the game and the strategies he uses to reason about this knowledge.

Similar considerations apply to theorem proving systems of this era. Jackson describes the most important discoveries of this period as the twin realizations that (a) problems of whatever kind could, in principle, be reduced to search problems providing that they could be formalized in terms of a starting state, an end state and a set of operations for generating new states, but (b) that the search had to be guided by some representation of knowledge about the domain of the problem. In most cases it was felt necessary to have some explicit representation of knowledge about the objects, properties and actions associated with the domain or to have a global problem solving strategy.

The Romantic Period is identified with the research in computer understanding that went on between the mid-1960's and mid-1970's. Whatever beliefs one may hold about the possibility of a computer understanding anything, the ability to represent knowledge about real or imaginary worlds and reason using these representations is certainly a prerequisite for understanding. Much research was devoted in this period to the development of general frameworks for encoding both specific facts and general principles about the world, and although the whole enterprise turned out to be a very non-trivial exercise, many of the approaches to knowledge representation to be described in this book have their origins in this period.

The Modern Period covers the latter half of the 1970's to the present day. There has been a growing conviction that the power of a problem solver lies in the explicit representation of knowledge that the program can access, rather than in a sophisticated mechanism for drawing inferences from the knowledge. This period has seen the development of a number of expert systems which perform well on non-trivial tasks. These programs generally have two components, a *knowledge base* which contains the representation of domain specific knowledge, and an *inference engine* which performs the reasoning. Jackson observes that these systems tend to work best in areas where there is a substantial body of knowledge connecting situations to actions. Deeper representations of the domain in terms of spatial, causal or temporal models are avoided, but these are problems that a general knowledge representation system cannot side-step quite so easily.

1.2 The Knowledge Representation Problem

Brachman and Levesque in their introduction to *Readings in Knowledge Representation* (1985a) remark that the notion of knowledge representation is essentially an easy one to understand. It simply has to do with writing down, in some language or communications medium, descriptions or pictures that correspond in some salient way to the world or a state of the world. As in other areas of computer science, it is also necessary to consider the ways in which the representation is to be manipulated and the uses to which it is to be put. As remarked earlier, the primary reason for wanting to represent knowledge is so that a machine can come to new conclusions about its environment by manipulating the representation.

The first ingredient of the knowledge representation problem is to find a *knowledge representation language*, that is some formal language in which domains of knowledge can be described. Most systems of practical interest then need to be able to provide their users with access to the facts implicit in the knowledge base as well as those stored explicitly, and thus it is necessary to have a component of the knowledge representation that can perform automatic *inferences* for the user. The third component of the knowledge representation problem is how to capture the detailed knowledge base that represents the system's understanding of its domain. This latter problem is beyond the scope of this book, however.

David Israel characterized the knowledge representation problem as follows:

> All parties to the debate agree that a central goal of research is that computers must somehow come to "know" a good deal of what every human being knows about the world and about the organisms, natural or artificial, that inhabit it. This body of knowledge - indefinite no doubt, in its boundaries - goes by the name "common-sense". The

> problem we face is how to impart such knowledge to a robot. That is, how do we design a robot with a reasoning capacity sufficiently powerful and fruitful that when provided with some subbody of this knowledge, the robot will be able to generate enough of the rest to intelligently adapt to and exploit its environment? We can assume that most, if not all, common-sense knowledge is general, as is the knowledge that objects fall unless they are supported, that physical objects do not suddenly disappear, and that one can get wet in the rain.

The following simple example, given by Minsky, points out that knowledge representation is not a simple problem:

> The only time when you can say something like, "if *a* and *b* are integers, then *a* plus *b* always equals *b* plus *a*", is in mathematics. Consider a fact like "Birds can fly". If you think that common-sense reasoning is like logical reasoning, then you believe there are general principles that state, "If Joe is a bird and birds can fly, then Joe can fly". Suppose Joe is an ostrich or penguin? Well we can axiomatize and say if Joe is a bird and Joe is not an ostrich or a penguin, Joe can fly. But suppose Joe is dead? Or suppose Joe has his feet set in concrete?

It is worth exploring this theme a little further. Some domains of knowledge, for example mathematical knowledge, are well-behaved in a certain sense, and are relatively straightforward to deal with. For example, a triangle is a 3-sided polygon, or the sum of the interior angles of a triangle is 180°. These facts are true of all triangles and can be used as definitions of the concept of a triangle.

For other domains of knowledge, it is not quite so straightforward. Some concepts, for example *bachelor*, have an explicit definition "a man who has never married" (at least that is true when the terms are used strictly!). However, the majority of names do not have simple definitions of this form. An important class of objects are *natural kinds* (naturally occurring species), for example *lemon*, and *elephant*. The book *Naming, Necessity and Natural Kinds* (Schwartz, 1977) contains a fascinating collection of papers on this subject which is well worth studying, if only to remind oneself that the problems of knowledge representation did not arise with the advent of digital computers, but have long been studied by philosophers whose writings ought not to be ignored by computer scientists.

Putnam in his paper "Is Semantics Possible?" in the above volume, looks in detail at natural kind objects. In the traditional philosophical view, the meaning of, say, "lemon", is given by specifying a conjunction of *properties*, akin to the definition of triangle. A lemon is something that has all of the properties in the definition. Putnam and the other authors in (Schwartz, 1977) challenge this traditional view. Suppose the defining characteristics of a lemon are "colour lemon", "tart taste" etc. The problem is that a natural kind may have abnormal members, for example there are green fruits that everyone would agree are lemons, and elephants with three legs are still elephants. It is argued that nouns meant to designate natural kinds do not have their extensions (the set of things to which they refer) determined by a finite number of concepts.

Suffice it to say in this chapter, that it is important when choosing a knowledge representation scheme for a particular domain of knowledge, to consider the types of objects in the domain.

Some of the issues that arise in knowledge representation are summarized below to give more of a feeling for the problems.

(1) *Expressive adequacy.* Is a particular knowledge representation scheme sufficiently powerful? What knowledge can and cannot particular schemes represent?

(2) *Reasoning efficiency.* Like all representation problems in computer science, a scheme that represents all knowledge of interest and is sufficient to allow any fact of interest to be inferred by no means guarantees that it will be possible to perform the inference in an acceptable time. There is generally a tradeoff between expressive adequacy and reasoning efficiency.

(3) *Primitives.* What are the primitives (if any) in knowledge representation? What primitives should be provided in a system and at what level?

(4) *Meta-representation.* How do we structure the knowledge in a knowledge base and how do we represent knowledge about this structure in the knowledge base?

(5) *Incompleteness.* What can be left unsaid about a domain and how do you perform inferencing over incomplete knowledge and revise earlier inferences in the light of later, more complete, knowledge?

(6) *Real-world knowledge.* How can we deal with attitudes such as beliefs, desires and intentions? How do we avoid the paradoxes that accompany self-referential propositions?

The remainder of the first part of this book describes four approaches to the knowledge representation problem which have acquired some degree of acceptability amongst researchers in the field. The four approaches are: logic, semantic nets, frames, logic and rule based systems. Subsequent chapters deal with each of these approaches in turn. The second part of the book covers some current research directions, and problems common to all of these basic approaches, for example the representation of time and the trade-off between expressive power and the computational efficiency of inferencing.

The next section gives a brief introduction to the four basic approaches.

1.3 Overview of the Basic Approaches

1.3.1 Logic

Mathematical logic is an attempt to make rigorous the reasoning process involved in mathematics. The starting point is the introduction of a symbolic language whose symbols have precisely stated meanings and uses. The next step is to define the rules by which these symbols can be combined and manipulated and then the properties of the resulting formal system are explored. Chapter 2 gives a detailed introduction to various systems of mathematical logic and their application to knowledge representation. In this introduction, we will give a flavour for the approach in a very informal style.

A recent paper by Sergot *et al.* (1986) describes the use of a certain system of logic to describe a large part of the British Nationality Act, 1981. The system of logic is known as definite Horn Clauses, which are essentially rules of the form:

$$A \text{ if } B_1 \text{ and } B_2 \text{ and } \cdots B_n$$

which have exactly one conclusion A, but zero or more conditions B. A simple example of a Horn clause is the following:

(Socrates is mortal) if (Socrates is a man)

The first clause of the British Nationality Act is as follows:

1.-(1).A person born in the United Kingdom after commencement shall be a British citizen if at the time of birth his father or mother is
(a) a British citizen; or
(b) settled in the United Kingdom.

Clause 1.-(1)(a) is represented as a first approximation by:

(x is a British citizen)
if (x was born in the U.K.)
and (x was born on date y)
and (y is after or on commencement)
and (z is a parent of x)
and (z is a British citizen on date y)

The symbols x, y and z are variables.

Using a slight extension of this mathematical apparatus, a major part of the British Nationality Act was represented. Having obtained such a representation, it can then be manipulated using the rules of logical inference, appropriate to this system of logic, so that answers to queries such as "is Peter a British citizen on 16 January 1984 given that he was born on 3 May 1983 in the U.K. and is still alive and his father William ...", can be given.

1.3.2 Semantic Networks

The study of semantics is an attempt to describe the concepts behind word meanings and the ways in which such meanings interact. It is such a description which semantic networks were designed to provide. A network is a net or graph of nodes joined by links. The nodes in a semantic network usually represent concepts or meanings (e.g. BOOK, GREEN) and the links (or labelled directed arcs) usually represent relations (e.g., a book IS COLOURED green).

Semantic networks may be loosely related to predicate calculus by the following substitution: *terms* are replaced by *nodes* and *relations* by *labelled directed arcs*.

A large number of semantic networks have been developed as variations on this simple pattern since Quillian (1968) first used one in a computer system. These networks share few assumptions, although they nearly all represent the relations between concepts using a semantic representation consisting of a network of links between nodes, a set of interpretative processes that operate on the network, and a parser. They also show a general commitment to parsimony.

The most often used link in semantic networks was introduced in Quillian's system to show that one concept is an example of another (e.g. canary IS-A bird). More recent systems have chosen their link and node types on the basis of epistemelogical concerns about how the knowledge will be used. These have shown that even the apparently simple IS-A relationship is more complex than had been previously believed.

Recent developments in semantic networks together with work on the theoretical underpinnings of this approach are reviewed in the chapter by Mac Randal.

1.3.3 Frames

The use of nodes and links to represent concepts and relations seems straightforward, but contains many pitfalls.

Some designers of network systems were not too careful about the way in which they assigned meanings to nodes. Thus, a type node labelled "elephant" might well stand for the concept of elephant, the class of all elephants, or a typical elephant. Similarly token nodes labelled elephant were open to interpretation as a particular elephant, an arbitrary elephant etc. Different interpretations support different sets of inferences and so the distinctions are important. There was thus a sense in which semantic network formalisms were logically inadequate in that they could not make many of the distinctions that can be easily made in mathematical logic, for example between a particular elephant, all elephants, no elephant etc.

Frames are ways of grouping information in terms of a record of "slots" and "fillers". The record can be thought of as a node in a network, with a special slot filled by the name of the object that the node stands for and the other slots filled with the values of various common attributes associated with such an object. Frames are particularly useful when used to represent knowledge of certain stereotypical concepts or events. The intuition here is that the human brain is less concerned with defining strictly the properties that entities must have in order to be considered as exemplars of some category, and more concerned with the salient properties associated with objects that are typical of their class.

Frame systems reason about classes of objects by using stereotypical representations of knowledge which usually will have to be modified in some way to capture the complexities of the real world, for example that birds can fly, but emus cannot. The idea here is that the properties in the higher levels of the system are fixed, but the lower levels can inherit values from higher up the hierarchy or can be filled with specific values if the "default" fillers are known to be inappropriate.

1.3.4 Rule Based Systems

A classic way to represent human knowledge is the use of IF/THEN rules. The satisfaction of the rule antecedents gives rise to the execution of the consequents - some action is performed. Such production rule systems have been successfully used to model human problem-solving activity and adaptive behaviour.

More recently, substantial knowledge-based systems have been constructed using this formalism, for example the R1/XCON computer configuration system, implemented in the OPS5 production rule language. Chapter 5 describes the basic operations of a production system and the problems which arise in systems involving large numbers of rules, as well as considering the suitability of this formalism as a general knowledge representation.

1.4 Psychological Studies of Knowledge Representation

The second part of the book is a review of how we, as humans, appear to represent knowledge.

In this chapter the schemes which have been suggested as being those used to represent knowledge in human memory are reviewed. These include the use of frames, schema, semantic nets and production rules described in the earlier chapters. Instantiations of these are described for which both the representations and processes acting on them are specified in sufficient detail to enable experimentally testable hypotheses to be drawn. Experimental evidence is presented which supports an argument that schemes using only one of these representation mechanisms are inadequate to account for the full range of phenomena exhibited in human performance, although individual models can account for the specific sets of phenomena which they are intended to address.

A class of analogical representations is introduced which has not been described in earlier chapters but which are capable of supporting the phenomenon of visual imagery. Evidence is presented as to the use of imagery by humans and the nature of the representations which would have to support it. This suggests that although it is possible to account for visual imagery by processes acting on a propositional representation, it seems more likely that some form of analogical representation is used by humans.

One use of analogical representations is to form models of situations so that reasoning can be performed on them. Johnson-Laird's (1983) suggestions as to how such *mental models* could be used to support inference are described, along with findings which suggest the use of both propositional and such analogical representations by humans. As well as describing the limitations of suggested representation schemes and providing evidence that supports the use of multiple forms of representation, this chapter provides a set of phenomena for which any representation scheme will have to account if it is to address the range of human performance.

1.5 More Advanced Topics in Knowledge Representation

The third part of the book reviews a selection of more advanced topics.

1.5.1 Conceptual Graphs

In Chapter 7 Jackman and Pavelin give an overview of the basic concepts of the conceptual graph knowledge representation language. This includes the concept of the conceptual graph, the type hierarchy, the basic operations that may be performed on conceptual graphs, and logical deduction. Reference is also made to the "maximal join" - one of the fundamental derived operations in the language. This operation would appear to be equivalent to the graph equivalent (with a type hierarchy) of unification.

1.5.2 The Explicit Representation of Control Knowledge

Production systems have been used in many knowledge-based systems to model human expertise in classification. For example, the MYCIN family of expert systems can identify which microbial organisms are producing symptoms of disease in a patient. Important criticisms of such systems have been made by Clancey and others. Although the systems effectively "do the job" of the expert physician, much of the knowledge has been compiled, which is to say that it has been compressed and restructured into effective procedures. Bainbridge in Chapter 8 shows this makes it difficult to re-use the knowledge in explanation and knowledge acquisition subsystems, since the knowledge is implicit and therefore unavailable.

An important research area involves reconstructing these systems to make the knowledge explicit and available for use, and from these implementations extracting general principles for making better expert systems which more effectively represent the knowledge in their domain.

1.5.3 Representing Time

One of the most fundamental, and deceptively simple, representations that humans have is that of time. A great deal of effort has been expended on attempting to formulate temporal representations for use by knowledge-based systems. Chapter 9 considers first the basic issues in the representation of time, such as the choice of point or interval representations, the treatment of fuzziness and granularity, and the problem of persistence. A number of approaches are then presented, with reference to the systems in which they have been used or the contexts in which they are appropriate. State-space modelling, date-based methods and before/after chains are all covered, along with temporal logics, which have attempted to place representations of time on a formal foundation.

1.5.4 Functional Approaches

An important approach to knowledge representation is the functional approach pioneered by Levesque and Brachman. There is a relation to mainstream computer science in that a knowledge base is regarded almost as an abstract data type with a set of operations defining the services it provides. The approach is motivated by the misuses or misinterpretations of knowledge representation formalisms which can occur when the user is allowed unrestricted access to representational structures: for instance, the nodes and links of a semantic net. Chapter 10 discusses some early work, and then describes Levesque's formalization in which he defines operators TELL and ASK for interacting with a knowledge representation system. Finally, the KRYPTON system is dealt with. It is the most advanced implementation of functional ideas, and it also incorporates multiple representations in having a taxonomic component for defining absolute relationships and an assertional component for making statements.

1.5.5 Expressive Power and Computability

There is a fundamental difference between a knowledge representation system and a database: the former will in general perform inferencing of some kind in order to answer queries about what is represented, while the latter is limited to retrieving the facts it contains. Databases cannot therefore represent incomplete information, for everything must be stored explicitly. Knowledge representation systems are more expressive, and their inference capabilities mean that they can act on incomplete knowledge. Indeed, when there is incomplete knowledge, queries to a database concern no more than what the database happens to contain; only a knowledge representation system can go further and attempt to deal with the world it represents. Of course the price to pay is in the computational effort needed to answer queries - the trade-off between the two factors is discussed in Chapter 11 by Williams and Lambert.

It is well-known that full first-order logic is not decidable, that is, a theorem prover cannot be guaranteed to terminate. Restricting the expressive power of the representation language results in systems that exhibit various degrees of tractability: though decidable, some are NP-complete, while others, less expressive, admit inferencing algorithms that operate in polynomial time. A number of the knowledge representation schemes described earlier in the book are discussed in these terms. It is not yet understood precisely how the tractability of a knowledge representation system depends on its expressiveness though there are some indications, but the trade-off may have important implications for our view of what service is expected of such systems.

2 Logic in Knowledge Representation

Cliff Pavelin

If your thesis is utterly vacuous
Use first-order predicate calculus
 With sufficient formality
 The sheerest banality
Will be hailed by the critics: "Miraculous!"

(Henry Kautz, from Canadian Artificial Intelligence, 9, 1986)

2.1 Introduction

Logic was originally developed to formalize the principles of valid reasoning. It has been studied since the time of Aristotle, although what is now regarded as Classical Logic was invented by Frege in the last century. His notation was diagrammatic and cumbersome; the current symbolic notation was introduced by Peano and perfected by Russell and Whitehead in 'Principia Mathematica'.

Logic attempts to make rigorous the reasoning process involved in science or mathematics; indeed Principia Mathematica was an attempt to reduce mathematics to Logic. It thus arises naturally in areas where deductive proof is required - for example proof of a geometrical theorem or proof that a computer program has the effect expressed by its specification. But Knowledge Representation problems typically relate not to formal domains but to ordinary discourse, to problems of everyday life, which are solved by informal reasoning often difficult to characterize. It is in such domains that the role of logic is not so clear.

An introduction to Moore's paper (1985a) in 'Readings in Knowledge Representation' observes 'an often furious debate over the proper role of formal logic in Knowledge Representation has raged almost unabated since the very beginnings of the field'. Moore's paper is in fact a prominent representative of the 'logicist' position, as is Hayes (1977a), while well-known expressions of the 'non-logicist' viewpoint are given in the Appendix to Minsky (1975) and Newell (1980). McDermott (1987) presents an account of the logicist position from the point of view of someone who has become less convinced. The two sides are respective subsets of the 'neats' and 'scruffies' identified by Bundy (1982).

What is it all about? Like many debates the issues become confused. Israel in a good analysis of some of these confusions (Israel, 1983) believes that failure to sort them out is one of the reasons for the inconclusive nature of the arguments. Not least of the problems is a lack of a consistent definition of 'logic'. In AI, it is likely to mean one of the following:

(a) First Order Logic (FOL) summarized in the next section.

(b) Some development of FOL which maintains its notation, its notion of a formal language, a deductive proof theory and a well defined model theory.

(c) Any *formally defined* method of representing knowledge and making inferences about it.

The alternative to all these is to have reasoning techniques that are embodied inside a computer program, non-explicit and without any general principles.

In this chapter we assume definitions (a) or (b) and examine the benefits and deficiencies of logic, in these terms, for the representation of knowledge. We do not take sides in the debate; indeed the cynic may suggest it is a somewhat contrived polarization which came about to give stimulus and sparkle to the development of the subject. However, long may it continue.

Logic is, by definition, formal, while this chapter, in an attempt to give an insight into the basic principles, is informal throughout. There are numerous modern textbooks on formal logic available and it should also be noted that many books on Artificial Intelligence (e.g. Nilsson, 1982; Frost, 1986) give substantial introductions to logic.

2.2 A Brief Introduction to First Order Logic

The modern basis of logic is 'First Order Logic' (FOL) - also known variously as Classical Logic, Predicate Calculus (PC), Lower PC, FO functional calculus and general logic. Most of the other logics being studied in AI are developments (in some cases supersets) of FOL and they inherit at least some of its notation, limitations and advantages. This section gives a very brief introduction to those elements of FOL which are most relevant to the discussion on Knowledge Representation.

2.2.1 Basic Elements

FOL attempts to abstract the essential features of deductive reasoning and express them in what could be called an algebra of propositions. Propositions are statements which can be regarded as either TRUE or FALSE - no half measures are allowed. For example:

144 is a square number
the internal angles of triangle ABC total 180 degrees
Puccini wrote 10 operas

are three propositions (the last is false).

FOL is defined on two quite distinct levels. At one level it is a formal language with formation rules to generate sentences ('well-formed formulae') in the language. At this level propositions are typically denoted by symbols like p,q,r etc. A correspondence can then be set up between the symbols of the language and objects or values in some domain - which may be arithmetic or Euclidean geometry or the 'real world' or whatever. The sentences in the language then map on to statements about the objects in the domain. Mapping on to a domain is known as *interpretation*. For example a sentence in the language might simply be:

p

Under a particular interpretation this may map onto the TRUE proposition that Puccini wrote 13 operas. If under a particular interpretation each of a set of sentences is TRUE, the interpretation is termed a *model* of those sentences.

It is important to maintain this distinction between language and interpretation. To say 'p is TRUE' is really nonsense as p is a symbol that cannot be true or false. The statement is just a shorthand either for 'p is asserted or can be proved' or sometimes 'under an interpretation currently being assumed, p maps onto a predicate that is deemed to be TRUE'.

Connectives

The next stage is to build up more complex expressions in the language by use of a further set of symbols known as 'connectives'. These correspond (under interpretation) to well known Boolean operators. Typically the ones used in logic are $\wedge$ (and), $\vee$ (or), $\neg$ (not) and $\rightarrow$ (implies).

$p \wedge q$	is TRUE only if both p and q are TRUE
$p \vee q$	is TRUE if at least one of p and q are TRUE
$p \rightarrow q$	is TRUE unless p is TRUE and q is FALSE
$\neg p$	is FALSE if p is TRUE and vice versa

$\wedge$, $\vee$ and $\rightarrow$ can be termed 'Dyadic' operators - they act on two truth values and produce a third. Since each proposition can have two possible values (TRUE or FALSE), there are four possible combinations of values, and the result of a connective must define a TRUE or FALSE for each, there are sixteen (four squared) possible connectives that could be defined (some are well known as basic electronic operations like NOR, NAND etc). The choice made in classical logic is somewhat arbitrary: those given above are traditional and make sentences easy to understand. There is redundancy - for example $\vee$ can be defined in terms of $\neg$ and $\wedge$:

$$p \vee q \text{ is equivalent to } \neg (\neg p \wedge \neg q)$$

It is possible to define all the connectives in terms of one carefully chosen one (for example $\neg (p \wedge q)$) and this is often done in formal mathematical logic in order to minimize the basic concepts - but expressions and proofs are then obscure to the human reader.

It is important to realize that 'implies' is just another logical connective.

$$p \rightarrow q$$

does not, under an interpretation mean there is a causal link between whatever p and q map onto. For example:

144 is a square number → Puccini wrote 13 operas
30 is a prime number → Puccini wrote 10 operas

are both TRUE in FOL although there is no causal link in either and neither the antecedent nor the consequent is true in the second (see section 2.5.2).

Using the connectives (and brackets) as above, expressions can be built up of arbitrary complexity, for example:

$$(p \wedge q \wedge r) \rightarrow ((p \vee q) \wedge (p \vee r))$$

and there is an elementary algebra of these symbols which allows simplification of expressions etc. As an example:

$\neg (p \wedge q)$ is equivalent to $(\neg p) \vee (\neg q)$

such transformations being easily checked by a 'truth table' which gives all possible assignments to p and q.

2.2.2 Predicates and Quantification

It is possible to construct a logic purely on the basis of representing propositions as above - generally known as 'propositional', 'primary' or 'sentential' logic'. But it is very limited in what it can express. First Order logic goes a very significant stage further in the refinement by which it can represent propositions by being able to represent statements about members of classes of objects. Take as example two of the propositions given previously:

The angles of triangle ABC total 180 degrees.
Puccini wrote 10 operas

These might be generalized by writing

The angles of triangle x total 180 degrees
y wrote z operas

As written, with the 'variables' x, y, z, these sentences mean nothing, but the statements acquire a meaning if the variables are *quantified*. In FOL, two quantifiers are introduced: ∀ (meaning 'for all'...) and ∃ (meaning 'there exists' ..).

Thus the following are first order sentences:

For all x, the angles of triangle x total 180 degrees

There exists an x, y such that x wrote y operas
(i.e. someone wrote some operas).

In the notation of FOL the above statements would be represented as:

$\forall x \;\; p(x)$
$\exists x,y \;\; q(x,y)$

p(x) represents the incomplete proposition 'the angles of triangle x total 180 degrees'; p and q are *predicate symbols*; x and y are known as *variable symbols* which must be quantified either by ∃ or ∀.

A predicate thus (under interpretation) represents the set of objects for which a certain property is true. In this case p represents *all* triangles. q is true for all pairs of people and numbers such that the person wrote the given number of operas.

It is this ability to make statements about all objects or assert the existence of objects, without necessarily mentioning the individuals, which gives FOL its power.

The language of FOL also has constant symbols which represent specific objects, and function symbols which map onto functions:

$\exists$ x q(x,c) where c is a constant symbol

Under an interpretation in which q had the meaning above and c mapped onto the number 10, this would mean:

Someone (i.e. some x) wrote 10 operas.

There is an algebra which allows some simplification of expressions and transformation into certain canonical forms. A simple example (which shows the redundancy in $\exists$ and $\forall$) is

$\exists$ x (p(x)) is equivalent to $\neg$ ($\forall$ x $\neg$p(x))

Example

$$\begin{aligned}\forall\, x,y\;\; & g(x,y) \rightarrow \\ & s(x,y) \vee (\exists\, z\; s(x,z) \wedge g(z,y))\end{aligned}$$

(For all x and for all y, g(x,y) implies either s(x,y) or the existence of some z such that s(x,z) and g(z,y)).

One interpretation of this is in the domain of integers where

g(x,y) means $x > y$
s(x,y) means $x = y + 1$ (x is the successor of y)

Then this sentence means for any choice of integers x and y, if $x>y$ then either $x=y+1$ or there is some number z which is greater than y and one less than x. This is plainly true and therefore this interpretation is a model of the sentence.

Another interpretation is the domain of people: g(x,y) means ‘x is an ancestor of y’, s means ‘is a mother of’. This says that if x is an ancestor of y then x is a mother of y or a mother of some ancestor of y. This interpretation makes the sentence FALSE; it would be a model if ‘mother’ were replaced by ‘parent’.

2.2.3 Interpretations and Models

The term ‘interpretation’ has been used as a mapping from the language (the syntactic level) onto some domain in which ‘TRUE’ and ‘FALSE’ have some meaning. This does not necessarily imply that an interpretation has any real-world connotations. A proposition p may correspond to an infinity of statements but *all that is relevant to Logic* is whether it is TRUE or FALSE - it has just TWO essentially different interpretations. A sentence in propositional logic (i.e. no variables) involving n distinct propositions has just 2^n interpretations - corresponding to an assignment of TRUE or FALSE to each proposition. The sentence:

$$(p \vee q) \wedge (p \vee r)$$

has just eight interpretations - the meanings of p and q and r in the normal sense, are irrelevant.

The same principle applies in predicate logic where there are variables, but the range of values of the variables must be specified and since this may be an infinite domain, the number of interpretations is typically infinite:

$$\exists\, x\ p(x)$$

To give an interpretation of this sentence means giving the domain of the variable - and then specifying the subset for which p is going to be designated TRUE. There is thus an infinite number of interpretations and all those for which p is designated true in at least one instance are 'models'.

An interpretation is thus rigorously defined, but it is concerned with no more of the content of a proposition than simply 'truth' or 'falsity'. This definition of truth and falsity by mapping from the language onto a domain is known as giving the logic a *semantics* (after Tarski). However, one should not necessarily equate 'semantics' with 'meaning' in the real world.

2.2.4 Deductive Reasoning

Deductive reasoning consists of taking a set of sentences, inferring from it new sentences which can form the basis of further inferencing and so on until some 'interesting' conclusion is reached.

In this way an edifice of theorems is built up upon a few assumptions, the system of Euclidean geometry being the exemplar. Each step in this process is the application of one of a few simple rules of inference. The aim of logic is to formalize and automate this process of reasoning; an ideal might be the mechanical generation of the whole of Euclid from his simple postulates.

Deductive reasoning is applied to sentences of the FOL languages without reference to what they mean in any interpretation. Some scheme of inference (variously rules of inference, 'axiom schema' or sometimes a diagrammatic method) is defined which effectively gives transformation rules which can be applied mechanically to sentences to form new sentences.

An example of a simple rule of inference is as follows. If

$$A \rightarrow B$$
$$A$$

are given, then

$$B$$

can be derived as a theorem. In this rule (the well-known '*modus ponens*'), A and B may stand for any well formed formulae. Other (fairly obvious) examples might be:

If $A \wedge B$ is proved, then A can be derived.

If $\forall x\ p(x)$ is proved, then $p(c)$ can be derived
where c is any constant symbol.

Any scheme of inference must preserve truth in any model (i.e. in any interpretation which makes the assumptions true). In the cases above this is obvious by the definitions of the connectives.

A proof starts from a set of 'axioms' and that proof will then be relevant to any model of these axioms. (A set of axioms, plus all the FOL rules etc. is sometimes called a 'theory'.) Since the same set of axioms may characterize many models, theorems in different domains may have a single proof when the abstraction into logic is made. For example, from the statements:

All men are mortal
I am a man

it follows:

I am mortal.

Another argument might run:

All AI professors are mad	(1)
Professor X. is an AI professor	(2)
Therefore (theorem) Professor X. is mad	(3)

In first order logic, both these arguments are represented by:

$\forall x\ p(x) \rightarrow q(x)$	(1)	assumption
$p(c)$	(2)	assumption
$p(c) \rightarrow q(c)$	(2a)	from (1)
$q(c)$	(3)	from (2) and (2a)

There are many schemes of inference for FOL - they are all equivalent in terms of what can be proved from a given starting point. An FOL proof scheme should have the following properties:

(a) It should be SOUND - any theorem must be true in any model of the axioms.

(b) It should be COMPLETE. If some sentence is TRUE under *every* model of its axioms, then it should be derivable from the axioms.

If (a) were not true, we could prove theorems which were false in a model - equivalent to showing both p and $\neg$ p.

If (b) were not true, we have two definitions of a theorem: (i) a sentence derivable from the axioms, (ii) a sentence which is true in all models. In higher order and other logics they are not necessarily the same - there are for example 'truths' in arithmetic which cannot be proved from the normal axioms of arithmetic. However FOL is complete in this sense.

2.2.5 Validity and Consistency

It is possible to construct sentences which are *inconsistent* - which no interpretation can possibly make true. Examples are

$$p \wedge (\neg p)$$

or

$$\neg p(10) \wedge \forall x (p(x))$$

The second, for example, is saying (under any interpretation) that some property is false of some object 10 but is also true for all possible objects. The inconsistency is obvious in these simple examples but in general will not be so. A set of sentences for which a model does exist is known as *consistent*.

Conversely a sentence can be a *tautology* - this means it is true in *every* interpretation. Examples are:

$$(p \wedge q) \rightarrow (p \vee q)$$
$$p(c) \rightarrow \exists x \; p(x)$$

Sentences like this are known as *valid*.

Since the rules of inference preserve truth under interpretation, a model of a set of sentences is still a model for all sentences derived under the rules of inference, i.e. for all theorems. Everything it is possible to prove from a set of consistent axioms forms a consistent set.

If the axioms are inconsistent, the whole system breaks down - in fact anything can be proved. If the axioms are valid, nothing can be proved other than more tautologies which can be proved anyway using the rules of inference and starting with nothing!

Proving a theorem can easily be seen as equivalent to checking a set of sentences for validity. Suppose A, B and C are a set of axioms and we wish to know whether a theorem T can be derived from them. If it can, then T must be TRUE under any interpretation which makes A, B and C TRUE. In other words no possible interpretation could make:

$$A \wedge B \wedge C \wedge (\neg T)$$

true. The proof of T is thus equivalent to showing this set of sentences is inconsistent - there is no model. It is also equivalent to showing that its negation:

$$(\neg A) \vee (\neg B) \vee (\neg C) \vee T$$

is always TRUE, i.e. the expression is valid.

How does one go about checking for validity? In propositional calculus, it is obviously always possible since the value of the expression can be calculated for every possible assignment of truth values (although in practice this may be a huge number). It is not so obvious for FOL, but in fact there are decision procedures which guarantee to terminate if an expression is valid. This is equivalent to the remark above that FOL is complete.

It can be shown there is no procedure which can be guaranteed to complete if the set is non valid - FOL is said to be semi-decidable for this reason. In other words it is not in general possible to prove that a sentence is *not* a consequence of some theory.

2.2.6 Theorem Proving in FOL

We have said that rules of procedure can be devised which are sound and complete for FOL, i.e. can, in principle, be used to derive any theorem. Devising such rules is unfortunately not the difficult problem. The difficulty which gives rise to most of the research into theorem proving is devising rules which are *efficient* for the typical problems which arise in practice. Issues of decidability are of little concern if proving a simple theorem takes millions of years of Cray time. Much research into computer methods of theorem proving was stimulated by a uniform method, suitable for automated inference, known as 'resolution'. Theorem proving in FOL is a very active research area although it is unfortunately still the case that some logical puzzles which may be very simple for an intelligent human to solve are quite stringent tests of state-of-the-art theorem proving.

2.3 FOL in the Representation of Knowledge

The semantics of FOL gives the basis of a powerful mechanism to represent knowledge of the real world in logic. As will be seen, however, this is much easier in 'formal domains'.

2.3.1 Formal Domains

To map a domain onto FOL, it must be possible to regard the domain as consisting of objects plus properties of, or relations between, the objects which in any particular case can be designated as TRUE or FALSE. In 'formalized' domains - for example mathematics, the law, the information represented in a computer database - this is likely to be the case and a mapping into FOL or some extension of it is normally a natural one.

A relational database can be regarded as already in FOL form: each relation type corresponds to a predicate and each relation to a proposition expressed with this predicate. Thus one may have an office-building database storing information about room numbers, occupants and telephones in each room. Suppose there are two relations: OCCUPANT (giving room numbers and occupants) and TELEPHONE (giving telephone numbers and occupants):

OCCUPANT		TELEPHONE	
person	room	number	room
D.Owen	1	345	1
D.Steel	1	123	1
M.Thatcher	2	639	2
N.Kinnock	3	639	3

Messrs. Owen and Steel share a room but it has two telephones in it, while Thatcher and Kinnock each have their own room but have a party line. Each relation can be viewed as a FOL predicate with an obvious interpretation in the real (or perhaps pretend) world represented by the database. Thus the FOL sentences would be:

occupant(D.Owen,1)
telephone(639,2)

etc.

The interest in logic database systems is to add *rules*, which can be expressed as logical implications, to such systems. The traditional use of relational databases often assumes implicit rules, e.g.:

(a) Objects have unique identifiers: room 1 and room 2 cannot stand for the same thing.

(b) If a relation does not exist, one can assume its negation: there is no telephone 639 in room 1 (this is known as the Closed World Assumption).

(In fact neither of these implicit rules can quite be expressed in FOL.)

Logic Databases can introduce *explicit* rules written in FOL, e.g.:

(c) $\neg \exists$ x occupant(x,4)
Room 4 is empty

(d) $\forall$ x,r occupant(x,r) $\wedge$ even(r) $\rightarrow$ nonsmoker(r)
Even numbered rooms contain non-smokers

This allows the construction of 'deductive databases' - one can infer much more than those facts which are directly stored in the initial relations. However, there are a number of theoretical problems, for example databases get updated. If a relation is added giving a definition of an occupant in room 4, the database is inconsistent with the rule (c) given above. Once implications are allowed, there are problems in avoiding inconsistency with the implicit rules given above. Particularly (b) becomes tricky even to define because a relation can be inferred from the rules rather than given explicitly in the database. Such problems are an active research area (Gallaire and Minker, 1978; Flannagan, 1986).

The issues are of much more than theoretical interest; if full FOL deduction is allowed, anything at all can be deduced from inconsistent data. However the problems are about consistency and proof in logic rather than about Knowledge Representation itself. It should be noted that the mere existence of a relational database assumes that the knowledge is highly structured already.

2.3.2 Non-formal Domains

In mapping formal systems like the database onto logic, the database system itself is a model of a logical theory and there is likely to be a straightforward mapping between the facts in the database and the real world. However in the case of real-world knowledge of an unformalized type, the mapping is not so obvious.

Take the example as a piece of 'knowledge':

John gives a book to Mary

This is a proposition, but simply representing it by a predicate p would be totally unhelpful. One would have as many predicates as ideas that could be expressed and no way of analysing their content. The constituent concepts that appear to be basic here are 'John', 'Mary', 'book' and 'give'. A natural approach may be to take 'John', 'Mary' and 'book' as objects and 'gives' as a relation between them. The import of the above sentence is presumably that John and Mary are specific individuals but that the book is some unspecified book.

$\exists$ x book(x) $\wedge$ gives(John, x, Mary)
i.e. there is some x which is a book and John gives x to Mary

One may say in addition that John and Mary are human beings:

human(John)
human(Mary)

However this representation will make it impossible to treat this act of giving as an object about which one may want to specify information - the time it took place (or does one want a special predicate for 'gave'?), the place etc. A more basic formulation can specify a 'giving event' and then associate other predicates with this, e.g.:

giving (e1)	e1 is a giving 'event'
agent(e1,John)	John was the giver
recip(e1,Mary)	Mary was the recipient
$\exists$ b, book(b) $\wedge$ object(e1,b)	A book is the object of giving

The act of 'giving' is then an individual in the domain just as are John, Mary, etc. Temporal information could be added:

time(e1,T)   the giving was at time T

This would seem to be a reasonable FOL formulation of the sentence. However one will presumably want to express also more general ideas about 'giving' which would allow reasoning about the situation. For example:

$$\forall \text{ e1,y,t,x giving(e1)} \wedge \text{recip(e1,y)} \wedge \text{time(e1,t)} \wedge \text{object(e1,x)} \rightarrow$$
$$(\forall \text{ t1, t1} > \text{t} \rightarrow \text{owns(y,x,t1)})$$

This is saying that if y was the recipient of x at time t then he becomes the owner for all time after this. (Notice that > signs have slid in here - let us assume they can be defined as a predicate.)

Of course this rule is not always true in the real world; in practice it is difficult if not impossible to write universal rules about the real world unless they be true by definition ('all dogs are mammals', for example). However, such a rule as above might be taken as true in some idealization of the world which a certain application can assume. In representing the real world as a logical model one has two problems - first idealizing the world and then selecting the best FOL formulation.

2.3.3 Semantic Networks

Various network and structured object representations described elsewhere in this book have been developed specifically for dealing with knowledge about the real world and it is very instructive to investigate the extent to which they are equivalent to logic.

A semantic network can be mapped into FOL by taking its nodes as corresponding to terms and its arcs to relations. Since all arcs link two nodes, the predicates will be binary. As pointed out in Deliyanni and Kowalski (1979), the semantic network representation draws attention to the advantages of using only unary and binary predicates in a logic formulation (as above). Some care must be taken because (typically) the nodes of a semantic network may be individuals or types. Figure 1 adapts an example from Deliyanni and Kowalski (1979), representing the example given in the previous section. Here the links from Mary and John to Human are 'instance-of' relations and are probably best represented by a predicate 'human' and statements of the form:

human(Mary)
human(John)

The 'isa' relation is often used in semantic networks to denote the subset (subtype) relation, as well as the instance-of relation given above.

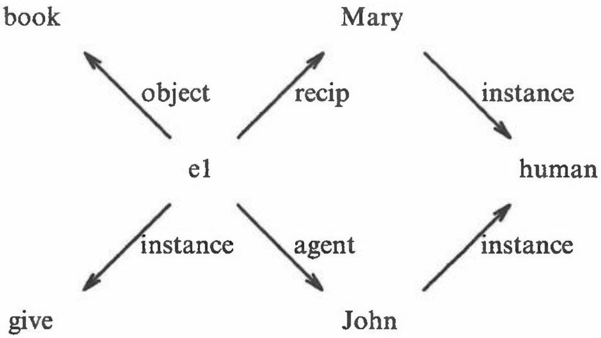


Figure 1

$$\text{human} \xrightarrow{\text{isa}} \text{animal}$$

This means ‘all humans are animals’; a translation into logic would read:

$$\forall\ x\ \text{human}(x) \rightarrow \text{animal}(x)$$

One can treat these links like any others, by having predicates ‘instanceof’ and ‘isa’, but this would be introducing predicates for concepts that are already built in to FOL and an additional set of axioms about these predicates would have to be introduced.

In the ‘conceptual graph’ representation of Sowa (1984), there are two types of node: concepts and relations (arcs are unlabelled apart from direction). Relations take the place of arcs in the normal semantic network. The previous sentence is represented by the diagram in Figure 2. A common confusion between ‘isa’ and ‘instance-of’ is removed in Sowa’s system by maintaining a type hierarchy external to the network. Thus the fact that:

$$\text{human} \xrightarrow{\text{isa}} \text{animal}$$

is not represented in the graph itself - indeed Sowa claims that being a different ‘order’ of link from those given, it should not be.

The ‘instanceof’ relation is similarly specially treated: the ‘John’ attached to the ‘human’ box is a referent which says that the individual John is of type ‘human’. Sowa gives a formal mapping from conceptual graphs into FOL; this one would be:

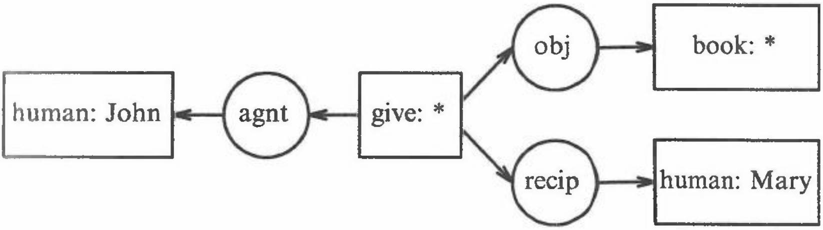


Figure 2

$$\begin{aligned}\exists\, e1,b1\ \ \text{human(John)} &\wedge \text{agnt(John,e1)}\\ &\wedge \text{recipient(Mary,e1)} \wedge \text{gives(e1)}\\ &\wedge \text{obj(b1,e1)}\\ &\wedge \text{human(Mary)} \wedge \text{book(b1)}\end{aligned}$$

Notice the concepts of Give and Book were not given individual referents and so existential quantification appears in the FOL formulation. Fuller details are given in Chapter 7.

These 'isa' and 'instance-of' relations map into particular cases of quantification in FOL. But there is no generally accepted notation for quantifications, disjunctions ($\vee$s) and implications in semantic networks although there have been a number of proposals (see Deliyanni and Kowalski, 1979; Sowa, 1984).

2.3.4 Frames

Mapping properties of frames (see Chapter 4) into FOL demands some assumptions about what they mean. Hayes (1979) makes assumptions but then proceeds to show that an FOL mapping is, on the whole, straightforward. A frame type represents some generic concept; an instance of the frame type, with appropriate values in the slots, represents an instance of this concept. Suppose C is a frame type and x is an instance, then C(x) can be taken as the predicate corresponding to 'the object x is a concept of type C'. Thus if C is the 'house' frame, and '10 Downing Street' an instance, then in FOL we would assert:

C('10 Downing Street')

interpreted by the fact that '10 Downing Street' is a type of house. The properties of 'slots' correspond to two sorts of logical implication. The first says that frame instances have slots with appropriate values:

$$\begin{aligned}&\forall\, x\ C(x) \rightarrow \exists\, y1\ \ RC1(x,y1)\\ &\forall\, x\ C(x) \rightarrow \exists\, y2\ \ RC2(x,y2)\\ &\ \ \ldots\end{aligned}$$

The yn represent slot values. The RCn predicate represents the relation between a frame and its nth slot; thus RC1(x,y1) would mean that the value of the first slot of the frame x is y1, and furthermore it is appropriate for that slot. If C is the house concept with the first slot giving the number of rooms, then a model of RC1(x,y1) would check that the y1 was a number in the right range.

The second mapping is slightly more subtle: some applications of frames assume that if some object has all its slots filled with appropriate values for a certain type of frame, then indeed it is taken as an object of that type.

$$\forall \text{ x,y1,...,yn } \ \text{RC1(x,y1)} \wedge \ldots \wedge \text{RCn(x,yn)} \rightarrow \text{C(x)}$$

In like manner the typing and inheritance characteristics of frames can be largely mapped onto FOL. Suppose we wish to say that the frame type 'bungalow' is a 'house' with its 'number of floors' slot constrained to contain the number one:

$$\forall \text{ x bungalow(x)} \rightarrow \text{RB1(x,1)}$$
$$\forall \text{ x bungalow(x)} \rightarrow \text{house(x)}$$

(It is assumed that the first slot gives the number of floors.) The second implication would show, together with the axioms given formerly, that the 'bungalow' is constrained by properties of slots in the 'house' frame.

One common characteristic of frames - inheritance of default values where no current values are given in the slot of a subtype - raises the same issues in mapping onto FOL as those addressed by 'non-monotonic logic'. We return to this in section 2.6.1. A general discussion of Hayes' 'Logic of Frames' is given in section 4.4.

2.3.5 An Advantage of Network Systems

If semantic networks and structured object representations, designed *a priori* to represent *ad hoc* knowledge, can map onto FOL or some extension of it, what advantages do these somewhat ill-defined formalisms have over the precise and well analysed language of logic?

The crucial advantage is practical rather than theoretical and is best shown by examples.

In a semantic net, all the predicates about Mary will have a link to the node Mary. A theorem-prover working on such a network and operating on this node may naturally follow links from it: compare this with looking at the set of predicates which contain Mary somewhere in them - an unlikely operation for an FOL theorem-prover.

Similarly the 'isa' link (or type hierarchy in the Sowa formulation) is explicit. In logic, to find out that Mary inherits the characteristics of a human, a theorem-prover has to stumble across statements of the form:

$$\text{human(Mary)}$$
$$\forall \text{ x } \ \text{human(x)} \rightarrow \text{some-property(x)}$$

One further example (taken from Reiter, 1985) is where a network representation is used to show mutual exclusiveness (see Figure 3). In some systems the diagram shown would be an is-a hierarchy which was meant to imply

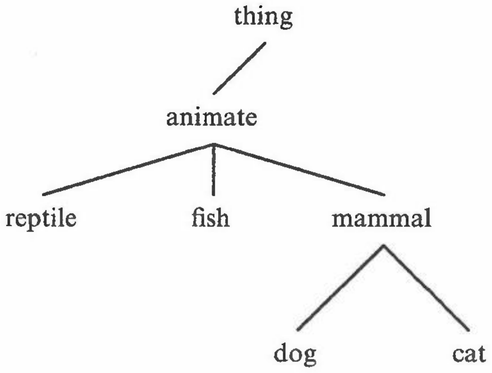


Figure 3

that reptile/mammal/fish are mutually exclusive classes. Although it is easily expansible into logic, the formulation is messy and the reasoning to discover that, say, a dog is not a reptile, is cumbersome.

This is all summed up by saying that a network representation embodies as *primitives* certain important relations or reasoning steps which in a *theoretically equivalent* logic representation may be deeply buried. The network may thus be clearer to understand and easier to reason with.

2.3.6 Logic Programming

Any computer program is a representation of knowledge and some mention must therefore be made of that programming paradigm known as 'logic programming'. In a logic program - the current manifestation being the now widely used Prolog language - the statements of the program can be regarded as assertions of a logical theory. The ability to make the equivalent of quantified logical statements, e.g. *all* of a set of objects have a certain property, gives a programming language of remarkable power. The aim of the program is to exhibit relevant inferences. Running the program is equivalent to proving a theorem in a sequential manner which makes predicates the analogue of procedures in a traditional programming language. The differences between running a program in Prolog and expressing a problem and solving it in FOL are:

(i) A Prolog program cannot express the whole of FOL; the particular restriction (to 'Horn clauses') is equivalent to disallowing implications which have disjunctions on the right hand side:

$$A \rightarrow B \vee C$$

(ii) A Prolog program typically contains features which are not equivalent to FOL (including the 'Closed World Assumption') or in some cases are not 'logical' at all. The advantage of such features is that they allow Prolog to be very effective as a general purpose programming language. The disadvantage is that they detract from the declarative character of the logical statement of the problem.

(iii) The built-in theorem-proving procedure is constrained to a specific technique (known as SL resolution). It is a straightforward one for a programmer to understand (at least when expressed in terms of implications given the Horn clause restriction) but, combined with the non-logical features mentioned above, can make answers dependent on statement ordering.

Current research into logic programming is attempting to design languages nearer to logic which will thus increase the declarative nature. At present however a typical logic program will combines features of pure logic with *ad hoc* representation or reasoning steps typical of any computer program.

2.4 Benefits of Logic

Before we discuss the deficiencies of logic in knowledge representation, it is as well to summarize its very cogent benefits.

2.4.1 Precision and Analysis

The great feature of FOL is its precisely defined and well understood notation, with a model theory (semantics) which gives precision to the mapping between the sentences of logic and some domain.

Thus the constructs:

$$P \xrightarrow{\text{isa}} Q \quad \text{(semantic net)}$$

p is a Q (English)

are made unambiguous when given a logical formulation:

$$\forall\ x\ P(x) \rightarrow Q(x)$$
$$Q(p)$$

If a representation can be expressed in FOL (or any of its derivatives) it is clarified not only to the reader but to the author. Subtle questions about the interpretation of a 'frame' can be seen to be quite unsubtle when expressed in logic.

It is difficult to think of any other ways than logic to explain precisely what a particular network or structured object formalism is supposed to denote. Sowa (1984) is careful to give all his constructs an expression in FOL other than when they go beyond what FOL can express.

2.4.2 Expressiveness

FOL is *expressive* in the sense that its notions of quantification and negation are not easy to represent unambiguously in network and structured data form. The difficulties of deciding what a frame formulation is meant to suggest will be returned to several times in this book - some of these problems amount to the differences between existential and universal quantification.

Expressiveness is discussed in Moore (1985a) where he observes that problems of reasoning and representation involving incomplete knowledge are typically solved only by systems of formal logic. Figure 4 shows his well-known example. Is there a green block next to a non-green block? The answer is clearly yes - but the statement of the problem and the reasoning required for the solution are difficult to give in many knowledge representation systems and require precisely the constructs of FOL.

2.4.3 Proof Theory

The triumph of FOL is its proof theory - in particular the completeness theorem that everything that is true in all models of a theory can be proved. There is a continuing large amount of work in developing wholly automated or computer-assisted proof mechanisms in FOL and its derivatives (some of which do not have the completeness property).

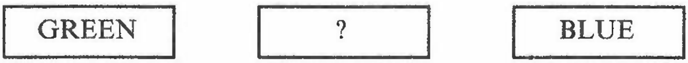


Figure 4

This feature is of course one of reasoning rather than representation. A and one of the criticisms of logic below will be the extent to which deductive proof is relevant to common-sense reasoning in a non-formal domain.

2.5 Limitations of Logic in Knowledge Representation

Any discussion of the limitations of Logic in knowledge representation must take into account what function logic is expected to play. It can be used to *represent* and *analyse* knowledge; it can also be used for (indeed was invented for) *deductive reasoning* over this knowledge. It will be seen that the limitations in representation have or are being successfully tackled by the development of more advanced logics than FOL. But the deficiences of logic in 'common-sense' reasoning (see sections 2.5.1, 2.5.3 below) appear to be more fundamental and have led some critics to deny a role for logic in representation. Israel (1983) analyses such criticisms; he concludes that although logical proof is but one tool to be used in reasoning, this is not at all a deficiency of logic in a representational role but an indication that one must clearly distinguish between logic and reasoning. Notwithstanding, we present here all the traditional arguments against logic.

2.5.1 Limitations of Deductive Reasoning

The major objection to Logic in knowledge representation applies only to its role in reasoning - this is the fact that most reasoning about the real world is not deductive. This is eloquently argued in McDermott (1987); that the argument is not new is shown in the following passage from Bertrand Russell (1945):

> The Greeks in general attached more importance to deduction as a source of knowledge than modern philosophers do. In this respect, Aristotle was less at fault than Plato; he repeatedly admitted the importance of induction, and he devoted considerable attention to the question: how do we know the first premises from which deduction must start? Nevertheless, he, like other Greeks, gave undue prominence to deduction in his theory of knowledge. We shall agree that Mr Smith (say) is mortal and we may, loosely, say that we know this because we know that all men are mortal. But what we really know is not 'all men are mortal'; we know something rather like 'all men born more than one hundred and fifty years ago' are mortal, and so are almost all men born more than one hundred years ago. This is our reason for thinking Mr Smith will die. But this argument is an induction, not a deduction. It has less cogency than a deduction, and yields only a probability, not a certainty; but on the other hand it gives *new* knowledge, which deduction does not. All the important inferences outside logic and pure

> mathematics are inductive; the only exceptions are law and theology, each of which derives its first principles from an unquestionable text, viz. the statute books or the scriptures.

It should be noted that even in mathematics, real proofs are not really checked, let alone designed, using FOL. Establishing a proof is typically by vague intuition or by mental leaps; checking it is by a 'consensus of the qualified' (see Davis and Hersh, 1981). The mechanization of proofs on high speed parallel computers may well extend the domains in which formal proof is practicable - but real problems take a lot of logic. It requires 362 pages to show that $1 + 1 = 2$ in Principia Mathematica.

2.5.2 Implications and Modal Logic

FOL is an *extensional theory*; it is described in terms of models which are sets (the extension of a predicate is the set of entities which make it true). Many of the ideas one may wish to express in knowledge representation involve *intension*: the qualities implied by a concept, rather than the set of objects it describes. An example from Sowa (1984):

All unicorns are cows
$\forall$ x unicorn(x) $\rightarrow$ cow(x)

Any model of the real world makes this implication TRUE, as there are no unicorns, whereas even in the real world we know that unicorns are mammals with one horn and certainly not cows. It was mentioned in section 2.2.1 that the FOL definition of implication (known as material implication or the 'Philonian' conditional) does not represent the normal 'causal' interpretation. This (again from Sowa, 1984) would be TRUE in FOL:

If elephants have wings then $2 + 2 = 5$

There are logics which are designed to overcome some of these representational deficiencies; the most well known are a series of 'modal' logics originally presented by Lewis (1932). They were initially based on a new form of implication ('strict implication') where

$$p \Rightarrow q$$

was true only if q 'could be deduced' from p. This concept is unrelated to the truth or falsity of p; it is saying that if p *were* true, then q would follow. This gives rise to 'modalities' of 'possibility' and 'necessity' that can be attached to predicates (see section 2.6.2).

2.5.3 Non-monotonicity and Defaults

'Monotonicity' is a feature of FOL which has come to be seen as a deficiency of FOL both in representation and reasoning, and which has been responsible for perhaps the greatest amount of work in logic in AI. If we have a domain defined by a set of logical axioms (a theory), then any additional axiom must be consistent with the original theory. Otherwise the whole system breaks down - anything can be proved. Having said:

$\forall$ x bird(x) $\rightarrow$ flies(x)
bird(ostrich)

we cannot then add:

$\neg$ flies(ostrich)

A new axiom cannot invalidate any of the previous conclusions. This property does not map onto the real world. The fact that 'Janet has no children' may be true now, false next week. An axiom that 'all birds fly' may be established in good faith before an exception is found. If someone parks his car outside, he will later reason on the assumption that it is still there - until he finds it has been towed away.

One can of course accept that a logical theory applies to a situation as exists (or is believed) at a certain point, and when anything changes all the necessary axioms change and one starts again. There are 'truth maintenance' systems which attempt to keep track of valid inferences in a changing situation. However, default assumptions do seem to be a necessary and basic part of common-sense reasoning and, this being so, any representation should itself include the fact that they are being made.

Changes in time can be incorporated by having an additional 'time' parameter in every predicate, although this in fact gives rise to need for many more default assumptions of exactly the above type (see section 2.6.1). Many 'temporal' logics are being designed to try and cope with the problem of time (see Chapter 9).

Section 2.6.1 discusses 'non-monotonic' logics which have been developed in an attempt to handle the representation of, and reasoning with, defaults.

2.5.4 Truth and Falsehood

Classical logics are based on the concept of a proposition which is either TRUE or FALSE. Propositions in the real world are not like that. There are degrees of uncertainty, degrees of judgement to be made, and these will be reflected in the inferences that can be drawn. From the statement that 'a man had large feet', one can make inferences about his shoe size, but they will be of the form 'he is almost certainly at least size 10, probably 11'. A representation which is constrained to truth or falsehood is not flexible enough to deal with much of the vagueness of the real world.

There are many schemes for representing and reasoning with uncertainty but many of them could probably not be described as 'logics' under our informal definition in section 2.2.1. Multi-valued logics have been developed within FOL - changing the semantics to allow values other than TRUE and FALSE in an interpretation. The most well known is a more radical departure known as 'fuzzy' logic where a predicate can take any real value between zero and one. Fuzzy logic was developed by Zadeh (e.g. 1974) and has influenced many expert system developments.

Some 'uncertain' logics are based on probability, some on numerical 'weights' whose interpretation in the real world may be as vague as the concepts they are expressing. Others represent uncertainty in some qualitative way. The success of these systems is only possible to assess in real life applications.

2.5.5 Reference to Predicates and Propositions

In FOL there is no way of referring to propositions or predicate names in other propositions.

One requirement is to quantify over predicate names. An example is a definition of 'equality'. FOL is often defined with equality as a special additional 'built-in' predicate. If it is not, one can define some of the properties, for example the reflexive property of equality:

$$\forall\ x,y\ \ \text{equals}(x,y) \rightarrow \text{equals}(y,x)$$

But what about substitution:

$$\forall\ x,y\ \ p(x) \wedge \text{equals}(x,y) \rightarrow p(y)$$

We wish to say that if p(x) is true and y=x, then p(y) is true *no matter what* the predicate p is. We cannot put a ∀ p on the front in FOL. This would be a statement of *second-order* logic. The statement: 'there is a set of people in this department whose members do not talk to anyone else' refers to the existence of a *set* with certain properties and is thus a second order statement. Second and higher order logics are well defined but the proof theories do not have the nice properties of completeness etc.

Perhaps a more common requirement in knowledge representation is the need to represent statements about *propositions*. One class arises from the modal logic mentioned previously:

It is necessary that ... some proposition.
It is possible that ... some proposition.

There are many circumstances in which propositions need to be referenced and many other logics which have been and are being developed to try and cope with them. Examples are:

X knows that ...	(epistemic logic)
It was true that ...	(temporal logic)
It was always true that ...	
It is permissible that ..	(deontic logic)

It might be thought superficially that such expressions can easily be incorporated in FOL by expressing the modalities by a predicate. Thus one may represent 'John knows the proposition P' by:

knows(John,P)

But we wish simultaneously to analyse P as a proposition; one may want to say:

knows(John,P) → P
∀ x knows(John,Q(x))

and so on. Making propositions into objects is not defined in FOL and its interpretation would become paradoxical. Modal logics introduce additional notation for 'modalities' like 'necessary that', 'knows that' etc., together with new rules of inference and semantics - a short discussion is given in section 2.6.2.

2.6 Non-standard Logics

Section 2.5 mentioned some of the many non-standard logics on which work is being pursued vigorously and which have found application in AI as well as other areas of computer science. Turner (1984) and several chapters of Frost (1986) survey advanced logics in this field. We discuss here two flavours of non-standard logic which have had the greatest influence in knowledge representation - non-monotonic and modal logics. Section 9.3.4 gives some discussion of temporal logic.

2.6.1 Non-monotonic Logic

As mentioned in section 2.5.3, non-monotonic logics have developed in an attempt to deal with representation and reasoning using default assumptions - which appear to be a ubiquitous characteristic of 'common-sense reasoning'. There are various formulations - notable are McDermott and Doyle (1980), McCarthy (1980) and Reiter (1985). Although the theoretical bases are very different, the difficulties are very similar and can be explained easily, if in an entirely non-formal way.

We wish to incorporate the idea of:

P normally-implies Q

i.e. if P is true we want to assume Q unless for some reason we *know* Q is not true.

Thus:

$\forall$ x (bird(x) normally-implies flies(x))

If bird(ostrich) we wish to assume flies(ostrich) unless we can *deduce* that an ostrich cannot fly, for example there may be a statement:

$\neg$ flies(ostrich)

Such a concept is not so easy to formalize in FOL. One obvious reason is that finding out that $\neg$ flies(ostrich) is not the case may not be decidable. A much worse problem is that having introduced such an abnormal implication we have to decide whether the inferences we make using it are allowed to be used in determining whether further abnormal implications can be made. An example with a simple model makes this clear:

1. P(gingko)	a gingko is a conifer
2. R(gingko)	a gingko is broad-leaved
3. $\forall$ x P(x) normally-implies Q(x)	conifers are evergreen
4. $\forall$ x R(x) normally-implies $\neg$ Q(x)	broad-leaves are deciduous

P(gingko) and R(gingko) are given and not much that is useful can be deduced with FOL reasoning. But since we cannot prove $\neg$ Q(gingko), we can assume (by 3) that Q(gingko). This presumably blocks the application of 4. But if we start again, we might choose to make 4 the abnormal inference we start with, thus proving $\neg$ Q(gingko). Whether we prove Q(gingko) or R(gingko) depends on whether our reasoning starts with 3 or 4. This is totally against the philosophy of logic where the set of theorems is not dependent on the proof procedure. Non-monotonic logics get around this problem by defining a theorem as that which is common to all the theories - i.e. is always proved. In the above example nothing new could be proved, but if 4 was not present, Q(gingko) would be provable.

Hanks and McDermott (1986) claim that in practice such logic is likely to be very limited. Their example is very instructive as it not only demonstrates this problem, but shows how non-monotonic argument naturally arises when arguing about time due to the 'frame' problem (nothing to do with frames). The example is thus worth presenting here, although in a much simplified notation.

Their problem concerns John and a gun which can be loaded or shot, and a succession of states. States change when something happens. Predicates are:

alive(s)	John is alive in state s
loaded(s)	The gun is loaded in state s

and functions:

result(load,s)	the state which results if the gun is loaded in state s
result(shoot,s)	the state which results if the gun is shot in state s
result(wait,s)	the state which result if we wait for a minute in state s

The axioms are:

alive(s0) John is alive at s0 (1)

$\forall$ s loaded(result(load,s)) (2)
If someone loads the gun when in any state s it becomes loaded in state result(load,s).

$\forall$ s loaded(s) $\rightarrow \neg$ alive(result(shoot,s)) (3)
If someone shoots a gun when it is loaded, John is not alive in the resulting state

One cannot deduce very much without 'frame' axioms. We wish to say that, for example if the gun is loaded at state s, it is loaded at whatever the next state is, unless we can prove otherwise. Otherwise, for every possible change of state in the world (for example someone walking into the room), we will have to define that it does not change the effect on the loading of the gun. This is the same as the 'car-park' assumption mentioned previously - the car is there unless there is some statement that it has been moved. The exact formulation of the non-monotonic axioms depends on which of the logics is adopted; in our informal notation we will say:

$\forall$ s,a loaded(s) normally-implies loaded(result(a,s)) (4)
$\forall$ s,a alive(s) normally-implies alive(result(a,s)) (5)

Guns stay loaded and people stay alive unless we can prove otherwise.

Suppose: result(load,s0) is denoted by s1
result(wait,s1) is denoted by s2
result(shoot,s2) is denoted by s3.

We can deduce loaded(s1) from (2). We cannot prove directly loaded(s2) but by the frame axiom (4) we can assume it since we cannot prove ¬loaded(s2). Now from (3) with s = s2, we see John is *not* alive at s3. This is all consistent and expected.

But even for such a simple problem, there is another solution, obtained by doing the proof in a non-intuitive direction. Since we cannot prove directly ¬ alive(s1), we can assume (by 5) alive(s2). A further application shows alive(s3). Then (3) shows that the gun was not in the state 'loaded' at s2. This may seem strange as it was loaded at s1, but everything is consistent. The only 'abnormal' inferences common to both scenarios are alive(s1) and alive(s2). This small example appears to show that non-monotonic logics as normally defined are not likely to be very useful.

2.6.2 Modal Logics and Possible Worlds

Many modern logics being studied in Knowledge Representation are derived from 'modal logic' developed by Lewis (1932). Hughes and Cresswell (1968) is a standard contemporary work. Modal logic was originally based on a concept of 'strict implication' (see section 2.5.2). Just as a language and theory of FOL can exist independently of its semantics, modal logic did not have a generally accepted model theory until Kripke formulated the 'possible world' semantics. This gave not only a precise model of various formulations of modal logic; it seems to be one which is relevant to the world that we wish to represent in an 'intelligent agent'. A logic of Knowledge and Belief, originally due to Hintakka, can be defined as an extension to the same semantics and this is formulated by Moore (1985b) - one of the most prominent attempts to apply modal logic in AI.

The basic notions introduced in Lewis' Modal Logic were 'necessity', 'impossibility', 'contingency', and 'possibility'. Any of these can be expressed in terms of the others so in fact only one need be defined as primitive - generally this is either 'possibility' or 'necessity'. Intuitively, 'necessity' is interpreted as 'could not fail to be true'. This obviously has no place in FOL where under an interpretation a proposition is either true or false; there is no other quality about it. In modal logic an interpretation embodies a parallel set of scenarios (see below) which could (for example) correspond to different hypotheses about the world.

The apparatus of modality is added to existing FOL. Thus a proposition p can still be asserted (and correspond to TRUE under interpretation) but one can also assert:

Lp

saying 'it is necessary that P'; and

Mp

saying 'it is possible that P'. The relationship between them is that p is possible if and only if it is not necessary that $\neg$ p:

Mp is equivalent to $\neg L(\neg p)$

The strict implication (which was originally the primitive from which modal logic was defined) is normally defined as:

$p \Rightarrow q$ is equivalent to $\neg M(p \text{ and } \neg q)$

i.e it is not possible that p should be true without q being true.

Additional rules of inference or axiom schema must be added to the normal FOL ones in order that deductive reasoning using these new modalities can take place. All modal logics would contain the following:

$LP \rightarrow P$

(i.e. if P - which can be any well-formed formula - is necessary, it is true) and:

$L(P \rightarrow Q) \rightarrow (LP \rightarrow LQ)$

Also any logical axioms (these are valid sentences in FOL) are necessary:

$L(p \rightarrow p)$ would be an example

A variety of modal logics can be defined by adding other rules of inference, of greater or lesser intuitive meaning, to the basic system (which is known as system T). For example:

$Lp \rightarrow LLp$ if p is necessary, then it is necessarily necessary

Quantification gives more room for varieties of axiomatic system, for example:

$\forall x\, Lp(x) \rightarrow L(\forall x\, p(x))$

The semantics of modal logic (without which the whole theory may seem rather obscure) can be regarded as an extension of that for FOL. Rather than having a single domain, an interpretation of a modal logic theory specifies a *set* of domains (possible worlds), of which one is distinguished - the *actual* world. This (say W0) acts as the model of the theory in the FOL sense, i.e. the non-modal propositions of the theory are interpreted in this world. An *accessibility* relation is defined over the possible worlds:

$r(W1,W2)$

defines W2 to be *accessible* from W1. Intuitively this means that an agent in

W1 can imagine a world as described in W2. The relation r is normally reflexive, and there may be other constraints on it (transitive, symmetric etc.) - it is the type of these constraints that precisely determines which of the various classes of modal logic is defined.

In each possible world various assignments may be made to the values of propositions just as with FOL interpretations (a complication is that the domains in possible worlds may be different also). An interpretation is a model of Lp just if p is TRUE in all worlds accessible from W0 (intuitively in all scenarios that someone in W0 can imagine). It is a model of Mp if p is TRUE in at least one accessible world.

By extension, the interpretation is a model of LLp if Lp is TRUE in all worlds accessible from W0, i.e. p is TRUE in all worlds accessible from these.

Moore (1985b), following Hintakka, developed a modal theory of knowledge.

Kap

means agent a knows proposition p. If a is kept fixed this becomes equivalent to the 'necessity' modality. (This means that another and perhaps better intuitive interpretation of Lp in ordinary modal logic could be 'I know that'; and the axiomatic structure may be much easier to follow.) The semantics is as above with the additional feature that an accessibility relation must be defined for each agent. A model of an agent's knowing a proposition must make that proposition TRUE in all the worlds accessible from W0 *for that agent*. Moore expresses the semantics itself in FOL; thus a formal translation can be made from the modal logic of knowledge into FOL, proofs undertaken and a translation made back.

2.7 Summary Comments

Logic is now widely studied not just by philosophers and mathematicians but by computer scientists and AI workers; the number of new logics being proposed is too great even to reference in this short chapter. The debate mentioned at the beginning of this chapter is but a continuation of argument about logic in the representation of knowledge which has continued for over two thousand years.

It might be thought that the development of mathematically sound semantics not just for FOL but (for example) for quantified modal logics, should have lessened the arguments, as much of the field becomes a part of uncontroversial mathematics rather than controversial philosophy. However this would be a mistake; the arguments are not about the soundness of models but about the extent to which they represent the world that we wish to reason about.

This remains to some extent a philosophical question and it is not clear whether the argument will ever be settled. But whatever the eventual role of logic, a firm grounding in it would be advisable for any student of knowledge representation, if only to understand and analyse the numerous other recondite representation schemes which will no doubt emerge.

3 Semantic Networks

Damian Mac Randal

3.1 Introduction

The study of language is usually divided into four fields: phonology, syntax, semantics and pragmatics. Phonology investigates the mapping of words onto sound. Syntax addresses the ordering of words and speech parts, often involving a grammar which specifies the criteria for acceptable sentences. Semantics is the study of the meaning of the individual concepts used in the language. Pragmatics maps these meanings and the other aspects of language onto the speaker's intentions behind an utterance (e.g. when crying out "the house is on fire" to someone standing in it, the intention is for them to leave, although this is not stated in the exclamation). The study of semantics is therefore an attempt to describe word meanings (and the usage of words where their meaning is ambiguous) and the conditions under which such meanings can interact to be compatible with the other aspects of a language. It is such a description which semantic networks were designed to provide.

A network is a net or graph of nodes joined by links. The nodes in a semantic network usually represent concepts or meanings (e.g. BOOK, GREEN) and the links usually represent relations (e.g., a book IS COLOURED green). Networks of this type not only capture definitions of concepts but also inherently provide links to other concepts. A large number of semantic networks have been developed as variations on this simple pattern. Some of these networks have been proposed as models of human memory and meaning representation, while others are used as components of language understanding and reasoning systems. The psychological validity of semantic networks will be discussed in Chapter 6. The development of semantic networks for computational purposes will be described here.

The origins of semantic networks lie in Aristotle's associationism (behaviour is controlled totally by associations learned between concepts) and reductionism (concepts are built of more elementary concepts; e.g. "bachelor" is built from "unmarried" and "man"). Much later associationism was extended and refined by philosophers and psychologists. Around 1869 James Mills showed that the use of a single concept term to refer to any occurrences of a concept leads to an ambiguity if that concept arose more than once (e.g. a representation of BOOK would not distinguish between "John's book" and "Mary's book"; one representation is required for each book). However, he did not specify the currently dominant solution to this problem of distinguishing between "types" (e.g. the concept BOOK) and tokens (individual occurrences of the concept; there are four tokens for "book" in the previous sentence, although "book" is just one type). Thomas Brown (*c.* 1820) contributed the notion of labelling links with semantic information (e.g. the book BELONGS TO John) instead of just giving them associative force (e.g. the book IS ASSOCIATED WITH John; the bird IS ASSOCIATED WITH green). Otto Selz in 1926 further added to the complexity of semantic networks by suggesting that paths between nodes across the network could be used for reasoning. All of these ideas were taken up by Quillian (1966) who proposed the first major computer system using semantic networks.

Since Quillian (1966) a large number of semantic networks have been proposed which share few features in common. A recent review of networks (Johnson-Laird, Herrmann and Chaffin, 1984) could only discern four assumptions which are common to the networks reviewed. These were:

(1) Network theories are designed to elucidate relations between concepts (intensional relations), in particular between the meanings of words. They embody no general principles concerning the relation between the concepts and the real world objects (extensional relations).

(2) A corollary of this: semantic networks are constructed on the assumption that intensional relations can be considered independently from extensional ones.

(3) Network theories are based on a formalism containing three components: a parser, a semantic representation consisting of a network of links between nodes, and a set of interpretative processes that operate on the network.

(4) There is a general commitment to parsimony.

Since there are so few assumptions which a set of semantic networks share, it is necessary to describe the details of several systems in order to investigate the differences and follow the direction of developments in semantic networks.

3.2 Outline

In section 3.3, two of the earliest and most influential systems, Quillian's and Winston's, are described in some detail. As the inaugural systems in the field, they tackled most of the basic components of semantic networks and supplied the framework upon which a lot of later systems were built. In section 3.4, the work on *case frames* leading up to Schank's conceptual dependency is described. Section 3.5 describes the work carried out in the mid to late 70's on the epistemological and logical basis for semantic network representations of knowledge. This work placed the whole field of semantic networks on a much sounder theoretical base and led to the development of the KL-ONE system. This is, perhaps, the most influential semantic network system that has been produced and has been the foundation upon which a lot of current research in Knowledge Representation is based.

Although in section 3.6 the quintessential features of some later systems are described, by this time, the mid eighties, single knowledge representations were not capable of handling the variety of knowledge structures that had to be manipulated. As a result, hybrid systems, using multiple representations, started to appear, and the work on semantic networks became more enmeshed with that on other representations. One such hybrid system is described in more detail in Chapter 10.

3.3 Early Developments

Ross Quillian is generally acknowledged to have been the first to apply the semantic network ideas in the AI field or, more specifically, in the field of natural language translation/understanding. In his PhD thesis in 1966, the central theme was "What sort of representational format can permit the *meanings* of words to be stored, so that humanlike use of these meanings is possible?". Towards this end, he proposed an associational model of human memory which he called *Semantic Memory* (Quillian, 1968). His idea was to capture the "objective" meanings of words in an encoding scheme of sufficient power to reflect the structure and capabilities of human memory, but simple and uniform enough to be implemented in a computer. These two goals conflict; implementation considerations require a small number of simple node and link types, while the representation of human knowledge, even "objective" knowledge, requires a complexity of representation approaching that of English itself. Figure 1 shows a fragment of Quillian's semantic memory, corresponding to two meanings of the word PLANT, which will be used to illustrate his model. Each meaning is defined in a "unit" bounded by a dotted line.

The other main, independent, example of the use of semantic networks was Winston's work on *Structured Descriptions* (Winston, 1975). He was working in the field of machine learning and was concerned with the learning-from-example of concepts behind common structures which he took from the blocks world, for example, pedestals, arches, tents, etc. built from rectangular blocks and wedges. The classical example used is his structured description of an arch, shown in Figure 2.

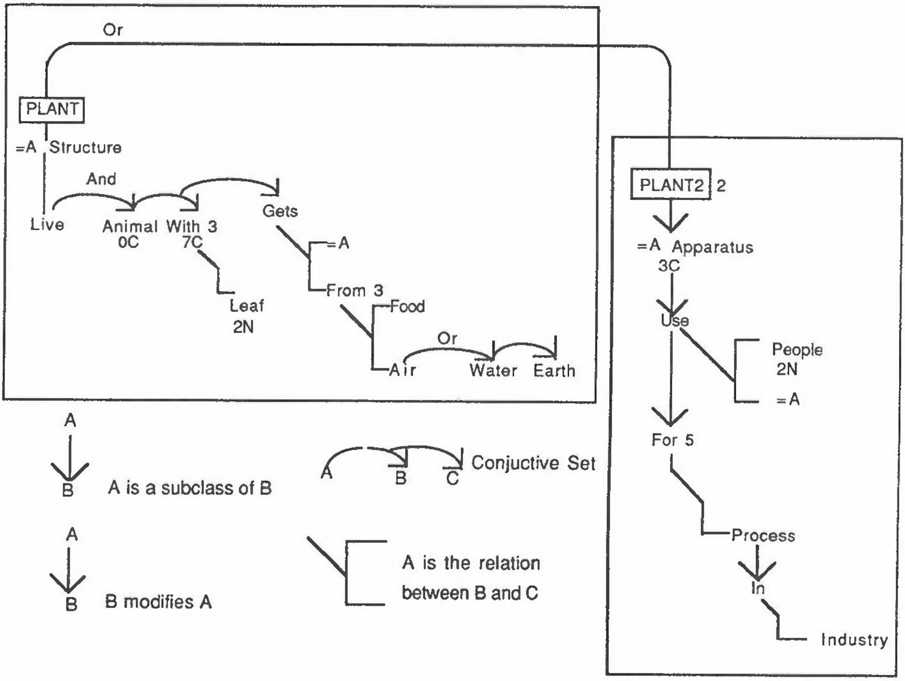


Figure 1 Quillian - Fragment of semantic memory for the word "PLANT"

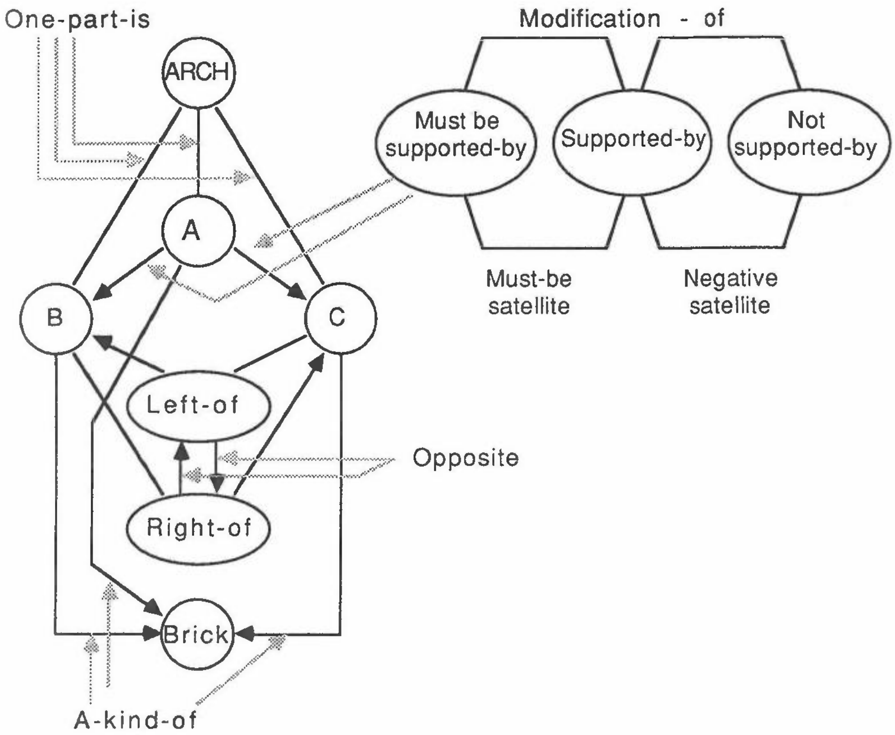


Figure 2 Winston - Structural description of an ARCH

The main part of Winston's work was on scene understanding and generalization from multiple instances. However, he recognized that the representation of knowledge, or scene description, as he called it, was the crucial part of the program. Like Quillian, Winston tried to develop a knowledge representation that was similar to the way humans apparently represented the concepts. The motivation for this was so that the examples and counter examples that would seem most natural for teaching these concepts to humans could be used to provide suitable input for his program.

3.3.1 Nodes

Quillian's model consisted of a mass of nodes, interconnected by different kinds of associative links. Each node basically corresponds to an English *Word Concept*, and represents the meaning of this "word" either directly or indirectly. For direct representation of meaning, a *type* node is used. These can be considered as the definition of the Word Concept, having associative links to other nodes (words) which *define* its meaning. Obviously, there is one, and only one, type node for each Word Concept in the model. For example, the two words contained in boxes in Figure 1, i.e. that appear at the top of a definition, are type nodes.

In contrast, a *token* node is used simply as part of a type node definition. In Figure 1, all words not in a box represent token nodes. The reason for introducing these token nodes, instead of using the type node itself, becomes apparent if a dictionary definition of one word in terms of other words ("tokens") is considered.

Dictionary definitions include sentences where words are ordered so that the syntax of the sentence shows the relationships between them. It would not be sufficient to present the words in a random order since their relations would not be known. Similarly, in a definition expressed in a semantic net it is necessary to have links between the nodes for the concepts. However, if these links existed between the only representation of each word in a system it would be impossible to follow any single definition since they would all involve links through the same words. Consequently, the definition of the type "plant" is a collection of nodes (e.g. "live") with links between them, where the nodes are copies of the node when it itself is defined. As in the paper dictionary, the word "live" in the definition "plant" is a copy (it looks the same) as the word "live" that heads its own definition. These copies of defined nodes are termed "tokens". In general for each type node there will be many token nodes scattered throughout the model.

The explicit distinction between type and token nodes was one of the important aspects of Quillian's system. In his later work on the Teachable Language Comprehender (TLC), he eliminated the explicit copying of type nodes to token nodes and used pointers (a type of link meaning "a copy of" rather than "is associated to") to the type node instead. The conceptual distinction remains while a reduction is gained in storage space. He also adapted the idea of Word Concept node to introduce "attribute values" to denote the strength of properties rather than simple association or its absence. This provided a mechanism for handling negation (an attribute with a value of 0) and quantification. For example in Figure 1, the token node "ANIMAL" in the definition of "PLANT 1" has a value 0C, underneath it. This indicates that the attribute ANIMAL is to be applied with a value (or precisely C for Criteriality) of "not at all" (0), i.e. the structure is not an animal. This introduction of values to properties was accompanied by a third important modification, the introduction of property inheritance. This mechanism allows a type node to inherit properties from superclass nodes in definitions. That is, if "PLANT" has a superclass of "LIVE", then it inherits the properties of "LIVE". This use of inheritance further reduces storage and is an illustration of the commitment to parsimony in semantic nets.

Quillian's work was directly taken up by J. R. Carbonell (1970), who used it to represent geographical knowledge for his computer aided instruction system, SCHOLAR. This provided students with a mixed initiative question and answer interface to a database about South America.

While building on the ideas in TLC, Carbonell introduced two further refinements to Quillian's idea of a node. Firstly, he drew a distinction between *Concept* nodes (e.g. latitude) and *Example* nodes (e.g. Argentina). This, of course, is the basis of instantiation. Secondly, he allowed Lisp functions to be attached to nodes to work out (infer?) properties that were not explicitly stated. This facility is the basis of the slot daemons used in frame systems.

Winston, in his system, also had two basic types of nodes. The first were nodes for those concepts corresponding to the physical objects in the scene. These were organized as a hierarchy, so that most of these nodes had a collection of other nodes as constituents. For example, a brick would be represented as a node, as would each of the faces that defined it.

The other type of node that Winston used was for concepts that corresponded to relationships that existed in the scene between the physical objects in the example being considered. For example, in an arch, the top block must be supported by the uprights, so the relationship "supported-by" was represented as a node. The reasoning behind this was that the nodes alone should contain all the information extracted from the scene. This simplified the comparison of different scenes for points of similarity or dissimilarity.

One consequence of the use of nodes to represent relationships was the ability to create new nodes (or "satellite nodes") to represent new relationships derived from old relationships. Since a node representing "supported-by" could be used in the representation of a counter-example to a concept, the concept learned from this ought to include the relation "not-supported-by". Therefore a negation of a relationship should be derived creating a new node. Similarly, if all the examples contained a relationship, then the learned concept should contain some modal necessity for this relationship (e.g. "must-be-supported-by"). Therefore a necessity modification had to be added to a relationship, creating a new node. Thus from the basic concept "supported-by" would be created a small collection of related nodes.

3.3.2 Links

Without offering any justification other than that they were needed to cope with the complexity of English definitions (but see below), Quillian introduced a number of different kinds of associational links. One of the most important link types is his *Special* link, where a token node "points" to its type node. For example, in the definition of PLANT 1, a link, drawn as a dashed line, points from the token node FOOD to the type node for this word concept. These Special links form the backbone of the Semantic Memory structure, linking related knowledge fragments into a graph. All other links occur inside a type node definition, and fall into one of six categories.

Four of these are:

> *Subclass links*, used to indicate that the type node is a subclass of another word concept (which, of course, is represented in the definition by a token node). For example, in Figure 1, PLANT 2 is a subclass of APPARATUS. This link permits the construction of taxonometric hierarchy.
>
> *Modification links*, used to show that the word concept represented by a particular token node is modified by the presence of another word concept, e.g. the concept APPARATUS is modified by the requirements of the USE structure.
>
> *Disjunction/conjunction links*, to indicate that two or more token nodes must, or must not, be applied at the same time. For example, food can be obtained from AIR, WATER or EARTH, so these token nodes are connected by a multi-arc link labelled "or", while a plant is a structure satisfying the token nodes LIVE, WITH leaves "and" not ANIMAL.
>
> *Relationship links*, used when a token node actually describes a relationship that must hold between two other token nodes. For example, the USE concept relates the user, PEOPLE to an object "= A".

One of the reasons for the poor performance of Quillian's later program, Teachable Language Comprehender, was that it did not take the semantic meaning of the links into account. Later Carbonell, in his system, introduced the idea of labelling the links.

In Winston's system, each link denotes a particular relationship between two Concepts. As there are many different relationships possible between Concepts, there are many varieties of link. However, not all relationships are represented as links, some being represented as Concept nodes. This creates two basic types of links, those conventional links which are themselves the relationship, e.g. in Figure 2, the "one-part-is" link, and those which are just associations between the object nodes and a relationship node which relates the objects, e.g. the "supported-by" link. This hints at the later separation, more fully developed by the case frame advocates, of relationships into syntactic ones (i.e. the grammar of the sentence) and semantic ones (i.e. the words of the sentence). Unfortunately, Winston is rather inconsistent about link types, the "supported-by" relationship sometimes being shown by a simple link, sometimes, as in Figure 2, by a Concept node. The only apparent criterion used to decide which link type to use is whether the relationship will be required to discriminate between the given examples and counter-examples. It can be seen in Figure 2 that the labelled links refer to relationships between nodes, while the relationship nodes refer to relationships between the concepts the nodes represent.

3.3.3 Discussion

Quillian's *Semantic Memory* model introduced, in some form or another, nearly all the important aspects of semantic networks. His model is based very heavily on the organization and layout long used by dictionary and thesaurus compilers, and he claimed to be able to express *anything* that could be expressed in natural language. Though in a sense this is true, the problem is that the concept definitions rely very heavily on the reader's human intuition, obviously not available to a program, as to the meaning of the nodes and, particularly, the links. Also, he recognized the apparent conflict between the associative and schema memory models, but claimed that the two could be handled in parallel by a sufficiently sophisticated program.

Quillian's notion of units contains the basis of a concept hierarchy, complete with an inheritance mechanism. Unfortunately, the notation is not sufficiently rich to distinguish between the different epistemological levels of the concepts that are represented by the same type of nodes and links. For example, the same interchangeable node type is used for a class, an instance, an event and a relation. The other interesting feature of Quillian's units was the later definition of a concept by a property (attribute/value) list.

Though Winston's semantic network was relatively successful for his purposes, it shares a lot of the failings of Quillian's. It contains both the idea of a concept hierarchy, related by "has-part" and "kind-of" links, and the distinction between class and instance nodes, although via the same "kind-of" link used between classes. It also demonstrates quite clearly that the representations of relationships as typed links and as Concept nodes are interchangeable, though at the expense of notational obscurity and computational complexity,

However, like Quillian's semantic memory, it also fails to identify the vital distinction between the links used as part of the representation, e.g. the "kind-of" link used to build the concept hierarchy, and the domain specific links, e.g. "supported-by". This lack of distinction rather obscures the concept hierarchy and forces the application, here a learn-by-example program, to have this information built in.

3.4 Linguistic Influences: Case Grammar and Conceptual Dependency

Whereas Quillian and Winston based their knowledge representations on psychological models of memory, other representations have been developed based on models developed in linguistics. One of the most influential models has been that of Case grammar, originally developed by Fillmore (1966) in the light of the questioned validity of the relations of "subject" and "object" found in the influential linguistic text by Chomsky (1965: 63-73). In Fillmore's grammar "case relations" were semantic relation primitives linking verb (and some other) structures to the nominal elements of sentences. The relations for each verb can therefore be specified by a set of cases. The set of cases which characterize a verb is called the "case frame" for that verb.

Fillmore originally suggested six case relations (Agentive, Instrumental, Objective, Dative, Factitive and Locative) as a "set of universal, presumably innate, concepts". However, it has been a recurring problem for Case grammarians to define a comfortable set of cases, and even Fillmore himself allowed the number and nature of cases to grow. A recently proposed set (Sparck-Jones and Boguraev, 1987) involves 28 cases.

A large number of successful computational systems have been developed which incorporate some aspects of Case grammar (e.g. the general purpose language front-end of Somers and Johnson (1979); Marcus' (1980) English parser; Binot *et al.'s* (1980) French parser; van Bakel and Hoogeboom's (1981) Dutch parser; Nash-Webber's (1975) speech understanding system; the medical expert system of Kulikowski and Weiss, 1971; Bobrow and Winograd's (1977) KRL). Simmons (1973) was the first to develop a semantic network using the set of cases as the set of possible link types. This provided a firm theoretical foundation and a clearly specified semantics for each link type, rather than choosing them on an *ad hoc* basis. In his system verbs were represented by nodes and the case links connected them to nodes for other concepts in order to represent sentences.

In contrast to these language oriented systems are others which claim to capture some deeper cognitive aspect (e.g. the long term memory model of Rumelhart and Norman, 1973; Norman *et al.*, 1975; and more deeply, Schank's (1972) "Conceptual Dependency"). Conceptual Dependency is different from other case-like systems since it is intended to be a language-free representation of concepts whereas the others are language dependent. Consequently both the set of cases used to describe relations and the representation chosen for actions and objects had to use language-free semantic primitives.

3.4.1 Conceptual Dependencies

Schank's (1972) conceptual dependency captures the underlying meaning of utterances as "conceptualizations" by reducing them to combinations of primitive "predicates" chosen from a set of twelve "actions" plus state and change of state, together with the primitive "causation", and seven role relations or "conceptual cases".

Schank attempted to express all verbs as some combination of his primitive actions. These included TRANS (transfer of possession); INGEST (the taking in of an object by an animal); PTRANS (the transfer of physical location of an object) and ATRANS (the transfer of an abstract relationship such as possession, ownership and control). It can clearly be seen from this subset of actions that a large number of verbs can be built up from them given the appropriate relations. However, some of these appear weak when considered. For example, walk can be defined as: PTRANS of x by x through MOVEing the feet of x in the direction of y.

Surprisingly, the cases in conceptual dependency are no more primitive than those of case systems which are more surface oriented. They are Object (in a state), Object (change of state), Object (of action), Actor, Recipient/Donor, From/To, and Instrument. This selection of high level cases suggests that their nature may vary depending on which predicate they are attached to (for example, is there a difference in Actor of "dance" and Actor of "hit"?). However, since there is such a small set of primitive acts, even if this were so, there would only be 37 relations, which is comparable to the number in some other Case systems. Each case is given a graphic representation designed to make illustrations of the semantic nets more readable. Figure 3 shows an example of a conceptual dependency incorporating primitive acts, conceptual cases (using this graphical representation) and objects to describe a state in which "Joe is drinking some soup with a spoon".

Schank's proposals are unsurprisingly inadequate to fulfil his objective of a universal language-free conceptual reasoning system. Conceptual dependency has been criticized by linguists since it is not a theory of language (although it was not intended to be); by psychologists since it requires an unrealistically precise definition of concepts and provides no mechanism to analyse pragmatics; by logicians since it does not capture the relative scope of existential and universal quantifiers; and by computer scientists because it is not described exactly enough to be understood and implemented.

Despite these failings conceptual dependency is important for two reasons. Firstly, it enabled the development of an inference engine for Schank's memory structures which was able to handle a much larger and wider range of language than any of the earlier systems. Secondly, it was the first major attempt to derive both the conceptual nodes and the relational links in a system from an abstract theoretical position.

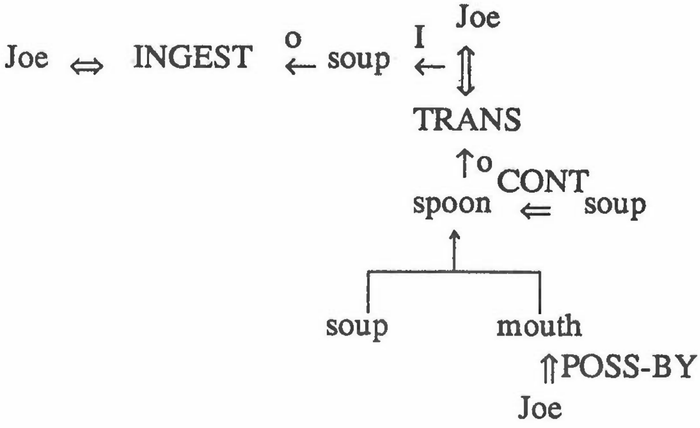


Figure 3 Schank - A conceptual dependency

There have been many attempts in formal semantics to define sets of semantic primitives which can be used to build the concepts with which we reason (e.g. Wierzbicka, 1972; Miller and Johnson-Laird, 1976) and attempts such as Case grammar to define the relations that hold between concepts. Conceptual dependency brought these together with a computational approach to knowledge representation and reasoning into a single system. It showed that we do not have an adequate theoretical knowledge to develop systems capable of universal reasoning, but it illustrates a methodology which can be applied in a structured manner to smaller domains.

3.4.2 Discussion

One of the things that made case structures so important in semantic networks was the success they had in natural language understanding. The impression is given, however, that a great deal of this was because the cases were chosen with great care to match the language style, the number and scope of the concepts were quite restricted, and the inference engines using the representations were hand crafted to suit the particular domain.

On the other hand, their development firmly shifted the emphasis of semantic networks away from the associative memory mould of the earlier work. One aspect of this shift is the move away from what Brachman calls the implementational level of semantic networks, where the network is interpreted simply as a data structure, to the conceptual level, where the links have a well-defined semantic content representing conceptual relationships. Another aspect was the attention they paid to the structure of the knowledge they were trying to represent, as opposed to the structure of the domain containing the knowledge.

3.5 Theoretical Underpinning

Nearly all systems developed before 1975, including those described above, and several after that date, generalized very badly from their test examples to real world situations. In most cases, it was the attempt to build natural language understanding systems that prompted the use of semantic networks, mainly because they fit so neatly onto the way humans verbalize their knowledge. As the subject matter grew more complex, the notation became less tightly defined and the more it was left to the user, or the application program, to ensure that the correct interpretation of the notation was made. As a consequence, the simple network formalism was extended to handle knowledge from a particular domain, or a particular subset of natural language, without much thought being given to the semantics of the structure used to represent the knowledge. This had several consequences, besides making the applications very brittle (for example, Quillian's Teachable Language Comprehender only worked on a handful of sentences).

The obvious problems were the logical and expressive inadequacy of most of the proposed notations. Though this could be, and was, tackled in an incremental manner, a large part of the difficulty was due to the lack of understanding of the semantics of the semantic network itself. These problems were tackled by a number of people during the mid to late 1970s, in particular Woods, Schubert and Brachman. They, among others, started addressing the epistemological issues raised by these knowledge representations and laying the foundations for an adequate theory of semantic networks. In this section, the work which led to semantic networks being placed on a sound footing will be described, together with the context in which it was carried out.

3.5.1 The Semantics of Semantic Networks

Woods (1975) was really the first to tackle head-on the major epistemological problems that beset earlier semantic networks. He strongly challenged the logical adequacy of previous notations, focusing on the need for care in the choice of conventions for representing facts as semantic networks, and on the need for an explicit definition of the meaning of the links and arcs used.

Firstly, however, Woods tried to clarify the meaning of the word "semantics". He identified in the literature three independent and conflicting usages of the term, covering the translation of natural language into a formal representation of its meaning(s), the meaning (truth value) of the formal representation, and the procedures that operate on the formal representation.

Woods himself holds the view that all three stages are necessary, and had earlier described such a mechanism based on *procedural semantics*, i.e. where the semantics of an entity is defined by the procedures that operate on it. He pointed out two common misconceptions of "semantics", firstly in extending the term to cover the retrieval and inference mechanisms of the semantic network, secondly, at the other extreme, in denying a fundamental distinction between syntax and semantics.

Having defined "semantics", Woods went on to examine "semantic networks". His target was a formal notation which will accurately and unambiguously represent any (humanly) possible interpretation of a natural language sentence. This he referred to as the *logical adequacy* of the semantic representation. As well as this, he required that the representation facilitate the translation from natural language and the subsequent use of the knowledge by an inference engine. One task he places outside semantic networks is the reduction of all equivalent propositions to their canonical form, i.e. the conversion of all sentences with the same meaning to the same internal form, even though this could resolve paraphrases without a combinatoric search. This is partly because he believes it impossible for full natural language, but mainly because he believes that normally paraphrase is not one of full logical behaviour, but only of logical implication in one direction. For example consider the two phrases, "he is my mother's brother" and "he is my uncle", where the first implies the second but not *vice versa*. Hence, the mechanism for searching for equivalent propositions is still required. Of course, on efficiency grounds, a certain amount of canonicalization might still be beneficial.

3.5.2 What's in a Link

As well as the semantics of the compete network, it is necessary to have a clear idea of the semantics of the components of the network, i.e. of the links and nodes themselves. One characteristic of the early semantic network systems was that a lot of the "meaning" of the links depended on the user's intuitive understanding of the labels on the link. Of course, given a user who was not totally familiar with the representation scheme, or attach the representation to an automated retrieval/deduction system, and the whole edifice collapses. Several people had considered this before Woods addressed it in his 1975 paper.

One of the earlier efforts to remedy these problems was the semantic network model developed by Shapiro (1971) to act as the database for a question answering machine. Although, as usual, nodes represented conceptual entities and the links the relations that held between them, he insisted, for obvious pragmatic reasons, that anything about which information can be

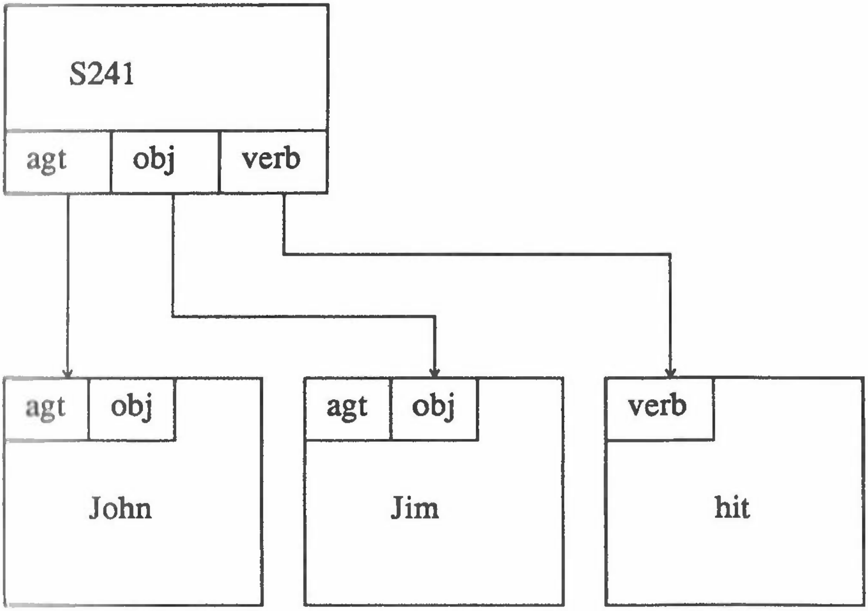


Figure 4 Shapiro - system relations for sentence components

given or questions asked, had to be represented by an item, i.e. a node. For example, in Figure 4, the statement S241 "John hit Jim" can be believed, discussed, and otherwise referred to, and therefore it must be represented as an item. Thus most relationships between items may also be required to be held as items. Shapiro claimed that eventually, if this requirement were adhered to rigorously, there would be some relations that were not conceptual but were merely used by the system to tie a fact-like item to the terms taking part in it. These could be represented as system relations, i.e. links.

Shapiro's system relations were not part of the semantics of the domain, but purely a part of the knowledge representation structure. This separation of the epistemological structure from the semantic structure was a major step forward towards networks that were logically and semantically sound. It was also a belated recognition that epistemology - the study of knowledge and the methods used in that study - should be distinguished from the subject of study. At the epistemological level, provision was made for different types of system relations. Unfortunately, though his examples show several different types, mainly linguistic cases such as object, agent, verb, etc., he does not discuss the semantics of a system relation, or what properties the set of system relations would have. In his quest for generality, he tries not to restrict the number or type of system relations, leaving it to the application developer in the hope that many different semantic networks could be built using his system.

Woods considered the links themselves from a more philosophical viewpoint and identified two different types of link, structural and assertional. An assertional link establishes a relationship between two existing nodes, while a structural link is one which exists only to provide meaning to a node, For example, in Figure 5, the link from "John" to "Mary" connects two nodes which have an existence beyond the concept of the link "hit" and the link makes an "assertion" about a relationship between them. However, the node "S1234" only exists as a focal point for the links "VERB", "AGENT", "RECIP", etc., and the sole function of these links is to provide the "structural" support for the node "S1234" (which otherwise would not exist). This is just the distinction that Shapiro made but his solution was to eliminate assertional links, representing them as relational nodes, rather than complicate the network by mixing the epistemological and semantic structure. Of course, this just relocated the problem in the semantic interpretation of the node, once again confusing semantics with epistemology.

At this point, it is worth jumping forward to Brachman's paper (Brachman, 1983) on taxonometric links in semantic networks. One of the major problems in earlier nets was the confusion they allowed between the subclass type of link and the instantiation type of link. This occurred mainly because in English both may be represented by the pseudo-word IS-A. For example "Tweety is a canary" and "A canary is a bird". Since these links form the taxonometric backbone of any semantic network and provide the inheritance mechanism that is the *raison d'être* of most implementations, it is important to have a clear understanding of the epistemological role of these links.

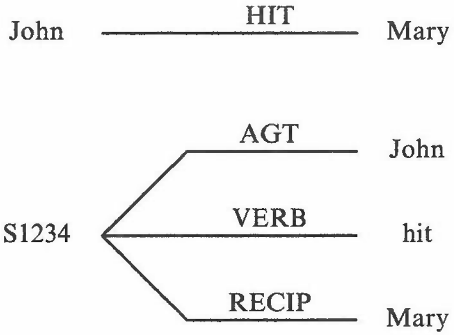


Figure 5 Woods - Assertional vs Structural links

Also, in order to consider the relationship of semantic networks to other knowledge representations, especially first order predicate logic, it is necessary that the function, or functions, of the IS-A link be clearly defined.

Brachman firstly divides IS-A links into two groups according to the type of participating nodes, either *generic*, i.e. a description applicable to many individuals, or *individual*, i.e. a description or representation of one individual. He then goes on to enumerate the different meanings that IS-A takes with these node types, including subset, superset and set membership, generalization, specialization and instantiation, etc. The more common of these are shown Table 1 below.

From this, Brachman identifies two basic types of IS-A, those that take one concept and form another out of it, and those that convey some sort of information about the relation between two sets, or between the arguments of two predicates. The latter category needs to specify: its *assertional force*, i.e. whether it is a statement of fact or not; its *modality*, i.e. whether it is part of the definition of the concept; its *quantifier*, whether it is universally quantified, i.e. always true, or just a default, i.e. true unless cancelled; its *content*, usually a set inclusion/membership or material conditional (if..then..)/predication.

He points out in passing that a number of these requirements, for example modalities and defaults, raise severe difficulties for standard predicate logics.

3.5.3 What's in a Concept

Having teased out a number of insights relating to the semantics of a link, the next step is obviously to tackle nodes. Like links, the early semantic networks relied on the user's intuition for the correct specification and use of nodes. They had been used to represent "facts", "events", "classes",

Generic	**to similar Generic**	**to Individual**
Set	**Subset**	**Member**
Predicate	**Univ. Material conditional**	**Predication**
Structured description	**Conceptual containment**	**Description (falls under)**
Prototype	**Sharing typical property**	**Similarity to prototype**
Role	—	**Specifies filler**
Predicate	—	**Abstraction**
Generic	**to different Generic**	**gives**
Set	*prototype/predicate*	*Characteristic of set*
Role	*prototype/predicate*	*Constraint on filler*

Table 1 Brachman - Summary of IS-A link flavours

"predicates", "relations" and even "meaning of sentences". Usually they are represented as groups of features, but again the structure of the grouping, and even of the features, left a lot to the imagination.

Woods, once again taking a deeper philosophical approach, pointed out the intensional nature of a lot of the concepts that semantic networks have to represent. The classic example is Frege's Morning star/Evening star, two phrases with different meanings (intensions) but denoting the same planet (extension). Handling extension (or denotation), i.e. representing the set of objects satisfying the concept, is straightforward, but can be computationally infeasible. Handling intension (or meaning), i.e. the concept itself, which may or may not be true of a particular entity, is more difficult. For example, as shown in Figure 6, representing the sentence "John's height is 1.82m" is straightforward: a link Height between an extensional node John and an extensional node 1.82m. If the sentence "John's height is greater than Sue" is added, it becomes clear that an intensional node representing John's_height is required. The main problem that intensional nodes raise is how the program distinguishes between the two types of node, ensuring the correct type is created when building the semantic network, and that inferences on these nodes are performed correctly. The solution Woods proposed was to make all nodes intensional, and add a specific predicate of existence where necessary. This also solves the problem of nodes which represent concepts which do not or cannot have a real world instantiation.

There is also the need to distinguish those links to an intensional node that are part of its definition, e.g. the Height link to John in Figure 6, and those which are assertional, e.g. the Is link to 1.82m. Woods introduced the idea of an "EGO" link, which was used by nodes to indicate their defining links. Thus, following the EGO link from a node such as John would get the information "I'm the guy whose name is John Smith, who works down

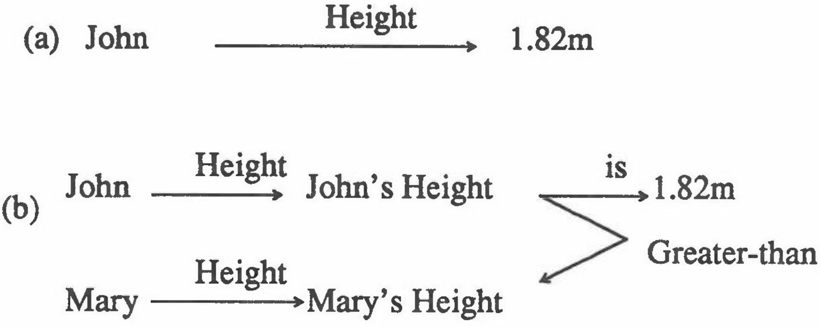


Figure 6 Woods - intensional nodes

the corridor, etc.", whereas the EGO link from John's height would give "the height of the John described by that node over there".

Later, Brachman (1977) addressed the epistemology of *Concept nodes*. In a similar vein to Woods' division of nodes into extensional and intensional categories, he divided nodes into two main types, those which represent particular things in the world, and those which represent a class of particular things. Examples of the former are nodes that represent *objects*, e.g. John, *factual assertions*, e.g. "John's height is 1.82m" or *events*, e.g. "John hit Mary". Nodes representing a class usually have links to instances of the class and to subclasses/superclasses. Unfortunately, this notion of class is frequently extended to try to capture the *Concept* behind the class, i.e. what it means to be a member of this class, as well as representing the set of class members. This is normally achieved by considering the node to be a collection of properties or predicates that somehow "define" the concept desired.

Apart from re-emphasizing the problems discussed by Shapiro, i.e. the distinction between structural and relational links, and Woods, i.e. the distinction between intensional and extensional nodes and between assertional and descriptional links, Brachman pointed out several other sources of confusion connected with nodes. Firstly, unlike properties of an individual which refer to the individual itself, properties of a class node refer to the *members* of the class, and not to the class itself. Of course, a way of talking about the class as a class is still required. A second problem is that the value of a property can either specify a particular value that holds for every member of the class, for example, "elephants are grey", or specify a class of values of which one must hold, for example, "John's height is greater-than 1.82m".

Brachman's solution to these problems was to remove the definitional properties of the Concept node to separate nodes called *role descriptors*. The role descriptor then holds all the information about the function of this definitional property, such as what class the property values belong to, the number of instances of this property that this Concept can have, etc. It also acts as a prototype for the instantiation of the appropriate part of the Concept. As well as specifying the properties, or Roles, the relationships between the properties are specified in a *Structural Description*. Thus, concept nodes could be defined in terms of other nodes, c.f. Quillian's planes, Schank's case frames, etc., by means of an organized collection of structured links. These ideas are described in more detail in section 3.6.2 on KL-ONE.

3.5.4 Expressiveness of the Notation

Woods' main criticism of extant semantic networks was their logical inadequacy, that is, their inability to express precisely, formally and unambiguously all the interpretations that a human listener would place on a sentence. He was most concerned with the rather *ad hoc* way that quantification was handled.

Schubert (1976) addressed the problems that semantic networks have with logical connectives, quantifiers, and modal operators. He approached this from the viewpoint of Predicate Calculus, which he considers to be almost isomorphic with semantic networks. Firstly, he developed a propositional notation in which the basic unit of information is the atomic proposition. This consists of a *propositional* node, a mandatory PRED link to a *predicate* node and links to the concept nodes serving as arguments of the predicate. This is simplified by replacing the propositional and predicate nodes with the predicate name. Figure 7 shows, for monadic and triadic predicates, both an atomic proposition and its simplified version, together with its predicate calculus equivalent. One of the more interesting features is that it is unnecessary to coerce monadic predicates into dyadic form, so that this need for an IS-A link is eliminated; at least in Brachman's *generic* sense.

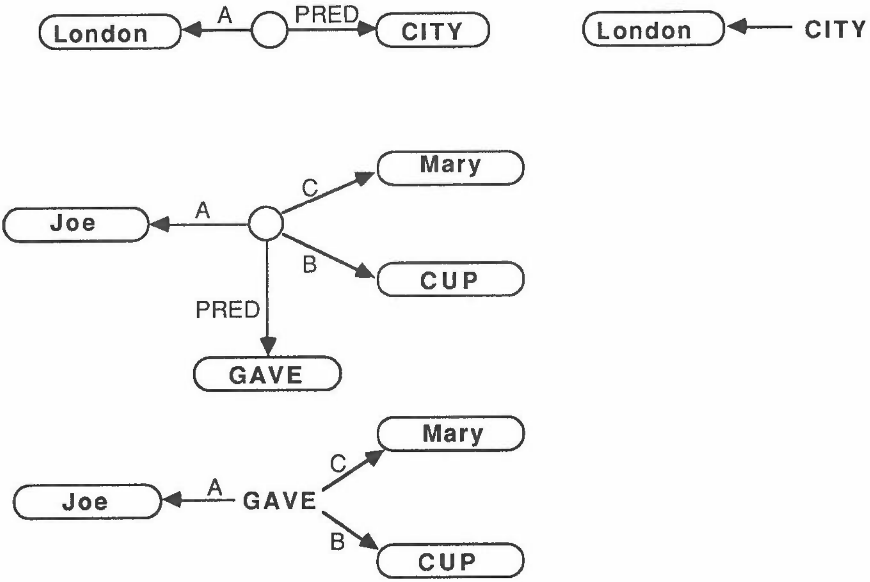


Figure 7 Schubert - Atomic propositions

Schubert also compared his notation to Schank's conceptual dependencies (see section 3.4.1). Although, as a semantic network, conceptual dependencies can mostly be translated into Schubert's notation, he claims that case-structured action propositions lack expressive power and are anyway at a semantically higher level than necessary, and are therefore not primitive.

Quantification

Woods discussed quantification, linguistic (definite and indefinite) as well as numerical and logical (universal and existential). He pointed out the need, not only for definite entities, which exist in the real world, and indefinite entities, which do not have to exist, but also for definite and indefinite variable entities, whose instantiation depends on the instantiation of those other entities to which they are related. For example, the system has to be able to deal with sentences such as "Every boy loves his dog" where "dog" is a definite variable entity whose instantiation exists but is different for each instantiation of "boy", and such as "Every boy needs a dog", where the node "dog" is an indefinite variable entity.

Woods also brought up the problems of numerical quantification using the sentence "three men saw two boats". He insisted that the three possible interpretations of this sentence had to be representable separately in the semantic network. Obviously, explicitly representing this with three nodes for the men and two for the boats, though acceptable here, is not generally possible, e.g. "50 million frenchmen ...". He also brought up the problem with universal quantification of ensuring that future nodes of the same type were created with the correct links. For example, after processing the sentence "Every boy has a dog", all new "boy" nodes must have a "has a dog" proposition added to them.

Woods proposed three methods for dealing with quantification. Firstly, quantifiers were added explicitly as higher order operators, represented in the network as a special node with assertional links to the quantification type, range, variable and proposition. This adds several extra indirections to the representation of the three men/two boats sentence, to hold the set of three men with a "for-all" link, etc. This is the method used by Shapiro (see above). Secondly, a standard resolution theorem proving technique was used to remove all existentially quantified variables from the expression, leaving all remaining variables universally quantified. This is reversible, so there is no loss of information. The advantage of this in a semantic network is that it is only required to indicate which nodes are universally quantified. The problem with it is that, whilst the process is reversible, it is not easy. Thirdly, a quantifier can be converted into a relation between the set of instances of the quantified variable and a predicate containing the rest of the proposition. For example, "All men are mortal" is converted into a relation between a set (all men) and a predicate (mortal). This can also be applied to existential quantification. Putting this into the network would require the predicate to be held as a special type of node that has to have a link to a "set" node.

Schubert in his system also addressed the issue of quantification, and came up with a scheme very similar to the second one above. One further enhancement, however, is that time is handled as quantification over the *moments* at which the proposition holds. A number of notational abbreviations were introduced to simplify the network and make knowledge input easier.

3.5.5 Discussion

The importance of Woods' paper was that it focused attention on the epistemology of semantic networks. Although the paper was rather negative, concentrating on pointing out what was wrong with existing systems, and the new ideas he put forward were rather weak, he did have a major impact on the evolution of semantic networks. By challenging the logical adequacy of previous systems, and clearly identifying the problems, he changed the development of semantic networks from a series of implementations of assorted psychological models of human memory into a serious knowledge representation methodology with an emerging strong theoretical foundation. The questions he raised were the focus for most of the work on semantic networks over the following few years.

Schubert provided a strong, predicate calculus-based foundation for semantic networks, which to a large extent satisfies Woods' point about logical adequacy. However, he does not give much help in representing knowledge in terms of his pseudo predicate calculus. Schubert himself admits this, and when referring to the problems with conceptual dependencies he suggests that higher order constructs, along the lines of the case structure, may be needed to handle real natural language. This contains the seeds of Brachman's five-level structure for semantic networks.

3.5.6 Inheritance and Defaults

Before moving on to discuss the KL-ONE system, there is one other issue that was raised and which had a major influence on the evolution of semantic networks. This is the whole matter of inheritance and defaults, which have an important role in the frame structures developed by Minsky. These are dealt with in greater detail in Chapter 4, but as they have been shown to be useful, not only in handling subset/superset and part/whole relations, but also in controlling search and delineating contexts, their impact on semantic networks will be mentioned here.

Hayes (1977b), developed a higher level structure, along the lines of a frame, but built and interlinked with a more conventional semantic network. This was one of the earlier attempts to merge the frame representation and semantic networks. These structures, called *depictions*, are really just subsets of a larger semantic network, but with a definite, generic head node, the *depictee*, which is connected to the rest of the conceptual hierarchy via IS-A links. Inside the depiction is a collection of other generic nodes called *depicters*, which are related to the depictee by PART-OF or CONNECTED links. Figure 8 shows a typical depiction, with the depictees being the solid nodes and the dashed lines indicating the extent of the depictions. Links leaving the depiction are "inside"; links entering are not. This allows the depiction to be viewed as an archetype, with the depictee universally

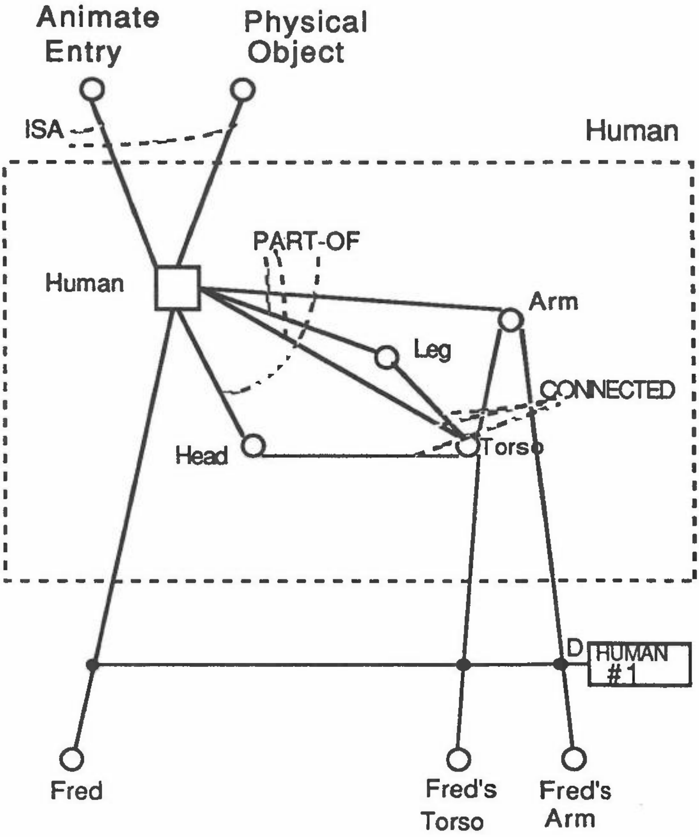


Figure 8 Hayes - An instantiation of a depiction of a human

quantified and the rest existentially quantified within its scope. Upon instantiation, a *binder*, e.g. the D-HUMAN#1 in Figure 8, creates an instance node bound to the depicters and inheriting their connectivity structure. Not all depicters need be instantiated: for example, there is not an instance for head in the Fred instance. If needed, these other depicters can be instantiated later.

The binder/depiction has a number of interesting features. It provides a context mechanism capable of handling referents and resolving ambiguities. For example, given an instantiation, Fred, of "human", "his head" would be instantiated as Fred's head, rather than the head of another person, of a hammer, etc.

Inheritance can be handled by making the depicters instantiations of depicters in a higher level depiction, e.g. dog-leg and human-leg are instances of creature-leg. Hayes recognized that these instantiations were fundamentally different from normal instantiation and had a different type of binder to handle them.

Inherited properties (links and nodes) can be cancelled or augmented without affecting the parent - the binder, in the child depiction, does the instantiation. Also, since the same node name can be re-used, search up through the hierarchy for defaults is fairly efficient.

Depicters can have numerical modifiers that specify the number of instance nodes that the binder can link to them, e.g. a human can have two arms represented by the same depictee. The different instances can optionally be modified by *distinguishers*, e.g. to distinguish right and left arms.

Depicter nodes can be defined further down the conceptual hierarchy by their own depiction. Thus the same node can act as a depicter and a depictee, i.e. it acts as a *role* specifier in the encompassing depiction and as an entity in its own right in its own defining depiction.

Since in some cases the role cannot exist without its depictee, e.g. an arm requires a human, SQN links are provided to ensure that all necessary superiors are instantiated when a subpart is instantiated.

The most important feature of Hayes' notation is his use of a frame-like construction to control instantiation. This, in common with most frame systems, provides a convenient, and easily controlled, mechanism for structuring knowledge, A second important feature is the distinction drawn between the use of a node in its own right to act as a prototype for an instance, and its use to specify a role that needs to be filled in order to create another prototype's instance. Hayes' attempt to capitalize on the work going on into related representations was one of the first examples of the recent trend towards hybrid systems.

Brachman (1983), in his paper on IS-A links, notes that one important use of IS-A in previous semantic networks was to give a default, i.e. a statement that holds unless explicitly cancelled. This is important because it permits exception handling, an essential for real world problems (this becomes clear if an attempt to define "elephant" is made), and was one of the reasons for the development of Frames. The difficulty with cancellable defaults is that they negate the usefulness of a concept hierarchy; for example if "Clyde IS-A elephant", then he has all the inherited elephant properties by default. However, if "Gerry" has all the elephant properties, it cannot be assumed that he is an elephant - he could be a giraffe with all giraffe properties cancelled and a few elephant-like properties added. Thus the node "elephant" does not represent the concept of an elephant any more, but merely acts as a placeholder for a bundle of typical elephant properties. A semantic network which permits the arbitrary use of cancellable defaults seriously jeopardizes its utility as a knowledge base for an inference engine.

3.6 An Epistemologically Adequate Semantic Network, KL-ONE

The work described in the last section highlighted a number of problems with the existing semantic networks. Although the problems were tackled and potential solutions suggested or demonstrated, this was predominantly carried out piecemeal and in isolation. Though they are all built on the same basic associational structure, the various network formalisms described above are all quite different from each other.

Following this period of exploration, Brachman produced his seminal paper identifying five independent levels at which semantic networks can be understood. This paper was a watershed in the development of semantic networks, providing an integrating framework in which the emerging ideas on logical adequacy and expressive power could be investigated. In it he gathered together under one overall umbrella most of the ideas and trends that had emerged from earlier systems. The epistemological framework was the basis of the knowledge representation used in the KL-ONE system. Since it first appeared, KL-ONE has been used in a number of applications, ranging from natural language understanding, to question answering systems, to the modelling of office automation.

In this section, firstly, Brachman's five-layer model will be examined, along with his proposed criteria for evaluating the "correctness" of semantic networks, and then the implementation of these ideas in the KL-ONE system will be described.

3.6.1 Conceptual Levels

Brachman, by examining the various representational primitives used, shows that these differences are a reflection of deeper and quite fundamental philosophical differences (Brachman, 1979). Four of Brachman's five layers concern the conceptual *levels* or viewpoints into which the various network primitives could be categorized. These levels are:

> *Implementational.* This level is the most basic form of semantic network, where links are merely pointers and nodes are merely destinations for links. At this level, the network is simply a data structure, with no real semantic content.
>
> *Logical.* A semantic network can be understood as a set of logical primitives with a structured index over those primitives. It bears a strong relationship to predicate calculus, with the extra feature of the network topology, and thus provides at least a basic method for factorizing and organizing knowledge. Nodes represent predicates and propositions and links represent the logical relationships between these nodes, such as "and", "subset", "there-exists" etc. This level deals with questions of logical adequacy, including quantification. It is best exemplified in the work of Schubert, and was also obviously influential in the work of Shapiro, Woods and Hendrix.
>
> *Conceptual.* At this level, the real semantics becomes very obvious, and the relationship with natural languages is strong. Nodes represent word-concepts, i.e. language independent object, action and event types, e.g. GRASP, INGEST, PTRANS etc., while links represent the *case structure*, out of which all expressible concepts can be constructed, e.g. AGENT, INSTRUMENT, RECIPIENT etc. This level deals with the issue of expressive power, in that the types of node and link provided dictate what language expressions can be handled. The champion of this approach is Schank, though it has been adopted by many others, for example, Rumelhart and Norman, Simmons, Rieger.
>
> *Linguistic.* The top level is really natural language itself. The example given by Brachman is the OWL system which, apart from the implementation level, has no structuring primitives other than those of English. The nodes in this scheme are words, and have context-dependent meanings, e.g. "fire" changes meaning when attached to "man", and the links represent real world relationships, e.g. Colour, Hit, etc.

In the above, each level is a self-contained network representation, more or less independent of the levels above and below it. By separating out the primitives like this, Brachman shows clearly that the primitives of different levels, e.g. there-exists, AGENT and Colour, are fundamentally and philosophically different from each other. This variety of primitive types had been implicitly recognized by the developers of some of the networks mentioned above, although it was not formalized as clearly as here. A lot of the problems that arose in the earlier networks were due to the mixing of primitives from several levels. The four level scheme described above does not quite cover all the notational features of previous semantic networks. For example, the binders of Hayes, and even the ubiquitous inheritance, are not logical primitives, and even though they are generally assumed by the case structure, they are not represented in it.

Brachman proposed introducing a fifth level, the *epistemological* level, to handle this formal structuring. The primitives of this level are for representing *knowledge-structures* and their interrelationships as knowledge-structures, independently of the knowledge contained within them. For example, an intensional entity has to be defined from lower level primitives, and has to be related as a unit to other entities, using epistemological links.

This paper has had a major impact, partly because it tied together a number of strands that were emerging in isolation, that is, it reconciled the work on logical adequacy which was driving towards a variant of formal logic, the conceptual dependency work which was driving towards more expressive power and the epistemological work that was starting to examine knowledge-structures as structures. A second reason for the impact was that the five level structure put forward was the basis for the KL-ONE system, and this, until recently, was the foundation for a large number of important semantic network research projects.

3.6.1.1 Criteria for Assessing Semantic Networks

Since each level represents a particular type of semantic network, Brachman explores the capabilities of the levels, and specifically the epistemological level, against the three criteria, *neutrality*, *adequacy* and *semantics*. By neutrality he means that each particular type of semantic network must not constrain the choice of primitives for the next level up. For example, the logical level must not contain features, such as inheritance links, that will affect the design or operation of the epistemological level. This offers the usual advantages of modularity. Nearly all previous semantic networks violated this criterion, and therefore not only were less flexible for building on top of, but were confusing to use. However, some of the logical networks, for example Schubert and Woods, go a long way towards this goal.

By adequacy he means that the each level must provide the facilities required to implement the next level up. For example, a conceptual level should be able to support any possible linguistic system of knowledge. Conceptual adequacy has been addressed in particular by Schank and Rieger. Logical adequacy has already been relatively successfully tackled by Woods and Schubert. The trend towards logical networks is partly explained by the extra difficulty in achieving adequacy in a mixed level network.

By semantics he means the provision of a formal specification of the *meaning* of each element and the operations that can be performed on them. Here, he considers the meaning of a primitive to be specified by the procedures that operate on it. For the logical level, if a mapping to predicate calculus is established, then the semantics is defined. At the conceptual level, Schank and Rieger have specified the inferencing operations for each primitive act. This is only possible since they have a fixed number of primitives. At the linguistic level, a formal semantic specification of natural language is next to impossible.

3.6.2 Overview of KL-ONE

Firstly, it should be pointed out that KL-ONE is more than just a representational language, as it includes facilities for the building, storing, querying etc. of the network. KL-ONE is an evolving system with new ideas constantly being added, and thus it is difficult to pin down. However, the main interest is in its capabilities to explicitly represent conceptual information as a *structured inheritance network*, a feature that has been fairly consistent over the various implementations. The system described here is a fairly recent implementation, and some of the elegance of Brachman's earlier ideas has been lost as new problem areas emerged (Brachman and Schmolze, 1985c). Although the various aspects of the system will be considered in some detail in later sections, an overview of the complete system will be given here.

KL-ONE is primarily an epistemological level network which provides the necessary primitives with which to describe and handle knowledge. The primitives are knowledge independent, in that they can be used to describe the internal structure of a broad spectrum of concepts. Briefly, the primitives used to represent the internal structure of a *Concept* are *Roles*, which represent the attributes associated with the Concept. A Role not only holds the information about the function of the attribute, that is, the intension of the attribute, but also acts as a description of the potential fillers, that is, the instances of the attribute. These Roles indicate the type and number of the instances permissible for this attribute. The interrelations between the Roles are handled by a *Structural Description*, which contains a set of relationships between the Roles that must hold between the Role fillers when the Concept, and hence the Roles, is instantiated.

Given this notion of a Concept, the epistemological level primitives therefore consist of the relationships between Concepts, Roles and Structural Descriptions, and the internal relationships of Roles and Structural Descriptions. As well as this, the relationships between two Concepts, and indeed between two Roles and between two Structural Descriptions, need to be addressed. These relationships are the basis of the inheritance mechanism, which, at its simplest, requires the Roles and Structural Descriptions of the parent Concept to be linked to the child Concept. Obviously, a mechanism must be provided to allow modification of the Role or Structural Description being inherited.

3.6.3 Concepts

KL-ONE *Concepts* correspond to conceptually primitive pieces of domain knowledge, and are either *primitive Concepts* or *defined Concepts*. Primitive Concepts are used for domain concepts that are atomic, i.e. have no internal structure, or that cannot be defined in terms of necessary and sufficient properties. However, primitive Concepts can still specify necessary properties, though they may not be able to define all of them. Defined Concepts are built up from primitive Concepts and other defined Concepts and have their necessary and sufficient properties defined. For example, the Generic Concept (see below) for a natural kind such as "elephant" cannot be defined by necessary and sufficient properties, so it is primitive. However, in Figure 9, the Generic Concept URGENT-MESSAGE is a defined Concept since it is completely defined in terms of REPLY-REQUESTED-MESSAGE and the "less than 1 hour" modification. Most Generic Concepts fall into the primitive category.

The most important type of KL-ONE concept is the *Generic Concept*, i.e. an intensional description of a class of domain objects, e.g. person, message, date etc. Generic Concepts are either primitive, or are defined, using *SuperC* links, in term of other Generic Concepts. This creates a basic taxonomy formed of those Concepts that *subsume*, or are subsumed by, other Concepts. The subsumption criterion allows multiple SuperConcepts, and the taxonomy is actually a lattice. By subsumption, Brachman meant that an instance of the lower Concept would always, by definition, be an instance of the higher Concept. Thus, a Concept gets its meaning from its SuperConcepts, possibly modified locally either by additional specific properties, or by restrictions on the SuperConcept's properties. As an example of this, in Figure 9 the Generic Concept URGENT-MESSAGE is subsumed by REPLY-REQUESTED-MESSAGE since it is completely defined by adding the local property "within one hour" to those properties of its SuperConcept, REPLY-REQUESTED-MESSAGE. KL-ONE provides a method of

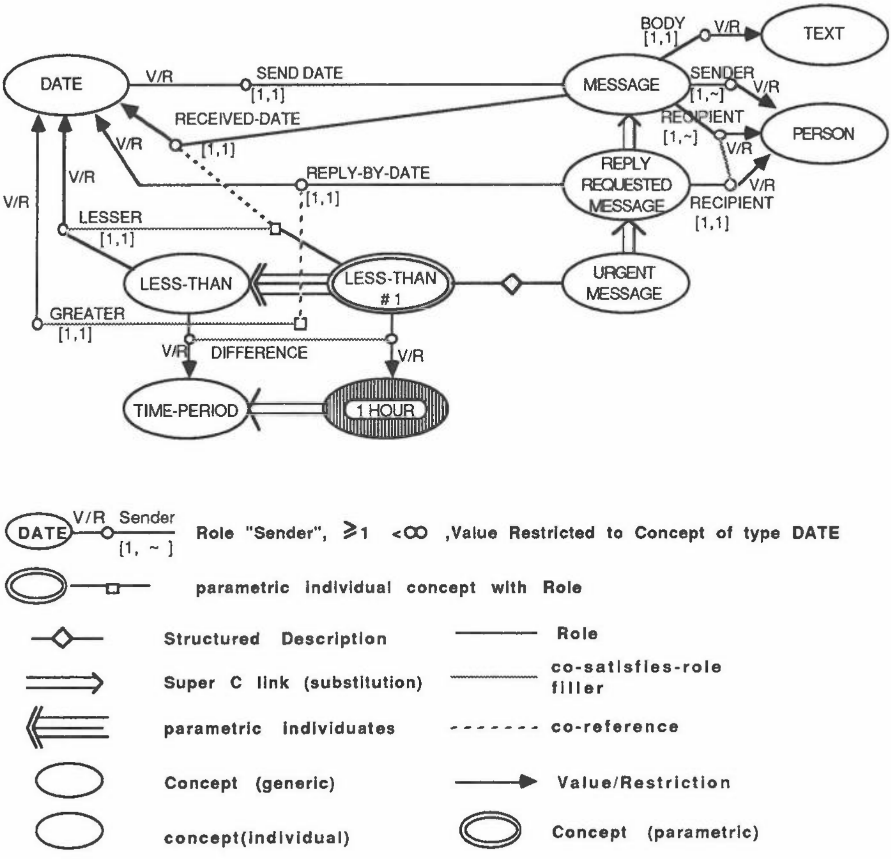


Figure 9 Brachman - Fragment of KL-ONE network

deciding if one Concept subsumes another, and this is the basis of the *Classifier*, which automatically places new Generic Concepts into their correct place in the taxonomy.

3.6.4 Roles

Before considering other types of Concept the internal structure of a Generic Concept will be discussed. The primitives used to represent the internal structure of a Concept are *Roles*, which represent the attributes associated with the Concept. Roles not only hold the information about the function of the attribute, i.e. the intension of the attribute, but also act as a description of the potential fillers, i.e. the extension, or instances, of the attribute. Since several different types of entities could satisfy these functional requirements, a set of Roles, called the *Roleset*, is needed to identify the different types of filler allowed for this attribute, e.g. the sender of a message could be a machine or a person. This identification is achieved by having a link, *Value/Restriction*, point to the Concepts that satisfy the functional requirements. Similarly, the function might permit multiple instances, for example, a message could have several recipients, so again the Role has to identify the number of instances allowed. As an example of this, in Figure 9 the Concept MESSAGE has a Role "Sender" whose Value/Restriction link is to the type PERSON and whose number is shown (under the Role symbol) as having a minimum of 1 and no maximum number of senders. Early versions of KL-ONE also allowed the Role to be optional, and provided a modality flag to indicate whether the role was a necessary part of the Concept, or was a derivable or optional attribute. However, as Brachman pointed out earlier, default cancellation destroys the logical adequacy of a semantic network, so in later versions Roles were restricted to necessary attributes of the Concept.

3.6.5 Structural Descriptions

As well as specifying the Roles that define a Concept, the relationships that must exist between the Role fillers have also to be specified. For example, the sent-date must be before the received-date, the recipient of a message might need to be the sender's supervisor etc. This is a function of the *Structural Descriptions*. These relate two or more Roles in terms of another Concept. For example, in Figure 9, an URGENT-MESSAGE is defined as a REPLY-REQUESTED-MESSAGE, with a Structural Description which relates the Reply-By-Date Role to the Received-Date Role via a particular version of the LESS-THAN Concept. This version is isomorphic with the Generic Concept LESS-THAN, but is a *parametric individual Concept*, parameterized by the URGENT-MESSAGE context. (There is also a shorthand notation for the common parameterized Concepts of equality and subset.) The required Roles of the LESS-THAN parametric Concept, identified by links to the corresponding Roles in the Generic Concept, are *co-referenced* to the Roles in the URGENT-MESSAGE Concept being related. Not all Roles of the Generic Concept have to be co-referenced, and can be instantiated in any appropriate fashion.

3.6.6 Individuation and Individual Concepts

All KL-ONE Concepts are intensional, so there are no Concepts to directly represent extensional objects, that is, objects in the real world. Individual objects are *denoted* by *Individual Concepts*, which are *individuations* of the appropriate Generic Concept. Brachman reserves the word *instantiation* for the association between the Generic Concept and the real world object. Individual Concepts individuate a specific Generic Concept, but describe at most one individual. As the Concept is intensional, there is no implication of existence of the individual being described. For each Role in the Generic concept, there is a Role filler in the Individual Concept. As well as matching Roles and Role fillers, the Individual Concepts pointed to by the Role fillers must accord with the relationships specified in the Structural Description.

3.6.7 Inheritance

Individuation can be considered as an example of one aspect of inheritance. In general, inheritance is the passing down of the properties of the parent Concept to the child Concept. To do this in KL-ONE requires not only an indication of the link between the parent and child Concept, that is the subsumption link described earlier, but also, for each Role in the SuperConcept, an indication of what, if any, restrictions are to apply. In the case where no restrictions apply, there is no Role shown for the child Concept, and the Role shown in the parent Concept is deemed to apply. If, as is usually the case, the child is to be a specialization of the parent, the Role in the child Concept has the new number of fillers or type of entity shown, along with a *restricts* link to the corresponding Role in the parent Concept. For example, the Concept REPLY_REQUESTED_MESSAGE inherits the Role Recipient from its SuperConcept MESSAGE, but restricts the number of Recipients to 1. Structural Descriptions, on the other hand, must be inherited intact.

Another controversial issue hinted at earlier was that of default values and cancellation. Brachman's stance, mentioned above, is that allowing cancellable defaults in the definitional, or description formation, aspect of a semantic network undermines the logical adequacy of the representation. He therefore insisted that Roles represent only necessary attributes of a Generic Concept, and thus, from his definition of subsumption, they are not cancellable. Non-necessary properties, which may need to be cancellable, are dealt with outside the taxonomy in the Assertion Language. The use of Reiter's default reasoning mechanism for this was suggested but no details of how this could be done were given.

3.6.8 The Conceptual Coat Rack

Brachman, and later Woods (1983), give an analysis of *procedural attachment* in KL-ONE, i.e. the mechanism by which the user of a semantic network can access the implementation code (interpreter) directly to attach a procedure to an entity. One reason for these procedural attachments is to represent metaknowledge, for example, about a Concept as an entity. This, called a *metahook*, is really a means for one level to perform functions of the next higher level. A second reason is to attach special interpreter code to an entity, for example, to short circuit, for efficiency, the normal code sequence the interpreter follows when handling a specific Role. This, called an *ihook*, is really a means for a level to modify the level below. (The hooks form a "coat rack" upon which to hang auxiliary knowledge.) Both these *escape* mechanisms are not philosophically necessary, and, if needed, demonstrate inadequacies in the semantic network, either at the current level or the one below (the interpreter). Brachman warns against abuse of these hooks.

3.6.9 Discussion

The main criticism of KL-ONE is the complexity of the Role and Structured Description, due in large part to the evolutionary nature of their development, which makes the system hard to use. However, there are also some more fundamental weaknesses which appeared in use. It should be noted that a number of these weaknesses are complementary, and the tradeoff between them necessarily fails to eliminate both, or either, of the problems.

The most serious of these is the incomplete treatment of Roles, in that they derived their semantics via other constructs, the Concept and its Structural Description. This lack of an adequate formalization led to the grafting on of such kludges as Rolesets, and even then the system could only cope with primitive Roles. Another shortcoming was the inability of the hierarchy to handle Concepts unless their necessary conditions could be specified. This arose from the desire to ensure that the representation, and especially the classifier, was demonstrably complete and sound, i.e. obtained all and only all the right results. However, the tradeoff for this was less expressive power, which did make the system less useful in practice. Finally, there was a lack of support for such things as representing exhaustion or exclusion among a Concept's subsumees or indication sequence in Role fillers. Most of these issues were tackled by one or another of the later systems that were built on or around KL-ONE, some of which are described in the next section.

3.7 Recent Systems

As well as acting as a representational language for application systems, KL-ONE has also been the foundation for some basic research in knowledge representation. KL-ONE was extended and refined over a number of years, with a number of new ideas being introduced. Most of these were application oriented and thus were more interested in being usable than in addressing the theoretical issues. One such system is NIKL, described below. One idea, that was raised but not fully developed in KL-ONE, was the separation of the description formation aspects of the knowledge representation from the assertion making aspects, and this led to the development of the KRYPTON system described in Chapter 10. In parallel with this work, several other unrelated systems were being produced. One example of this is the Conceptual Graph system of Sowa, which was biased towards the logical representations to about the same degree as KL-ONE was biased towards the schema representations. This is described in outline below and in detail in Chapter 7.

3.7.1 NIKL

NIKL (a New Implementation of KL-ONE) is one of the many offshoots from KL-ONE (Kaczmarek, 1986). As a new implementation, it follows its parent system fairly closely. However, as well as improved efficiency, there are some significant differences between the two systems and, interestingly, a number of similarities with KRYPTON. The major change is in the representation and use of Roles. Roles were now thought of as representations of conceptual Relations which are 2-place relations in the same way as Concepts are 1-place relations. They could then be organized in a separate taxonomy and given a domain and a range, i.e. restricted to a particular set of Relations and particular range of values of each Relation. For example, the Concept "parent" could have a Role "child" which is restricted to the Relations "daughter" or "son" and with a numerical range "> 0". This gives the advantages that the user, or system, can define and refer to Roles in an analogous manner to Concepts.

Another of NIKL's enhancements was the provision of better support for reasoning, in particular classification-based reasoning. Unlike KRYPTON, the emphasis was placed on efficiency, forgoing completeness in favour of expressiveness. Firstly, facilities were provided to allow the user to specify that a set of Concepts was disjoint, i.e. mutually exclusive in the real world, or covered another Concept, i.e. every extension of the covered Concept is described by at least one of the set of Concepts. Secondly, support for partial orderings of Roles is provided in the shape of Relations that allowed sequences to be described. For example, an initialization phase can be forced to come before the main phase, which in turn precedes the terminal phase. Thirdly, the ability was provided to specify, as a Relation, a set of Roles that are sufficient conditions for a Concept - the necessary conditions having been defined by the Concepts position in the taxonomy. In coping with these extra features, the classifier, which automatically classifies Concepts in terms of Concepts already existing in the Concept hierarchy, is actually carrying out quite sophisticated classification-based reasoning. This greatly reduces the load on the user specifying the Concept hierarchy as well as on the application program's reasoning ability.

3.7.2 Conceptual Graphs

One modern semantic network not based on KL-ONE is the Conceptual Graph system of Sowa (1984). Sowa was interested in natural language processing and his system reflects this, though it is based strongly on logic and was designed to support logical inference. A Conceptual Graph is a mini-semantic network representing a sentence. A Concept node represents entities, attributes, states or events, while a Relation shows how the Concepts are interconnected, i.e. the semantic relationship between two Concepts. Links have no meaning in themselves, other than to indicate the Concepts dealt with by each Relation. The Concepts and Relations have referents, i.e. refer to either a specific individual, an unspecified individual or a set of individuals. For further details see Chapter 7 of this book.

3.8 Conclusions

The various "extensions" that are currently being grafted onto semantic networks seem to indicate that they are not sufficient in themselves to be an adequate knowledge representation language, though they provide a powerful and flexible base on which more complex hybrid systems can be built. However, there is a trend for the more philosophically sound of these extensions to be subsumed into the network notation, e.g. Minsky's notion of schema was first tagged onto the network as KL-ONE Roles / Structured Descriptions and then in NIKL became part of the infrastructure connected with Relations. Certainly semantic networks appear to be a very intuitive representation, but this very intuitiveness can lead to logically unsound systems unless a lot of care is taken with the notation. In common with most other representations, the main outstanding problems are how to describe natural kinds, how to handle defaults and negation and how to deal with incomplete, or incorrect, information.

4 Structured Object Representation - Schemata and Frames

Gordon Ringland

4.1 Introduction

In this chapter we discuss the representation scheme called frames or schemata. Though this representation has been attacked as adding nothing really new to the tools of AI (Hayes, 1979) it remains widely popular both in practical applications and in research. The reason for this popularity lies in the fact that much knowledge has a structure, arising either out of the structure apparent in the domain to be represented, and/ or the structures we have to impose to be able to deal *usefully* with large amounts of knowledge. To the extent that structured object representations (afterwards called *frames* or *schemata*) can reflect the structure natural to given sets of knowledge then it will be advantageous to use them. Even in the cases where frames are logically equivalent to representation by randomly ordered sequences of clauses in first order logic, it does not follow that the readability and expressive power of the two representations are equivalent.

In the next section we discuss Minsky's original paper on frames (Minsky, 1975). The hope here is to give some help to those who will read this rather difficult paper, and to motivate the discussion.

In section 4.3 we present examples of the use of frames which bring out their usefulness in representing structure. Section 4.4 gives a presentation of the influential paper by Hayes (Hayes, 1979) which argues there is little new in the frame idea. Though we defer to the clarity and scope of the paper, our conclusion is that Hayes' claims are too strong. Section 4.5 reviews the most recent important contribution to the frames literature, that of Brachman (Brachman, 1985). Broadly speaking, Brachman shows that the use of frames for 'common-sense' reasoning is, if not impossible, at least fraught with traps for the unwary. In section 4.6, we review other approaches to default reasoning and note that the problem of 'common-sense' reasoning is still a major research issue and at the time of writing this problem cannot be definitely asserted to be solved.

4.2 Minsky's Paper (Minsky, 1975)

The notion of organizing perception into some kind of unitary whole dates back as far as Kant's Critique of Pure Reason, first published in 1781 (Kant, 1787) and is represented in this century by the work of Bartlett (Bartlett, 1932). Minsky squarely acknowledges his debt to Bartlett and observes that similar ideas were in the air in AI two or three years before the publication of the first version of his frame paper in 1975. However it cannot be denied that the paper by Minsky has had great influence on the enterprise of Knowledge Representation and is probably the most widely referenced contribution to the field. For this reason alone it would be right to devote a section to the paper, but it is also worth some discussion to allow us to compare Minsky's hopes with their realization.

The opening section of Minsky's 1975 paper captures much of what has been influential. He begins by asserting that most theoretical work on AI and in psychology has been too fine-grained, local and unstructured to account for effective common-sense thought. At this stage it is appropriate to make explicit a significant part of Minsky's argument. By linking work on AI and psychology he has staked out a definite and contentious position on AI: essentially that though artifact is unavoidable we should try to represent the real thing (human intelligence) as effectively as we can. For an extensive discussion of human knowledge representation see Chapter 6. This should be contrasted with McCarthy's position which emphasizes the A(RTIFICIAL) in AI and consequently makes psychological reality a subordinate or even irrelevant issue (Kolata, 1982).

Next, his opening emphasizes 'common-sense thought' as a process that AI must capture. There are two important issues here. The first is plain and should not be contentious, namely the necessity for any satisfactory AI system to display the sense and reasonability tests most humans apply most of the time as a part of their 'common sense'. For definiteness let us consider a simple case. Suppose you, the reader, a human intelligence, accept, along with a great many other things (the preceding qualification really is important), that P implies Q and also accept the antecedent P. Are you then forced to accept the consequence Q? Should you accept it? The answer "not necessarily" is a display of common-sense. Clearly you could have excellent reasons for believing not-Q, and if so you might cease to accept either the conditional P implies Q, or the antecedent P. The point here is that while rules of proof sanction the conclusion Q, the rules are local to the relation P implies Q, and the antecedent P, and apply to those individual syntactic forms. But the *overall* excellent reasons for not accepting Q may be global, involving reasoning and judgement over all 'the great many other things' you accept. This is the reason for Minsky's complaint against locality.

The second and less obvious point is the use of 'thought' in the desideratum 'common-sense thought'. I believe that using 'thought' rather than 'reasoning' in this term is significant and is an example of a confusion on the part of the anti-logicist school as exemplified by Minsky, and also the logicist school as represented by McCarthy and Hayes (Hayes, 1977a). We distinguished, quite deliberately, rules of proof as in logic from the more general 'reasoning'. There has been a tendency to equate the two terms, leading to a view that if one uses logic as a representation one is inevitably committed to a particular formal deductive machinery and, to avoid such a machinery, one must avoid logic as a representation. We shall return to this point in discussing Hayes' critique of frames (Hayes, 1979) in the following section.

Minsky asserts that "chunks" of reasoning and the representation of language memory and perception should be larger and more organized than, say, production rules, and frames are the device to provide the structure. This structured representation and the interaction of these structures is generally taken to be the essence of frames or schemata, and is surely the aspect of Minsky's essay which has had the most practical effect.

A simple example of a frame might be the frame for a domestic pet

FRAMENAME	PET
SLOT 1	DISPOSITION: FRIENDLY
SLOT 2	HAS: OWNER
SLOT 3	HAS: A HOME

Minsky's original idea is that the upper levels of the frame are fixed and represent unalterable truths about the object or situation, while lower levels consist of 'terminals' or 'slots' (the usual notation) which are filled with specific instances. In practice this distinction between upper and lower levels is not much used, except for the name of the frame itself.

No particular originality was claimed for the notion of frame and he acknowledges the parallel work of others in attempting to move away from representing knowledge as 'collections of separate simple fragments'. More novelty is claimed for the notion of 'frame systems', which are collections of related frames which are linked together by the sharing of slots. This sharing allows lower level frames to inherit the properties of higher level frames -

the implementation of this mechanism is not very explicitly discussed, but the idea is that the linkage mechanism is some sort of 'information retrieval network' using a 'matching' process.

Consider the two linked frames in Figure 1 to see what results are expected. Through the retrieval network the two frames are compiled to produce a new explicit frame for DOG shown at the bottom of the figure. Here the first three slots of the new frame for DOG are inherited from the MAMMAL frame via a matching of the frame name MAMMAL with the IS-A slot value MAMMAL.

Though some slots (higher level) are presumed to be inherited without exception, Minsky required that a frame's slots would usually be filled with 'default' and that these default assignments should easily be replaced by values which better represent the situation. This requirement arose from Minsky's view of cognitive memory and he believes that much of the power of the theory stems from these default assignments. He held that on considering a new situation a frame is selected from memory, and this remembered framework is adapted to the actual situation by changing details (slot values) as necessary. Indeed he asserts that this is the essence of the theory. The role of the frame in memory (and knowledge representation) is to represent stereotypical situations.

We illustrate this notion with the following very simple example of a stereotype of elephant:

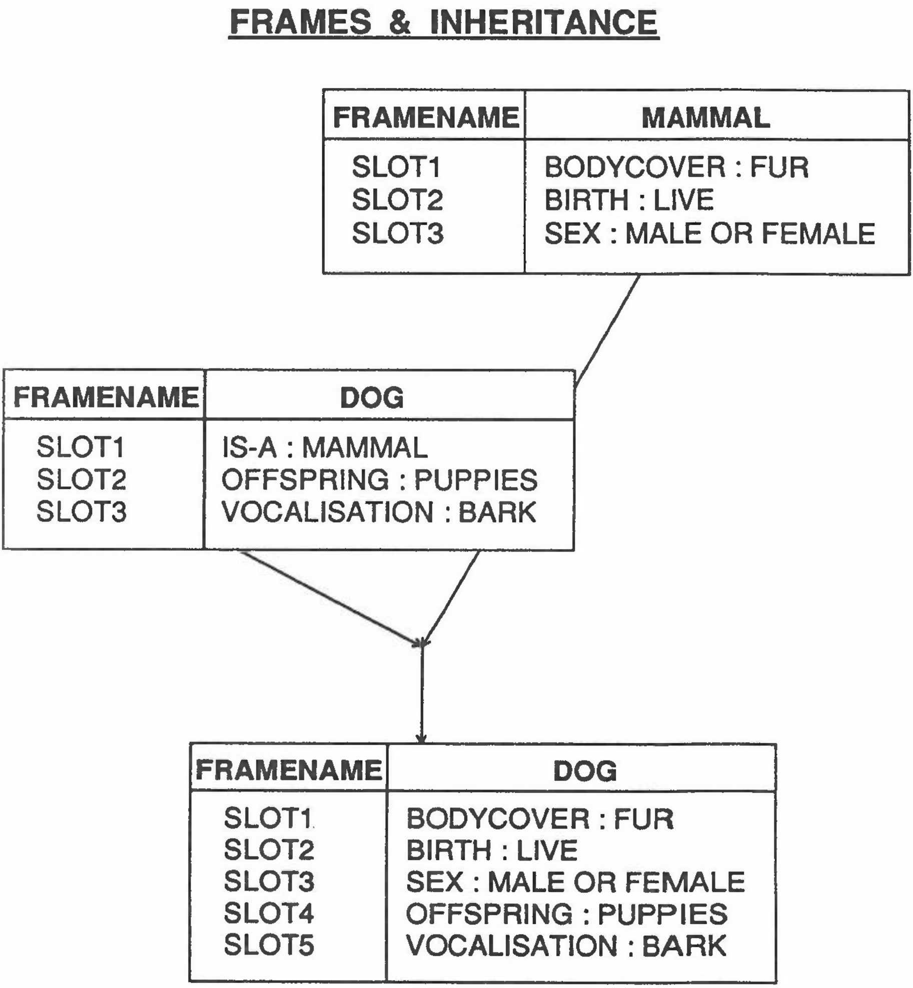
FRAMENAME	ELEPHANT
SLOT 1 SLOT 2	IS: A MAMMAL LEG: CARDINALITY: 4

Here the first slot is to be inherited without exception - an elephant must be a mammal. But the second slot is a default of the stereotype elephant, in that most people conjuring up the notion (frame?) elephant would attribute four legs, but any specific elephant, inheriting the ELEPHANT frame might have lost a leg and therefore have 3 for the second slot value. This slot then has a default value which should be easily replaced according to circumstance. Stereotypes will usually have many (perhaps most) slot values which are not strictly entailed by the top level node or frame name.

In summarizing what Minsky takes to be the essentials of frames and frame theory, at least one thing should be clear - a snappy one line description is not appropriate. So two definitions of frames from the literature

- 'A generalized property list' (Winston and Horn, 1984)
- 'An example of a structured object' (Bonnet, 1985)

hardly capture what Minsky was advocating. Sowa comes closest to capturing some of the essence of Minsky with his one line description



FRAMES & INHERITANCE

FRAMENAME	MAMMAL
SLOT1	BODYCOVER : FUR
SLOT2	BIRTH : LIVE
SLOT3	SEX : MALE OR FEMALE

FRAMENAME	DOG
SLOT1	IS-A : MAMMAL
SLOT2	OFFSPRING : PUPPIES
SLOT3	VOCALISATION : BARK

FRAMENAME	DOG
SLOT1	BODYCOVER : FUR
SLOT2	BIRTH : LIVE
SLOT3	SEX : MALE OR FEMALE
SLOT4	OFFSPRING : PUPPIES
SLOT5	VOCALISATION : BARK

Figure 1

'Prefabricated patterns assembled to form mental models' (Sowa, 1984) but clearly much is left out.

Another notion which Minsky wished to have embodied in 'frames systems' was what he called 'view changing'. This could be the changes of view a vision system experiences from relative rotation of the viewed scene, or in language systems 'procedures which in some cases will change the contextual definitional structure to reflect the action of a verb'. No real idea was given of how such a 'view changing' facility might be implemented. However the notion appears to have been influential in the use of 'viewpoints' and 'Kee worlds' in the ART and KEE commercial knowledge-based systems.

Minsky's paper concludes with a criticism of logic in knowledge representation. I believe that in this he was partly right in spirit, but mostly wrong in letter. These issues are discussed in the sections concerning Hayes' and Brachman's critique of frames.

4.3 Applications of the Frame Idea

In the preceding section we discussed Minsky's influential paper with some difficulty - since the rather vague allusive style makes it hard to clearly understand what exactly is being advocated. We now discuss some applications of the frame idea.

First we present story understanding, an application discussed by Minsky in his original paper. Then we review a medical diagnosis system: CENTAUR (Aikins, 1983).

An interesting application of frames is to story understanding. I am concerned that as an outsider to natural language work I may give a wrong impression to other outsiders. This concern arises because the results do not look impressive. Presumably the moral to be drawn is that, since the researchers are very able, then the problem is hard. Our example is taken from work on understanding news stories (De Jong, 1979) about earthquakes. The frame construction anticipates what would be expected in a short account of an earthquake:

FRAMENAME	EARTHQUAKE
SLOT 1	PLACE: LOWER SLABOVIA
SLOT 2	DAY: TODAY
SLOT 3	FATALITIES: 25
SLOT 4	DAMAGE: 500,000,000
SLOT 5	MAGNITUDE: 8.5
SLOT 6	FAULT: SADIE HAWKINS

The slot values are filled as above from the following news story.

"Earthquake Hits Lower Slabovia

Today an extremely serious earthquake of magnitude 8.5 hit Lower Slabovia killing 25 people and causing $500,000,000 in damage. The President of Lower Slabovia said the hard-hit area near the Sadie Hawkins fault had been a danger zone for years".

Having obtained the slot values from the news story these are inserted into the Earthquake Summary Pattern:

"Earthquake Summary Pattern

An earthquake today occurred in *value in location slot value in day slot* There were *value in fatalities slot* fatalities and *$ value in damage slot* in property damage. The magnitude was *value in magnitude slot* on the Richter scale, and the fault involved was the *value in the fault slot*."

This produces the following summary after instantiation:

"An earthquake occurred in *Lower Slabovia today*. There were *25* fatalities and *$500,000,000* in property damage. The magnitude was *8.5* on the Richter scale, and the fault involved was the *Sadie Hawkins*".

Now that seems quite good, though one might be worried about a news story which mentioned 25 *injured*. Wouldn't the system kill them off in the summary? A cruder problem is what happens to a story concerning earthquakes but not actually reporting one. Take the following news story:

"Earthquake Study Stopped

Today the President of Lower Slabovia killed 25 proposals totalling $500,000,000 for research in earthquake prediction. Our Lower Slabovia correspondent calculates that 8.5 research expenditures are vetoed for every one approved. There are rumours that the President's close adviser, Sadie Hawkins, is at fault".

This would produce an identical summary to the earthquake story. The solution, in principle, was stated by Minsky - namely give the frames sufficient information and procedures for them to recognize when to act - the solution in practice can be rather elusive.

The expert system PUFF (Kunz *et al.*, 1978) for lung function test is of considerable interest. Though not nearly so well known as MYCIN (Shortliffe, 1976), it is in widespread everyday use, whereas MYCIN has only ever diagnosed one patient 'in anger'. Another reason for interest is that Aikins, prompted by its deficiencies, produced CENTAUR (Aikins, 1983) which makes extensive use of frames to improve upon PUFF. PUFF is a clear descendant of MYCIN, using simple unstructured rules to represent knowledge, strategy and control. This modular, uniform representation was at one time held to be a positive advantage. In analysing PUFF, Aikins noted that though PUFF was adequate as a problem solver it had important shortcomings which she traced to the flat knowledge representation of rules. The problems found were:

(1) It was hard, or impossible, to represent typical sorts of patient and disease patterns.

(2) During a consultation it was difficult to modify the order in which questions were asked since these questions were generated by rule firings controlled by the interpreter, and much of the control information was implicit, buried in the rules themselves.

(3) Maintainability and modifiability were problems because of unanticipated side effects of rule changes or additions.

Frames provide, as Minsky intended, a suitable mechanism for representing stereotypical uses and also for embedding the specific stereotype in a more general stereotype. So, for instance, Aikins has a frame for OBSTRUCTIVE AIRWAYS DISEASE with lower order frames representing ASTHMA, BRONCHITIS and EMPHYSEMA, and also a set of four frames representing the degree of severity of the disease.

The issue of control as a problem for flat rule based systems had also been recognized by Clancey (1983) and Szolovits (1983). Szolovits noted that the overall strategy of MYCIN - try the most probable cause first - is nowhere explicitly represented, but is instead encoded separately and implicitly in each of the rule sets representing the 26 blood infections considered by MYCIN.

In CENTAUR Aikins solved this problem by using frames to give an explicit representation of how reasoning was to be controlled and to keep this separate from inferencing from data. Hence the name CENTAUR - the head is of different type to the body. The control of question asking mainly resides in the stereotypical frames - which may contain sets of production rules for inferring a required value - if the rule set does not instantiate a value then the user is questioned.

A related benefit from grouping production rules in sets of frame slots is that the rules are explicitly organized in relation to the stereotype being matched to data. This makes for ease of understanding, maintenance and modifiability. Though CENTAUR was no better as a problem solver than PUFF, it was argued that the representation of knowledge for both disease stereotypes and strategy greatly improved intelligibility and maintenance. These benefits, both for the expert and knowledge engineer, are clearly much more than symbol level implementation issues; they are of importance at the knowledge level.

Apart from Aikin's original paper, a thorough discussion of CENTAUR is given in Jackson's excellent book (Jackson, 1986). Jackson also discusses another medical expert system, INTERNIST, where frames were found necessary to organize the knowledge comprehensibly.

4.4 The Backlash - Hayes' 'Logic of Frames'

Some four years after Minsky's paper, Hayes (1979) analysed what frame representation had achieved. His conclusions were mostly negative. He held that with the exception of 'reflexive reasoning' no new insights had been achieved from work based on frame representations. Though we record this view as overly harsh, and indeed believe some of Hayes' assertions to be wrong, the paper is important in that it was the first attempt to subject frame structures and theories to a systematic logical analysis.

In his introduction, concerned with representation and meaning, Hayes loses no time in expressing a distaste for the lack of tight analysis in Minsky's paper and some subsequent work. "Minsky introduced the terminology of 'frames' to unify and denote a loose collection of related ideas on knowledge representation: a collection which, since the publication of his paper has become even looser. It is not clear now what frames are, or were ever intended to be."

He opens by discussing three different views of frames:

(1) as a formal language for representing knowledge, to be compared with, say, predicate calculus (representational);

(2) as a system which presupposes that a certain kind of knowledge is to be represented - this he calls the 'metaphysical' interpretation;

(3) as an implementation issue - frames are to be viewed as a computational device for the organization of memory, retrieval and inference.

Though he observes that Minsky seems to speak to the 'metaphysical' and implementation (or heuristic) interpretations, he largely bases his analysis on the representational interpretation. He notes that it has been common - and still is - to confuse these views, particularly the representational language in that it has a semantic theory which defines the *meanings* of expressions in the language. It is the semantic theory that changes a formal language into a representational language, and this theory must explain the way in which expressions carry meaning.

Having discussed the meaning of meaning, Hayes is ready to address the meaning of frames. To motivate his discussion he first considers a frame representing a typical house and then specializes to an instance of a particular house by giving (some) slots values. This example he then rewrites as a set of assertions in predicate calculus.

He concludes that, used in this way, stereotypical frames are bundles of properties expressible in predicate calculus, and particular instances are simply instantiations. This, then, suggests to him that frames are merely an alternative syntax for predicate logic - i.e. expressions about the relationships between individuals. However he notes that though the meanings appear to be the same, the inferences allowed by frames may be different from those sanctioned by logic, in some important way. It is then suggested that we must examine how frames are used to get a better insight into their meaning. Here "use" is plainly meant to mean what inference rules are used. This is a rather strange path given Hayes' representation - Hayes' prime focus is at least partly separate from inference. In particular, commitment to a representation does not automatically require commitment to a particular inference mechanism. Indeed, commitment to logic as representation does not commit one to deductively sound rules of proof. The reader is strongly encouraged to read the article by Israel (1983) for an excellent account of the distinctions (and confusions) between representation and reasoning.

Nonetheless, we shall follow Hayes' discussion of the *use* of frames. He considers a form of inference suggested by Minsky - 'criteriality'. The idea is simply that if we find values for all the slots of a frame, then we can infer that an appropriate instance of the concept represented by the frame exists. So if 'Dunroamin' has the appropriate slot fillers for kitchen, bathroom etc., then, by virtue of possessing these attributes, 'Dunroamin' satisfies the necessary and sufficient conditions to be a house. Hayes simply takes this example and maps it onto first order logic, where such slot names as kitchen become the function kitchenof. This example reinforces Hayes' belief that frames are just another syntax for first-order logic, with the defect that it is unclear whether criteriality is being assumed, whereas in clausal form this is obvious.

Hayes now considers a third form of frames reasoning - *matching* as made concrete by Bobrow and Winograd (1977). If we have an instance of a concept frame, say John Smith as an instance of Man, we can regard John Smith as an instance of another concept frame, say Dogowner. Hayes observes that a match may be established if the man frame had a slot for *pet* and this was filled by an object known to be a dog, or if the Dogowner frame had a slot for owner name and dog and name was filled by John Smith, and dog filled by anything. Either would be sufficient for the instance John Smith to be matchable to Dogowner. He points out that the knowledge is expressible in first order logic, and the result is obtained through the standard inference rules of logic. He leaves out entirely the situation where 'matching' fails - Minsky envisaged this as a trigger to seek alternative frames, or construct new ones. Hayes allows that more profound use of the matching process might be envisaged, but would probably be only expressible as in higher-order logic, and remarks that it may not be possible to implement such schemes.

This hints at a tendency which becomes more marked in the discussion of defaults and stereotypes - choosing examples which are expressible in first order logic, or if they are not so expressible ignoring them.

Take first stereotypes, which Minsky asserts to be a very significant component of the frame notion. Hayes distinguishes three ways in which stereotypes may be used.

The first is simply filling in details - this may result in the satisfaction of criteriality or in matching - in either case expressible in predicate calculus.

The second is as the *correct* way of looking at a single thing. Once again this can amount to criteriality - but there can be a difficulty - a single thing may have apparently contradictory properties or be seen from different points of view. He takes an example of someone who is unfriendly at work but friendly at home. This again is dismissed as a non-problem with three sketched possible solutions; which, unless the solution is translated into assertions with consequent contradiction, he has no theory for. There does appear to be a solution at least partly in the spirit of Minsky; namely, the "Viewpoints" of ART or the 'Kee worlds' of KEE. This allows for the maintenance of internally consistent hypothetical worlds which may contradict each other, and for drawing inferences between these worlds. Though Minsky appears to have some of the idea of this mechanism, and presumably influenced its development, his suspicion of inference apparently did not allow him to discuss consistency in each separate viewpoint. Nevertheless this seems evidence against Hayes' assertion that no new insights resulted from the frame movement.

His third view of stereotypes is to understand them as representing a metaphor or analogy. Frame-like structures were used for analogical reasoning in MERLIN (Moore and Newell, 1973) and Minsky acknowledges this as a major influence. The example considered is 'pig' as an (unkind) metaphor for man.

Hayes seems to make heavy weather of metaphor, noting at some length that a man cannot literally be a pig. This means, in the jargon, that matching of a metaphor should never establish criteriality. But if matching does establish criteriality, as it may to Bobrow and Winograd (1979) - but not to Minsky - then frames have a problem in that they cannot distinguish a metaphor (or 'mere caricature' to Hayes) from a real assertion. Hayes does note however that the bundling of the appropriate properties in the frame used in metaphor would be a criterion for pig-likeness as distinct from pig-hood. Having scented 'criterial' he then notes that a systematic vocabulary translation could allow logic to do all (and more) than is claimed for the frame as metaphor. Though he allows that logic itself does not provide the syntactic machinery for this translation, an analogy is sketched to an unreferenced use of analogy in mathematics. It is claimed that if a caricature frame contains the translation information then all is reducible to first order logic. Surely the point here for frame aficionados is that once they are allowed to characterize a frame as a metaphor, matching then validates the metaphor as a metaphor and not a true assertion - so no translation is needed.

The weakest of all the discussion in 'Logic of Frames' concerns defaults. One has the feeling in this section that Hayes' considerable, but incomplete, success in showing a correspondence between frames and first order logic has now convinced him that all aspects of frames are expressible in first order logic. He defines a default as the slot value in the absence of contrary information, but does not add that with such contrary information the value should be replaced. He does not say that defaults take us out of first order logic - but only that they *seem* to do so. This apparent unequivalence is asserted to be a consequence of a *naive* mapping of default reasoning onto assertional reasoning. After discussing an example he concludes that what is required for default reasoning is some process which subtracts previous beliefs or assertions. The trouble is that classical logic does not allow this - it is monotonic - beliefs can only be added and conclusions increase monotonically. Hayes simply disagrees - he asserts that no new logic is needed - merely some new primitives and the ability to "talk about the system itself". This is firmly disputed by at least some of the workers cited as producing such a system. Etherington and Reiter (1983) state "... common sense reasoning about exceptions is non-monotonic, in the sense that new information can invalidate previously derived facts. It is this feature which precludes first order representations, like those used for taxonomies, from formalizing exceptions". We should conclude that Hayes fails completely to establish an equivalence between first order logic and frame defaults. The situation is clearer now (1987) than at the time of Hayes' paper (1979) and it appears that most workers in the field agree with the quote from Etherington and Reiter.

For exceptionless hierarchies - which are representations of universal set inclusion - it should come as no surprise that first order logic *can* represent the knowledge as can frames and there is a mapping/transformation between the two. This of course does not show frames are 'merely' first order logic. It is hard to believe that anyone who has seen a frame representation for a taxonomic hierarchy could fail to believe, whatever the logical equivalence, that for this case frames are a superior representation to an unstructured sequence of clauses. An example from physics - cartesian and spherical polar coordinates are equivalent - but cartesian coordinates are far superior for solving problems about particles in cubic boxes, spherical coordinates superior for solving the hydrogen atom. So I would claim the 'bundling' aspect of frames, even for those cases where there is a mapping onto predicate calculus, is by no means the triviality Hayes makes it out to be - and has been one of the major practical attractions of the frame idea.

Just because replaceable defaults cannot be handled by predicate calculus does not mean that frames do not have problems and in the next section Brachman's concerns (Brachman, 1985) are discussed.

In the final section of 'The Logic of Frames' it is argued that one positive product of the frames movement is the idea of *reflexive reasoning*. Reflexive reasoning is that in which the reasoner thinks about himself, in particular about his own reasoning process. Interestingly, the possibility of this knowledge level activity arose in part out of an implementation (or symbol) level issue. In some, but not all, frame systems the frames are implemented as structured objects in the sense of Object Oriented Programming (OOPS) - see for instance (Bobrow and Winograd, 1979) and (Bobrow and Stefik, 1983). This implementation, at least in principle, allows for the possibility of reflexive reasoning. So far there are only two or three pioneering examples in the literature, and these suffer from a somewhat *ad hoc style*; however, work is being pursued to put the basis of reflexive reasoning on a firmer foundation - see (Maes, 1986).

Summarizing, then, Hayes set out to show that almost all of frames is equivalent to some of first order logic, and so the movement had produced no new insights. Although he did some of the job, he did not complete it . His dismissal of the use of frames as stereotypes, their use for 'seeing as', is unconvincing, and his section on defaults, where he asserts that no new logic is required, is wrong. Still, the paper is important, and beautifully written - whether or not one agrees with Hayes there is usually no doubt what he is saying and why he says it - not something that can be said of Minsky's paper.

4.5 Problems with Defaults and their Cancellation - Brachman's Elephant Joke

In the discussion of Hayes' 'Logic of Frames' we observed that though Hayes was wrong to say that no new logic was needed to handle defaults, this did not mean that all was well for every, or perhaps any, frame representation of default reasoning. Brachman, in a superb essay (Brachman, 1985) which sets depressingly high standards for just what can be achieved in a popular account, brings out the problems of replaceable defaults in frame (or semantic network) representations.

Very crudely summarizing, though Hayes argued that most of the frames idea is equivalent to predicate logic, Brachman makes a case that none of the default aspect of frames is logic, and indeed they let you do very little consistently. Refining the above caricature a little, Brachman's main point is that once overrideable defaults are used uniformly then definitional (or criterial) conditions cannot be used, and without definitional power frames cannot express simple composite descriptions such as 'polygon with four sides'.

The argument is organized as follows: first it is shown that the use of cancellable defaults forces us to represent everything in terms of defaults, and this does not allow us to represent universal truths. Then comes the really great problem - this means that even the simplest of composite descriptions cannot be constructed - so every description must be a primitive.

Now one might think that the first problem is terrible - being unable to represent universal truths - like 'every rhombus is a quadrilateral'. But that is not so, for two reasons. First, if you stick to the 'ideal' world of geometry no default cancellation is needed, and frames, logic etc. work fine. But much more importantly, if you do not stick to 'ideal' worlds but try to deal with the real world as the frames movement wants to do, then you have to recognize that 'natural kind' concepts like elephant cannot be defined (Putnum, 1977). So it could be argued that the loss of definitional capability is no great loss after all if our main goal is to represent the real world and the real world is not amenable to definition. Similarly, classical logic is inadequate, in the sense that 'natural kind' concepts cannot be represented by a finite number of necessary and sufficient conditions. Now what is devastating about Brachman's argument is that though you can define 'polygon' and you cannot define 'elephant' you surely should be able to conclude that an ELEPHANT-WITH-THREE-LEGS is an elephant just as you can conclude that a THREE-SIDED-POLYGON is a polygon. But, argues Brachman, with cancellable defaults you cannot - ELEPHANT-WITH-THREE-LEGS is not a composite description, it is a new primitive because we have lost all definitional capacity.

Before wrapping up, he addresses the question of what frames in systems with cancellable defaults might be, discussing various meanings of the IS-A link.

A word of caution - before deciding you will never (again) have anything to do with frames, keep in mind that Brachman's argument bears on frame systems with " .... a uniformly applicable facility to cancel properties". Brachman himself notes that it is possible to have systems where some defaults are explicitly uncancellable whereas others are explicitly cancellable and notes that thus explicit marking of defaults may also have expressive value. So frames are not necessarily bad for you, but badly (or not at all) thought out knowledge representation frameworks are.

Now to the argument. Brachman takes frames with inheritance to mean that subframes inherit all the properties of their parent frame. The specific example considered is the frame for elephant which is a subframe of the parent mammal frame. This could be represented as:

FRAMENAME	ELEPHANT
SLOT 1	SELF: A MAMMAL
SLOT 2	TRUNK: A CYLINDER
SLOT 3	COLOUR: GREY
SLOT 4	LEGS: CARDINALITY: 4

Now what does this mean? The first possible meaning discussed is essentially that put forward by Hayes. If the inference mechanism of frame systems is inheritance, and no defaults are cancellable, then one could argue that the elephant frame is stating the necessary conditions for class membership. So if CLYDE is an elephant by virtue of some kind of IS-A link to the ELEPHANT frame then CLYDE necessarily is a mammal, has a cylindrical trunk, is coloured grey and has four legs. In this interpretation Hayes would be quite right to interpret the slots of the frame as right-hand sides of conditionals.

This interpretation simply will not do for the 'natural kind' concept elephant. It is entirely thinkable that a given elephant would have none of the properties specific in our frame, apart from being a mammal. Brachman conceives of an unfortunate elephant suffering from hepatitis which is consequently yellow not grey. (Honestly it won't affect the argument if elephants stricken with hepatitis do not go yellow, or even if elephants are immune to hepatitis.) To handle this problem of exceptions most frame systems allow the overriding or cancellation of a property that would normally be inherited. Now once this is so we cannot use the Hayes interpretation of the colour slot - 'every elephant is grey'. Instead there has to be an interpretation such as 'the typical elephant is grey' and grey is a *default* value - in the sense originally suggested by Minsky.

Brachman observes that typical has nothing to do with frequency - one is not saying most elephants are grey (though probably they are) or even that grey is the most frequently observed elephant colour. Instead, the force of the statement is "in the absence of any evidence to the contrary, assume that any given elephant is grey". This interpretation is consistent with Minsky's paper - though he does not give it - and also with subsequent more or less formal approaches to default reasoning (Reiter, 1978).

At this stage we ask what has been given up to talk about conceivable real elephants. If the mechanism is the *uniform* (my emphasis) possibility of default cancellation then we have lost the possibility of representing necessary universal truths like the truths of geometry or arithmetic. Of course we might puzzle over why someone would like to represent and reason about the abstractions of plane geometry in the same system that considers real elephants. Remember there is a subtle difference between the statement that 'all quadrilaterals are polygons' and 'all elephants are mammals.' The first is true by definition and cannot in principle be otherwise, the second is a matter of scientific discovery - we *could* have been wrong.

A slightly less obvious problem is that abandoning ‘every’ means that we cannot represent contingent universal facts with a single statement. This is definitely a nuisance. One might well observe that *all* the vehicles in the parking lot are cars - there are no vans, motorcycles etc. Having a true every, as in predicate calculus, allows us to capture this without even knowing, as one need not, how many cars there are, still less representing each one. This is clearly a loss compared with predicate calculus - but not, as Brachman brings out - the most severe loss.

The worst problem concerns that of description. The essence of the problem is simply represented - once we have the concept elephant, never mind how we came by it, then we should be able to construct an indefinite number of composite concepts, each of which is related to the parent concept ELEPHANT by definition - i.e. necessity and sufficiency. So consider the concept of an elephant with three legs - the frame for this might be called ‘ELEPHANT-WITH-THREE-LEGS’. This concept is just the composition of two attributes, each necessary and both together sufficient. We require that it is impossible to have an elephant with three legs that is not an elephant, and that it is impossible for something that was both an elephant and had three, and only three, legs not to belong to ELEPHANT-WITH-THREE-LEGS. Stating the problem in a slightly different way, ELEPHANT-WITH-THREE-LEGS should stand in the same relation to ELEPHANT as POLYGON-WITH-THREE-SIDES stands in relation to POLYGON. Now the ‘natural kind’ concept ‘elephant’ is very different from the mathematical idealization ‘polygon’ in that the latter is a well and simply defined concept, whereas the former has no definition at all. But still we must, without fail, infer that an elephant with three legs is an elephant just as we conclude that a polygon with three sides (the primitive triangle) is a polygon. The crunch is that in allowing the *uniform* (my emphasis again) overrideability of defaults we have lost the definitional capability to do this - and cannot represent that ELEPHANT-WITH-THREE-LEGS and similar compositions are more like POLYGON-WITH-THREE-SIDES than they are like ELEPHANT. The analogy to natural language is clear - it is as if we had given up the possibility of noun phrases and were just left with simple nouns - since now all frames are primitives.

At this point Brachman rather goes overboard and tells us that frame systems cannot deduce that a rhombus (a polygon with four equal sides) is also a quadrilateral with four equal sides, and that the concept of a three-sided rhombus would be just as acceptable. It must be noted that (most) frame systems allow inheritance without cancellation, and that is just what is wanted for completely definitional representation. Indeed the construction of a frame system which never lies about plane geometry is a standard tutorial exercise used by the makers of one commercial AI-toolkit.

One of Brachman's parting shots concerns IS-A links. He observes that the informal practitioners of knowledge representation often confuse different meanings of IS-A. He gives three different meanings of IS-A:

(1) the *concept* of a *kind* of thing (e.g. elephant);

(2) a generic *description* specifying the properties that *typically* apply to instances of a kind of thing;

(3) a "*stereotypical*" (he says prototypical) *individual* somehow typifying the kind.

Confusing these different meanings is evidently no way of going about knowledge representation - but it has happened. One might, in mitigation of the informal school observe that the verb 'to be' has given philosophers a lot of trouble - see, for instance Russell's History of Western Philosophy (Russell, 1945). The prosecution might argue that, not knowing the relevant literature, particularly from another field, is a frequent failing in AI. Sowa (Sowa, 1984) has made this point with many telling examples.

Brachman's conclusion is that the very proper desire to represent 'natural kind' concepts had led to naive mechanisms that admit arbitrariness and force ignorance of crucial facts. As he notes, when *all* rules (of inheritance) are made to be broken, then no rules are left. But the rule that elephants with three legs are elephants is not to be broken. As he pithily remarks "(some) AI systems have thrown out the compositional baby with the definitional bathwater".

As we have hinted, though Brachman's argument is convincing, it does not close the door on an effective and useful frame representation. What is needed is a representation where some properties and property values (slot values) are inherited without exception ('sacred') and others are explicitly marked as replaceable. These 'sacred' markings should also distinguish between the slot value being uncancellable and the slot name (or attribute) being uncancellable. As he notes, the presence of an *explicit* cancel link adds to the representations' expressiveness. A cancelled attribute could give information about the property's history and applicability. This is clearly an area for more research and Brachman and friends have been active. One apparently promising approach used in the system KRYPTON (Brachman, Fikes and Levesque, 1983b) is to separate as completely as possible definitional and factual information. Lambert discusses this work in Chapter 10.

Finally we should note that though the 'informal' school of knowledge representation has suffered some well-aimed blows (and others not so well aimed) as reviewed in these last two sections, the 'formal' school appears to be having considerable problems with formalizing 'common-sense' reasoning. We give a short account below.

4.6 The Problems of Formalizing Default Reasoning

Hayes' analysis of default reasoning in 1979 (Hayes, 1979) mentioned work in progress by McDermott and Doyle and also by Reiter which it was hoped would give a formally satisfactory account of default reasoning. This is also referred to as work on the problem of non-monotonic logic. Classical logics are monotonic in the sense that assertions once made cannot be retracted, and so the numbers of conclusions increase monotonically with the number of assumptions. In default reasoning one is allowed to retract assertions in the light of evidence - the conclusions of a logic which allows this is non-monotonic with respect to assertions.

The enterprise of formally representing default reasoning looked very promising (see McCarthy, 1980; McDermott and Doyle, 1980 and Reiter, 1980). But prior to 1986 problems have surfaced in the work of Reiter and Doyle and McDermott.

The problem is that the non-monotonic extensions to predicate calculus seemed to allow only weak, or even no, conclusions be drawn. McCarthy's more elaborate formalism was not known to suffer from these problems until the analysis of Hanks and McDermott (1986). In this paper they claim to show by a detailed example that McCarthy's approach also, in some circumstances, yielded no useful conclusions in the sense that the system yielded two contradictory conclusions, and gave no guidance in selecting one of them. At the time of writing this is still recent work and we cannot claim any final conclusion. However there is at least some evidence that formal methods may not be able to handle the sort of default reasoning required for representing common-sense reasoning. Since this is one of the key issues in AI we should expect much work and lively debate on these issues.

4.6.1 The 'Frame Problem' - a Brief Note

Since Hanks and McDermott mention the 'frame problem' it is worth noting that this problem has nothing specifically to do with frames as discussed in this chapter. The frame problem recognized by McCarthy and Hayes (McCarthy and Hayes, 1969) concerns the problem of expressing information about what remains unchanged by an event. The essential assumption is that a state persists between events. This is discussed at greater length in Chapter 9.

4.7 Comparisons and Conclusion

The use of frames to represent knowledge about structured domains or structured knowledge continues to increase in popularity, particularly as systems get larger. The wide variety of hopes (explicit and implicit) in Minsky's paper (Minsky, 1975) are unlikely to be fulfilled completely - in particular 'common-sense' or default reasoning cannot be effectively represented by the cancellable inheritance of defaults.

Though Hayes (Hayes, 1979) showed a logical equivalence of a particular use of frames to first order predicate logic, this does not diminish the utility of the frame approach, since the additional structure which can be represented or imposed by frames has considerable value.

One demonstration of this is the reconstruction of knowledge bases originally expressed in unstructured production rules into frame systems and the consequent improvement in system understanding and ease of maintenance.

Frames have some similarity to semantic networks - a frame system whose frames consist of the framename and top relation slot is just a semantic network of nodes joined by relation arcs defined in the relation slot. The issue of inheritance and cancellable defaults arises in exactly the same way for semantic networks, though the argument may be a little less clear.

5 Rule Based Systems

Tony Williams and Brian Bainbridge

5.1 Introduction

The knowledge representation that may be already familiar to the general reader is the production rule formalism. In this chapter, the origin and development of production rule systems will be examined, applications will be described and an evaluation of the formalism will be made.

5.2 Basic Components and a Simple Model

In general, production systems have three main components: working memory, rule memory and the interpreter. The architecture and execution cycle of a simple production system comprising these three components is given in Figure 1.

The *working memory* is a store containing objects defined by attribute-value lists. These objects represent facts about the world, either given, observed or inferred. As we shall see, they may also represent working hypotheses, rather than real facts. These working hypotheses may be modified or withdrawn in the light of subsequent information. The term *fact* is used loosely in this chapter to refer to all kinds of object in working memory.

For example, using a Lisp-like notation:

```
(       (Patient-ID  12345)
        (Patient-name Smaug)
        (Complaint Bad-breath)
)
...
(       (Patient-ID 12345)
        (Skin-condition  Green-scaly)
)
...
```

This set of objects includes the information that a patient named Smaug has complained of bad breath. The physician has observed that the patient has green scaly skin. The system has assigned an identifier for this patient, and no doubt has other information stored.

The *rule memory* contains rules governing the system's behaviour. These rules have the form

IF condition(s) THEN action(s)

At first sight, these rules appear similar to conditional statements in conventional programming languages, such as the logical-if statement in Fortran. The differences will be discussed later. The conditions govern the premises for selection of this rule, and are sometimes referred to as *antecedents* or left-hand sides. A condition defines a pattern to be matched against the content of the working memory. Such a pattern can match one or more objects whose attributes conform to requirements expressed in the pattern.

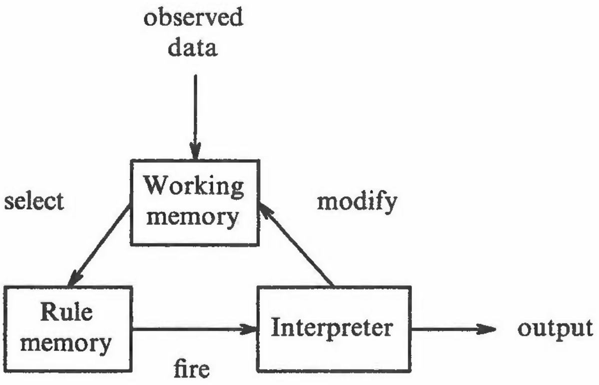


Figure 1 Production system execution cycle

An action defines modifications or additions to the working memory, and may include side-effects, such as output. The resultant changes to working memory play the role of inferences in an expert system. Actions are sometimes referred to as *consequents* or right-hand sides of the rule.

Contrived examples of rules might be:

```
IF    (Complaint Bad-breath)
AND   (Patient-species dragon)
THEN  (assert (remedy mouthwash))

IF    (remedy ?x)
AND   (no-side-effects ?x)
THEN  (print "Take" ?x "and see me in a month")

(?x signifies a variable)
```

The first rule states that the way to cure bad breath in dragons is for them to gargle with mouthwash. The second rule states that if a remedy has been inferred (as indicated by the presence of a remedy attribute in working memory) and there are no unwanted side-effects, then the patient can be given the prescribed remedy.

Rules have a readily understandable form, provided that the condition part does not become too complex. They can be used in explaining to the user why the system made a particular deduction, because they directly state the information on which the deduction was based, and the reason why the deduction holds.

The *interpreter* (also called the inference engine) is the active component of the system. It selects rules from the rule memory that match the contents of the working memory, and performs the associated actions. This is termed *firing* a rule.

Two factors distinguish rules from conventional conditional statements:

(1) the conditional part is expressed as a (possibly complex) pattern rather than a boolean expression;

(2) the flow of control (as found in conventional languages) does not pass from one rule to the next in lexical sequence but is determined entirely separately, by the interpreter.

The first distinction can be seen as mere syntactic sugaring, as an equivalent pattern matching function can be called from within a boolean expression. The second distinction is more significant. It allows separation of the knowledge from control of how the knowledge is applied. Knowledge bases can be expressed as sets of rules, each of which can be validated independently of the control structure. Each rule expresses a relationship between antecedents and consequents which must hold in a static way: the "truth" of the rule must hold independently of when it is applied. This implies that the antecedents of the rule must adequately determine the context in which the consequents apply.

5.3 Survey of Production Systems

Post (1943) used production systems in symbolic logic and invented the name. In mathematics they first showed up as Markov algorithms (Markov, 1954), and they have been used in linguistics under the name of rewrite rules (Chomsky, 1957). Later, the formalism was used in programming languages such as SNOBOL (Farber, Griswold and Polonsky, 1964), and in compiler translation languages (Floyd, 1961).

An illustration of the use of production systems in linguistics is in expressing a context-free grammar. For example, an insult grammar (a favourite of introductory AI texts) could be expressed by these rules (after Bundy *et al.*, 1980):

```
insult   => suggest 'you misname
suggest  => 'buzz 'off
suggest  => 'go 'jump 'in 'a 'big 'hole
misname  => 'nasty 'fellow
misname  => 'little 'toad
(a quote signifies a literal)
```

This will generate such marvels as

"go jump in a big hole you little toad"

The first use of production systems in knowledge-based systems seems to be in 1965, when Herbert A. Simon and Allen Newell at Carnegie-Mellon University used them in a chess analysis program (Simon and Newell, 1965). Since 1954, these researchers have worked on aspects of human problem-solving (Newell and Simon, 1972). They have studied the performance of intelligent adults on short (half-hour) tasks of a symbolic nature, tasks not centrally concerned with perception or motor skill. Three main problem areas have been used - symbolic logic, chess problems and algebra-like puzzles. Test subjects are asked to perform the task, and to "think aloud" as they do so. These protocols are recorded and transcribed to form the data to be represented in a production rule formalism. The resultant system behaves in a similar way to the human problem-solver, and this can be interpreted as being a result of the similarity of the two information processing systems.

These studies have typically not been concerned with learning and age-related differences or development. Later applications of production systems have been more concerned with such areas, and have dealt with such domains as seriation (putting physical objects in order) and learning arithmetic operations (Young, 1976, Young and O'Shea, 1982, Evertz, 1982).

In these types of research, the production system is regarded as being a psychological model of human knowledge and skills. It is possible to model adaptive behaviour by using rules which modify rule memory (seen as akin to human long-term memory) and/or the working memory (seen as akin to human short-term memory). For example, it has been suggested that user interfaces to computer systems could be built which would adapt to the user, and accelerate the more common interaction sequences (Hopgood and Duce, 1980). Conway and Wilson discuss this approach to modelling human procedural knowledge in Chapter 6 of this text.

The other main approach has been technological in thrust rather than psychological. Whatever their significance for psychological modelling, it can be said that production systems offer a useful *ad hoc* programming formalism. However, there are problems in using them for realistic knowledge-based systems applications, some of which are touched on in succeeding sections. Useful work has been done to improve efficiency and representational power, and commercial quality tools such as the OPS languages (Forgy, 1981, 1982) and ART (Laurent *et al.*, 1986) are now available. Large applications such as R1 (McDermott, 1982b) and MYCIN (Buchanan and Shortliffe, 1984) have been developed. Finally production system architectures are being designed, such as the RISC machine based on gallium arsenide technology proposed by researchers at Carnegie-Mellon University (Lehr and Wedig, 1987).

5.4 Extensions to the Simple Model

The simple model of production systems described in section 5.2 is inadequate for implementing commercial quality knowledge-based systems. In the first instance, problems arise concerning rule selection, control strategy and permissible actions.

5.4.1 Rule Selection

The first problem with rule selection is *conflict resolution*: determining which rule to fire when more than one set of conditions match the working memory. The simple solution is to fire the first rule whose conditions match. However, this strategy means that the designer of the system has to ensure that the rules are in the correct order. For example, it might be required to deal with exceptions or unusual cases first. This means that rules with more antecedent clauses, ones with more conditions and which are therefore more specific in their application, would have to be at the beginning of the rule set. If a new rule were added, it would have to be inserted into the rule set at the 'correct' position. One of the advantages of production rules, that the rules are modular and represent a separate chunk of knowledge, only weakly coupled to other rules, has been lost if ordering has to be done. More sophisticated production systems, e.g. the OPS family of languages, provide explicitly for a conflict set: the set of instantiations of rules that match the current contents of working memory. The user is allowed to choose a particular conflict resolution strategy, such as giving preference to rules that operate on the most recent information (this provides focus), or giving preference to those rules that match the most items (to implement the strategy described previously). As a last resort, if the conflict resolution cannot indicate which rule should be selected, a rule is chosen arbitrarily.

A second problem arises with large rule sets: there may be many rules to choose from, only some of which lead to the desired result quickly. In this case, some systems use heuristics, perhaps encoded as metarules, to perform more sophisticated conflict resolution. (Metarules are rules which control the use of domain rules.) The metarules could refer to particular domain rules, either by name or by pattern matching their conditions and/or actions. The conflict resolution strategy is thus embodied in the metarules. Figure 2 shows the schematic architecture of a production system with metarules.

An example might be a financial expert system which could contain the metarule:

IF the company is seeking finance
THEN first consider rules that conclude medium-term finance

(possibly because experience has shown that medium-term finance is the best candidate as a source of company finance). This is an instance of the more general metarule which states that the most likely candidates should be tried first. An example of the more explicit use of metalevel control is the latest version of the ROSIE production rule language (Sowizral and Kipps, 1986), a much augmented successor of the earlier RITA system (Anderson and Gillogly, 1976). ROSIE has become its own metalanguage, in that in the language itself it is possible to define the action of the inference engine and the conflict resolution strategy.

5.4.2 Control Strategy

The simple model given in section 5.2 describes a forward-chaining data-directed production system. The system starts from observed data, and proceeds to infer all possible consequences (at least in principle). Particularly for usefully large rule sets, this may lead to a combinatorial explosion

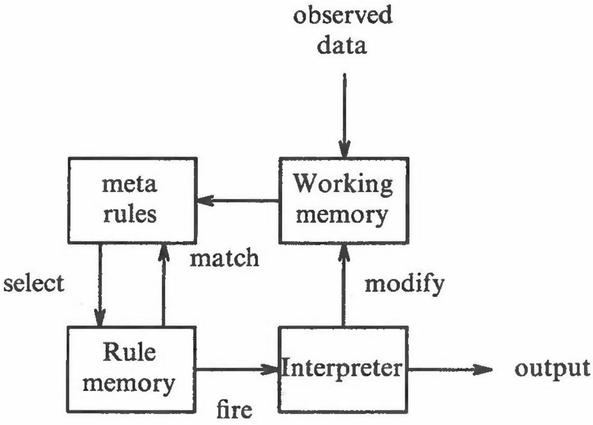


Figure 2 Production system with metarules

in the working memory, and in the conflict set.

An alternative control strategy is known as backward-chaining, or goal-directed search. In this model, the system is given some goal to achieve. The interpreter selects rules that may lead to that goal, and infers the subgoals required to satisfy those rules. The subgoals are then put into working memory, and the cycle continues until all subgoals are satisfied. The intent is that the system will perform more efficiently and in a more focused way, because the rules are being selected in a sequence which is proceeding towards the desired goal. Again, conflict resolution is required, and heuristics or metarules may be applied. The MYCIN series of medical expert systems use a goal-directed control strategy to implement the classification strategy of a doctor treating microbial disease, and are described in Chapter 8 by Bainbridge (and see Buchanan and Shortliffe, 1984).

The forward-chaining production system architecture can be viewed as one particularly suitable for dealing with data opportunistically, as it arrives. It is also suitable for dealing with synthetic tasks, such as configuration of computer systems or flexible manufacturing systems where there are very large numbers of possible goal states and there is really no choice but to be driven by the data to a suitable goal.

The possibility of backward-chaining and a goal-directed strategy represents a considerable extension to the power of a production system. Often a subtask will involve a small number of pre-enumerated goals. For example, a medical diagnosis system dealing with microbial infections might have only 20 candidates. Backward-chaining gives a focused and efficient way to deal with such a situation. Recent systems, such as ART, offer hybrid strategies, with both types of chaining. Such systems have to be used with care, as noted below.

5.4.3 Permissible Actions

The action part of a rule normally contain a series of actions to modify the contents of working memory, by adding, removing or altering facts. Facts are added to working memory as new information is inferred. Facts can be retracted, to prevent their being matched in future rule selection. Facts can be modified, to add information about them.

Other types of action may include side-effects such as output to the user of the system, or to some other subsystem. Such actions make the effect of a rule firing externally visible. Additionally, actions might alter some global state information within the overall system, or even modify the rule memory itself. Such assertion and retraction can cause problems, particularly with hybrid control strategies. Such a strategy may find a rule sequence by backward-chaining in an attempt to prove a goal. If in the course of this chaining a rule fires forward, side-effects may destroy the justifications of the current state, since facts (and rules) may be retracted or asserted that invalidate that state. If it is desired to use such non-monotonic reasoning, the programmer will have to be very careful indeed to ensure adequate control of side effects. ART recognizes this problem by offering an assumption-based truth maintenance system (known as Viewpoints). It would be fair to say that this area is still not understood well.

Other possible forms of interference are global side-effects which alter conflict-resolution strategies or heuristics. Again, it is the job of the programmer to deal with this in a principled way.

5.4.4 Improved Representation of Domain Knowledge

First, a production system provides an empty 'shell' in which domain knowledge can be embedded. It can be used to implement any system you want - a credit-card authorizer, a chemical spill disaster system, an assistant to help intelligence analysts to deal with international terrorism, or whatever. At least, that is what the vendors tell you.

Real-life experience serves to modify this simplistic view. Any given domain will have specific representation requirements. For example, the spill system might require the representation of a drainage system as some sort of graph which can be searched. The international terrorist system could involve the representation of a taxonomy of terrorist organizations. It would be convenient if such representations were easily implementable and easy to manipulate.

Second, a production system is not really an empty shell. The interpreter contains proceduralized metaknowledge about how to deal with rules. It would be useful if this metaknowledge were available for use and manipulation (Clancey, 1983).

We will now consider these two factors in detail.

5.4.5 Complex Domain Knowledge

The simple way in which facts about domain objects are recorded in a classical production system makes it difficult to represent complex information, such as:

(1) a cluster of facts about a given object;

(2) complex relationships between objects, such as taxonomies;

(3) information about prototypical objects;

(4) exception information about object classes or instances.

Logic-based representations, semantic networks and frames are often more convenient representations. For example, in a frame system, it would be possible to define a frame to represent the class of small companies and to create instance frames to represent individual companies. It would then be possible to deal with

```
(frame   ABC_ltd
         (instance-of small-company)
         (capitalization  100000)
         (market-sector double-glazing)
         (created 1984))
```

rather than the facts:

```
(small-company ABC_ltd)
(capitalization ABC_ltd 100000)
(market-sector ABC_ltd double-glazing)
(created ABC_ltd 1984)
```

Examples of additional representational power added to production systems are CENTAUR (Aikins, 1983), which uses frames to represent knowledge about the form of a consultation and the taxonomy of lung diseases (see Chapter 8), and AM (Lenat, 1982), which uses frames to represent "interesting" mathematical concepts.

Other areas where there are particular representational problems are the representation of uncertain knowledge and knowledge about time (temporal knowledge). The latter is treated by Kwong in Chapter 9.

5.4.6 Domain-related Control Knowledge: R1 as a Case Study

One of the advantages of production systems is that the control knowledge is simple and is embedded in the interpreter. But often we have a great deal of domain-derived knowledge concerning control, and find some difficulty fitting this into the Procrustean bed provided. We wish to use our own metaknowledge, rather than that supplied.

As a case study, it is useful to consider the control aspects of the R1 system used by DEC to configure VAX and PDP11 minicomputers (McDermott, 1982b; Bachant and McDermott, 1984). (R1 is known within DEC as XCON. The original name, according to McDermott, came from his realization in 1982 that "Four years ago I couldn't even say knowledge engineer, now I ...").

The implementation language of the present version is OPS5. The syntax of OPS5 is similar to the simple system described in section 5.2. The language has been optimized for maximum efficiency of the computationally expensive process of pattern matching. It uses a mechanism known as Rete match (Forgy, 1981, 1982) which avoids the repetition of attempted matches, as a collection of production rule antecedents are matched with a collection of working memory elements.

OPS5 is a general-purpose production system language - it "knows" nothing of the configuration task. However, it is used in R1 to represent the knowledge used in configuration. The interesting question is how good, in some sense, is the representation?

The task itself is complex, and divides into two subtasks:

(1) check that the order from the customer for the set of items that make up a VAX (or PDP11) minicomputer system is complete, and correct it if it is not;

(2) determine the spatial arrangement of the components.

The output is a correct component list and a set of diagrams specifying the cabinets required, the position of the units within the cabinets, the control panels, the cabling and the floor plan.

What is required is a satisfactory solution, with no missing or unwanted components, and without excessive unused space within the cabinets.

A sample rule, to illustrate the type of knowledge used, is:

```
RULE VERIFY-SBI-AND-MB-DEVICE-ADEQUACY-3
          IF    the most current active context is verifying
                SBI and Massbus adequacy
          AND   there are more than two memory controllers on the order
          THEN  mark the extra controllers as unsupported
                (i.e. not to be configured)
          AND   make a note to the salesperson that only two memory
                controllers are permitted per order.
```

(Rendered into near-English from OPS5)

The meaning of this rule is reasonably obvious (even though what SBI and Massbus mean might have to be guessed by the reader). What is not apparent is the nature of the configuration strategy.

There are not a small number of configurations which can be recorded or generated and then tested for suitability. There are a very large number of possible configurations, and it is simply not possible to search the solution space blindly. The search has to be massively constrained. Part of the knowledge elicitation process was to get the technical editors (the domain experts) to expose these constraints. McDermott determined that the experts:

(a) have a highly reliable, if sparse, picture of their task domain, which they describe in terms of the subtasks involved and the temporal relationships between these subtasks.

They describe 6 major subtasks:

1. determine gross errors in the order;
2. put the appropriate components in the cpu cabinets;
3. configure the boxes and the components in the boxes in the unibus expansion cabinets;
4. put the panels in the expansion cabinets;
5. lay out the system on the floor;
6. do the cabling.

(b) have a great deal of detailed knowledge about how unconfigured components (ones that have not yet been assigned positions etc.) and particular partial configurations can be extended in particular ways.

He found that it was relatively easy to express this task knowledge as rules. Originally there were about 100 concerned with which subtask was to be initiated and about 400 rules concerned with situations in which some partial task was to be extended. The design process did contain a certain amount of backtracking as a state was transformed into the succeeding state. What McDermott did was to slightly redesign the process of transforming from state to state (from partial configuration to slightly less partial configuration) so that a possible solution is always available at any stage and it will never be necessary to undo that solution and retry. This seems to be possible since certain features of the domain are not too closely constrained. For example, a power supply does not have to be fully loaded; a panel does not have to be totally filled up, and nor does the space in a box; a data bus does not have to have the maximum number of devices attached to it. Search can be eliminated by providing a generously wide path.

The major stages of configuration are known in R1 as contexts. The word "context" has been used with a variety of meanings in knowledge engineering. It usually refers to some mechanism which provides some degree of focusing of rule use, and involves metalevel knowledge, which might be explicit or implicit. In R1, this metalevel knowledge is about configuration stages. In other systems, it may be about other aspects of the problem. As is explained in Chapter 8, the MYCIN medical system contains metalevel knowledge about the types of objects relevant to a consultation about a patient with a microbial infection, and this knowledge is represented in a "context tree". As far as the present discussion goes, the important points are that both the types of knowledge elicited from the domain experts have been represented in the R1 system, and that different expert systems may have different architectures depending on their application domains.

R1 is a successful system, and has to date processed about 90000 orders, each one taking about 2.5 minutes, including the printing of the results. Its success has been facilitated by the careful design of OPS5, particularly by the fast matching provided by the Rete algorithm. However, a great deal of its success is due to McDermott's knowledge engineering skill. His elicitation and redesign of the experts' heuristics have made it a realistic system - not the rather limited representational power of OPS5.

R1 has now grown to 6200 rules, of which approximately 50% change every year (Soloway *et al.*, 1987). Its performance continues to be satisfactory, but it is becoming increasingly difficult to change. It seems that the problems of updating this large piece of software are growing more than linearly. Soloway and his research partners at DEC attribute this to:

(1) the dynamic properties of rules. To obtain the required sequencing, "tricks" have been used to override the domain-independent conflict resolution strategy of OPS5. This explicit domain-dependent control knowledge is encoded as extra clauses to rules, and can be hard to

understand by the different programmers working on the rule-base.

(2) the static properties of rules. The action part of an OPS5 rule can be almost anything. As pointed out in section 5.4.3, great care is needed to avoid unwanted side-effects. When a new device is available from DEC, what has happened is that the programmers have often created a new rule to represent the system's knowledge of that device by editing a pre-existing rule for a similar device, possibly without always being sure what the rationale for all the functions is.

This implies that what software engineers call a "degradation in integrity" is occurring in R1's rule-base. Parnas (1985) reports:

> That example is always the same – a program designed to find configurations for VAX computers. ... Recently I read a paper that reported that this program had become a maintenance nightmare. It was poorly understood, badly structured, and hence hard to change.

What Soloway *et al.* are proposing is a design for a re-implementation of R1 (XCON). The knowledge base is to be re-expressed in a language called RIME, which will then be compiled into a runnable OPS5 form. It will be possible in RIME to make domain knowledge more explicit, both in structuring the rules themselves and in controlling rule firing, in a similar way to that suggested and implemented by Clancey (1983) with respect to the knowledge in the MYCIN systems. The new system (XCON-IN-RIME) will be built with the help of the rule developers from DEC (by metaknowledge engineers?)

The reader might find it of value to compare this short case study with that done in Chapter 8 of MYCIN. Similar problems of metalevel control and the encoding of different types of knowledge within a homogeneous rule syntax seem to have come to light. At a lower level, the Rete algorithm of OPS5 is paralleled by fast hashing algorithms underlying EMYCIN (van Melle, 1981) and Interlisp (Kaisler, 1986).

5.5 Summary

5.5.1 Advantages of Production Systems

(1) Production systems exhibit useful modularity, in that rules are independent of each other, and of the rest of the system. Each rule encodes a 'chunk' of independent domain knowledge.

(2) The explicit representation of rules permits the system to allow enquiries about rules, such as what rules would indicate a particular conclusion.

(3) The straightforward if-then form of a rule often maps well into English, for purposes of explanation.

(4) Simple chaining methods can be used to implement inference procedures which are not unlike those used by humans. Production rule systems can be built which seem to model closely human problem solving in some domains.

(5) Very large rule-based systems can be built which model expert behaviour in narrow domains, e.g. medical diagnosis and computer system configuration.

5.5.2 Disadvantages of Production Systems

(1) For each rule, information has to exist in the system somewhere as to its context of use. This can result in overlarge rule antecedents, or in implicit knowledge, such as that contained in rule order. Either way, control knowledge is often not clear.

(2) Rule sets have no intrinsic structure, which makes management of large knowledge bases difficult.

(3) Not all human problem-solving methods are easily represented in the production method formalism.

(4) The matching involved in the match-select-fire is an inherently inefficient computational process. This has serious implications for realistic applications.

(5) Because of the independence of the rules from each other and from the control strategy, it is all but impossible to determine rigorously properties of the system's behaviour by static analysis. It is necessary to test the system with the data of interest to see what it will do. Since realistic production systems cannot be exhaustively tested, they cannot be used in safety-critical applications.

5.5.3 Further Reading

Introductory texts

Young (1987) gives a simplified account of the field with various clearly-explained examples. Hasemer (1984) fully describes a Lisp implementation, with special reference to matching and conflict resolution.

Research texts

There is a noteworthy collection of papers (Waterman and Hayes-Roth, 1978), now rather dated. Brownston *et al.* (1985) and Buchanan and Shortliffe (1984) describe recent research.

Human problem-solving and its modelling

Simon and Newell's work is well-documented (Newell and Simon, 1972). A more recent reference is Klahr *et al.* (1986).

Applications

One of the largest users of production systems technology is Digital Equipment Corporation (DEC). See Kraft (1987) and Polit (1985).

New architectures

Both hardware and software architectures are being designed. See Lehr and Wedig (1987) and Rosenbloom *et al.* (1985).

5.6 Concluding Remarks

Production systems have a long and respectable history as a knowledge representation formalism. They have been used for the modelling of human cognitive processes and as an implementation language for knowledge-based systems. It has been possible to abstract from the implementations guidelines and metrics which are being used to design new hardware and software architectures to embody problem-solving strategies.

6 Psychological Studies of Knowledge Representation

Tony Conway and Michael Wilson

6.1 Introduction

The only model we have for a working intelligent system which uses and represents large amounts of knowledge is the human. This chapter describes psychological studies which investigate some of the knowledge representation schemes suggested as structures for representing human knowledge. By including a psychological viewpoint on knowledge representation it is being argued neither that the human should be a model for machine representation, nor that artificial intelligence studies should be the model on which psychological descriptions should be based. It is, however, assumed that an exchange of ideas would be fruitful between two fields where an understanding of knowledge representation is desired, both to examine possible representations and to identify phenomena against which to evaluate them.

The first section of this chapter will outline the motivations and concerns that guide psychological studies, so as to provide a background in which to place the remainder of the chapter. The second section describes various forms of representation that have been suggested for human knowledge of procedures, semantics and images which specify control and representation to different degrees. The third section then describes the use of reasoning by humans and the representation formulated as *mental models*. This approach illustrates how more than one type of representational format can be combined to represent the knowledge required to support the inferencing demanded by a variety of tasks.

6.1.1 Methodology in Cognitive Psychology

Common-sense psychology provides explanations for people's actions in terms of motives and desires. In contrast, explanations in cognitive psychology, are phrased in terms of the mental processes and representations drawn on during the performance of tasks such as problem solving and comprehension. Unlike explanations in common-sense psychology, explanations in cognitive psychology are presented as theories and models which can be tested experimentally.

Experiments designed to test hypotheses about knowledge representation do so by testing differences in the performance of tasks when some aspect of those tasks is manipulated. Since such experiments are performed by measuring behavioural phenomena, models are described in terms of the constraints of the experimental situation as well as the theoretical mental processes and representations. For example, experiments that test the representation of conceptual relationships such as 'DOG is a MAMMAL' will actually test subjects' performance on tasks involving the confirmation of statements such as 'a dog is a mammal' and the disconfirmation of statements such as 'a dog is a fish'. For models of performance on such tasks to yield testable hypotheses, they must include accounts of the decisions and the responses which are made, as well as the representation of knowledge. Therefore, it is these models of task performance, rather than the theories of knowledge representation, which are actually being experimentally tested. Consequently, much of the psychological debate about knowledge representation will be a debate about details of the models of experimentally testable performance, rather than abstract representation schemes themselves. It has even been forcefully argued (Anderson, 1978) that it is impossible to evaluate any claim for a particular sort of representation unless the processes that operate on that representation are specified in the theory. In the descriptions which follow, these issues of task performance will be avoided as much as possible, but it should be borne in mind that they provide the basis for any statements that are made about knowledge representation.

Different experimental tasks offer different views of the underlying knowledge representation. Therefore, models developed for different techniques will be models of the cognitive system from different viewpoints. Theories will, consequently, also be presented from different viewpoints on knowledge representation. If these theories use different terms, it does not follow that they are incompatible, merely that they focus on different aspects of the cognitive system. In cognitive psychology (as in other sciences) there can, of course, be more than one theory that explains the data: no observations can ever establish definitely that a single unique theory is the correct one, although the converse is, of course, true.

6.1.2 Levels of Description

In order to interpret psychological theories it is necessary to understand the level of description they offer, not only of the data, but also of the cognitive system.

When providing explanations of psychological phenomena there are at least four possible levels of description. Firstly, there is a general competence level. Descriptions at this level include linguistic theories of grammar that describe the knowledge which an individual may tacitly hold about a language, but do not describe how an individual puts that knowledge to work in speaking and understanding. An example of an explanation at this level in the field of computer science would be the theory of possible database structures. Secondly, there is an algorithmic level of description. A computational example at this level would be a specification of the algorithm for a relational database. Thirdly, there is an implementational level description. This would describe the details of the algorithm as implemented in a particular program. Fourthly, there is an implementational description which includes details of the substrate in which the implementation is made. In a computational example, an explanation of the structure of the circuitry on which a particular algorithm for a database is implemented would be at this level. A psychological example of this level of explanation would be a theory of visual perception which specifies the neurophysiological structures that perform the required computation. Newell (1982) has also argued that a theory of cognition in a particular domain first demands a theory of the domain itself, which he calls the *knowledge level.* This is not a level on the same dimension as the four levels of description of process or representation, but is a requirement for a theory of content. In Newell's terms, the levels distinguished here are all at the *symbol level* since they appertain not to the content of the information that is represented but the form of representation described.

Most theories in cognitive psychology are described at the algorithmic level, in that they draw functional distinctions between mental processes and explanations, without giving details of the exact implementation or the neurophysiological structures. For example, three forms of mental representation are generally posited. The first is a propositionally based approach in which knowledge is assumed to be represented as a set of discrete symbols or propositions. The second is to use an analogical representation in which the correspondence between the represented world and the representation is as direct as possible, traditionally using images and other analogical representations. The third form is a procedural representation in which knowledge is assumed to be represented in terms of active processes or procedures, directly interpretable by action systems. This distinction is one at the algorithmic level, since at the implementational level everything could undoubtedly be reduced to a common code in the language of the brain, just as the data structures of high level programming languages can be reduced to patterns of bits in the machine code of a computer. There has recently been great interest in the proposal for a common code of representation (McClelland, Rumelhart and the PDP Research Group, 1986) in terms of parallel distributed processing. It is an issue of debate (see Broadbent, 1985; Rumelhart and McClelland, 1985) whether this description lies at the algorithmic or the implementation level; however, it will not be discussed further here. Within the algorithmic level of description, most theories in cognitive psychology are built on the view that the human is an information processing device.

6.1.3 The Human Information Processing Paradigm

The currently dominant view of cognitive processing is that it proceeds in a linear, sequential fashion through a series of stages (Norman and Bobrow, 1976). Details of the stages vary from one author to another, but the general assumption is that processing in task performance proceeds from the perception of cues; the processing of these cues in a short-term memory to retrieve action plans from long-term memory and then the execution of plans by effector systems responsible for the articulation of sounds or physical movement.

A more detailed description of the working model would be that signals (auditory, visual, tactile) are received by transducers, which transform them into a form which can be stored in a temporary sensory information store. Pattern recognition processes then attempt to identify the physical signals by matching them against stored patterns in long-term memory. If a match is found, then a word or concept which identifies that pattern will be stored in short-term memory (STM). Without this information being actively maintained in STM (by methods such as its rehearsal) it will be lost within a few seconds. The concept will be processed to construct a description which will be used to retrieve items from long-term memory. After processing, a plan will be passed to effector systems where it will become an action. The perceived concept and a record of processing may themselves be encoded and stored more permanently in long-term memory than in STM.

There are several memory systems used by the general human information processor. The perceptual memory systems are very short term stores of the transducers' output. The short-term memory is more complex. Early experiments (Miller, 1956) illustrated that educated adults can repeat back about seven digits, words or letters. This storage could be increased if there was a structure to the items which permitted their chunking into categories. For example, twenty words could be remembered instead of seven, if there were five words from each of four categories. This initial description of a device with a limited capacity which can apparently be increased by the imposition of categorization has been developed so that contemporary theories (for example, the working memory model of Baddeley, 1983) include not only the main short-term memory (or central executive) but also limited capacity stores for verbal material (an articulatory loop), and spatial imagery (a visio-spatial scratchpad). Similarly, it has been suggested that the long-term memory store can be divided. Tulving (1972; 1984) has suggested that in our long-term memory we have both a memory for procedures and a declarative memory which is further split between episodic knowledge and semantic knowledge. Episodic knowledge concerns temporally dated episodes or events, and temporal-spatial relations among them, whereas semantic knowledge is information which a person possesses about words, their meaning, and rules for the manipulation of symbols, concepts and relations.

Much of the research on the representation of knowledge in short-term memory has focused on issues about the sensory form in which that information is perceived and processed. In contrast, most of the research on long-term memory has focused on how it is encoded, indexed and retrieved. Some of these studies and the models that result from them will be discussed in the remainder of this chapter.

6.2 Suggested Knowledge Representation Schemes

Several forms of knowledge representation will be described along with the arguments as to their relevance to the general model of the human information processor outlined above. It was noted above that, at the algorithmic level of description, there are generally suggested to be three forms of representation: procedural representation; propositional representation; and analogical representation. A complementary distinction between forms of knowledge representation is in the way that the control of the representation is handled. The first form of representation discussed focuses on the control of the representation while specifying a very general structure for that representation itself. The second form discussed is a procedural representation which combines the control and representation into a single form. The other forms use: propositional representations with other levels of more abstract content patterns to organize and index the propositional information (schemata and frames); lower level propositional representations of concepts (semantic nets and semantic feature models); and analogical representations. The focus of these forms of representation is on the structure of the representation rather than on their control. The later sections will describe an approach which suggests how a combination of these approaches can be used to represent the knowledge used to make inferences in a variety of tasks.

6.2.1 Headed Records

The Headed Records approach to memory has been proposed as a powerful framework within which a wide range of data and observations can be encompassed (Morton, Hammersley and Bekerian, 1985). In particular, the framework is aimed at encompassing observations of day to day remembering and forgetting. The model of memory suggested is very simple in principle. It consists of a set of discrete records into which our experience has been divided by some means. Each record is associated with a heading. The heading is made up of a number of distinct elements, not necessarily related or of the same kind. Thus, the heading for a record about a particular individual may include his name, a representation of his face, and his relationship to the owner of the record. Single events may be encoded in more than one record, but there are no explicit pointers from one record to another to indicate a relationship or continuation.

These records are accessed much as a file is accessed in a filing cabinet. To locate a file of which a description is known, a drawer is searched by reading the headings on the files until a heading is found which matches that description; then that file is retrieved for further examination. Within the headed records framework, the demands of a specific task are turned into a retrieval specification. This is an intermediate stage in which relevant material is assembled. The specification can include the purpose for which the information is required, such contextual information as one has concerning the conditions under which the information sought was originally encoded, and a more or less complete description of the information being sought. Unlike other theories where descriptions have direct access to record contents (e.g. Norman and Bobrow, 1979; Williams, 1978), in the headed records framework the description is used to match with only the headings of memory records. Not all the elements in the heading need be matched by the description, nor all the description be found in the heading. When the heading is matched, the consequence is that the record is accessed. The information in the heading will not be made available for further processing; the heading is solely a means of accessing the record. When a record is accessed it has to be evaluated for suitability according to criteria established with the retrieval specification. As a result of this evaluation, the record may be judged to be the one sought. Alternatively, information in the record may be used to refine the description or the verification process; then the record will be rejected, and the cycle recommenced.

One feature which distinguishes the headed records approach from most other attempts to describe complex remembering using semantic nets or frames (e.g. Norman and Rumelhart, 1975; Anderson, 1976) is that there are no direct connections between records. When the system is operating to narrate a story which covers several records, each record will provide the information required to produce a suitable description for the next one in the sequence. This information will always be content based, and never a direct internal pointer. There are two further principles of the headed records framework which make it unusual. Firstly, unlike many other theories that demand overwritable or decaying records (e.g. Loftus and Loftus, 1980), once a record has been laid down there will be no loss or change to it other than what might be called 'physiological decay'. Secondly, unlike several other accounts, in the headed records framework, when memory is being searched for a record, headings are scanned strictly in sequence from the most recent backwards in time.

This simple model of memory and knowledge representation is able to account for many of the phenomena associated with remembering and forgetting information. A full description of these phenomena cannot be given here, but an account of one may make the operation of the memory model more clear.

It has been suggested that memory representations for scenes and events can be altered by subsequent presentations of misinformation concerning what had been presented. In one study (Loftus, 1975), subjects were presented with a filmed car accident. Later they were told that a barn had appeared in the film. Although the barn had not appeared in the film, over 17% of the subjects, when questioned a week later, agreed that they had indeed seen a barn. This compared to 3% of agreement by subjects who had not been given this piece of misinformation. This result may be explained by a representation scheme which permits the overwriting of the record of the original event by the misinformation. It can also be explained in the headed records framework by assuming that when the misinformation is given, a new record is laid down describing the accident. When the subjects are later questioned, they search their memory starting at the most recent events and locate the record containing the misinformation before the original record of the accident. A second study (Bekerian and Bowers, 1983) using a similar technique and materials has shown that when sufficient cues are given during the questioning to facilitate subjects' access to the original memory, that memory is retrieved instead of the misleading information. This evidence is consistent with the original record being maintained, and inconsistent with the view that the original record is updated. This example also illustrates two general mechanisms for forgetting in this framework. Firstly, that new records are laid down after an event with headings that satisfy descriptions which would be created to access the original record, and yet the new records do not contain the detail in their bodies that the originals did. Secondly, that the description used at a second retrieval may not be as rich as that used at the retrieval that illustrated that the knowledge was encoded, and subsequently could not access the relevant record.

6.2.2 Procedural Knowledge

Most of our knowledge is declarative, in that it makes statements about the world. For example, a statement of the form "This chapter was written by two authors" is a typical declarative statement. Knowledge about how to change gear when driving a car is a typical piece of procedural knowledge. We can generally describe our declarative knowledge, as it tends to be accessible, but procedural knowledge is rarely accessible or describable. Thus, although we can change gear in our cars when driving, in order to describe how we do it, we have to imagine the movements of the foot on the clutch and the hand on the gear stick, and enact the procedure. Then we can describe this enactment. We do not have access directly to the knowledge we use when performing the task. One can obviously represent a procedure for performing a task as a declarative sequence of propositions. Therefore, the feature of procedural knowledge that distinguishes it from declarative knowledge is that it cannot be retrieved in the same form as declarative knowledge. This distinction is therefore one concerning the control of the representation as well as the representation itself.

One can separate the control and the representation mechanism for procedural knowledge to give rise to the required effects. For example, one could represent procedural knowledge in the headed records framework by having headings to the records for procedural knowledge that were inaccessible to descriptions produced for the task of describing the contents of those records. Alternatively, one could allow access to procedural knowledge by the process that verbalizes descriptions, but those records would be written in a code that this process could not interpret. However, the major form of representation suggested for procedural knowledge is the production system (Newell, 1973) which uses a procedural representation rather than a separation of representation and control. Productions are active data structures that sit above a database (or 'working memory') waiting for patterns relevant to them. Whenever such conditions occur for a production, it will be 'triggered', and perform its actions. These actions usually involve writing something to working memory, deleting or changing items in working memory. These actions will set up conditions which will allow other productions to 'trigger'. A production therefore contains a procedure to be enacted and a representation of a control structure, thereby containing a combination of the two elements required to differentiate procedural from declarative knowledge. The common data structure in a production system is usually called the 'working memory', and the 'condition → action' relation is stated within each production (or production rule) and commonly has the following structure:

IF *condition-for-triggering* → THEN *do-these-actions.*

The architecture of production systems with a working memory and a body of production rules is argued to match that of human processing. Consequently, various models of human processing have been proposed which incorporate production rules (e.g. Anderson, 1983; Kieras and Polson, 1985). Three properties of production systems in particular have been equated with aspects of human processing. Firstly, working memory may correspond to short-term memory, but the unlimited size of working memory in such models, and the actual size of working memory required to get production systems to work correctly, far exceed estimates for human short-term memory. Secondly, production rules are modular, which can permit their addition and deletion from carefully structured systems without affecting other knowledge or control structures in the system. This property has been employed to model learning and the acquisition of new skilled procedures (e.g. Anderson, 1983). Thirdly, the control mechanism of production systems permits a conflict between rules with the same conditions, or subsets of one another's conditions, as to which should act. This conflict has been used to model the failures in skilled behaviour exhibited by humans when we select the wrong skill routine. For example, when one hears the door bell and the telephone ring at the same time, picks up the telephone and says "Come in". Such action sequences can be described by the firing of a production rule to lift the telephone and then another to respond to the door, because its conditions appear to be met, although the resulting actions are inappropriate. There are various conflict resolution procedures that can be used by production systems to order the operation of rules which give rise to, or avoid, such conflicts (e.g. the computer language OPS5). However, to be experimentally testable a particular production system must have such details exactly specified. There is a body of research which investigates particular production systems (e.g. Anderson, 1983) but this does not bear on the potential of production systems in general for representing knowledge. The major limitation on production systems in general as models of human knowledge representation remains the lack of a limitation on the size of working memory. Until this fundamental inconsistency is overcome, production systems remain a form of representation which alone cannot represent the architecture of cognition, but must be incorporated with other mechanisms and forms of representation. The most obvious forms of representation which must be included are those which will account for declarative, verbally describable knowledge, such as schemata or semantic nets.

6.2.3 Schemata and Frames

Schemata and frames are discussed at length in Chapter 4 of this volume: in this chapter, they will only be discussed as possible forms of human knowledge representation. Although the idea of schemata has roots which go back as far as Kant (1787), its introduction to psychology was through the work of Bartlett. In one of his most famous studies (Bartlett, 1932) he used a story based on a North American Indian Legend 'War of the Ghosts'. He gave this story to people to read and then tested their recall for it after various time intervals. Bartlett was concerned with the systematic errors which non-Indians made in recalling the story (he deliberately chose a story which did not fit with the cultural pre-conceptions of the subjects in his experiment). His subjects forgot aspects of the story which were incompatible with their knowledge.

To account for his findings, Bartlett proposed that when the story was read, subjects recruited abstract representations of knowledge which are generally used for encoding and retrieval. These abstract representations are not tied into any specific event knowledge; they are called schemata. Secondly, he proposed that they created a schematized representation of the story. In this version, the "irrelevant" event-specific information will have been lost and schematic default information will have been assumed to apply and be stored as being present. To be confusing, this schematized representation of an event is also called a schema (e.g. Bartlett, 1932; Rumelhart and Ortony, 1976). In general, it is the deduced schematized representations of an event which are cited as evidence for the existence of schemata. There is evidence that schematized representations exist (Bransford and Franks, 1971; Bartlett, 1932; Owens, Bower and Black, 1979; Friedman, 1978; Galambos, Abelson and Black, 1986). However, schematized representations could arise through the type of post-event restructuring of knowledge suggested in the headed records approach. The more problematic and significant issue is whether abstract schemata exist.

The most influential introduction of schemata into the AI community has been as frames (Minsky, 1975), which were developed to show how knowledge should be interrelated so that computational systems could use knowledge efficiently. However, the form which has resulted in most psychological study is that of scripts (Schank and Abelson, 1977). These were developed to account for the ability of readers to fill in information required to understand the simplest text. For example, to understand the two sentences 'Tony sat down in the restaurant. The waiter took his order.' we need to have a lot of knowledge about restaurants. We need to know the role of the waiter, and that 'ordering' refers to a request for food prepared in restaurants. Schank and Abelson referred to such social knowledge frames, or schemata, as 'scripts'. Scripts represent performed, ordered sets of knowledge about stereotyped cultural events. There would therefore be a script for visiting the restaurant, the doctor or the dentist. The possession of a script allows a speaker to leave many things unsaid with the certainty that a listener will fill them in by default. If enough is stated to elicit the appropriate script then it can be used to fill in unstated detail. Since scripts only describe stereotyped events, a separate mechanism was envisaged which would create plans of less usual events.

Among the psychological studies following this work, Bower, Black and Turner (1979) demonstrated that when people read a story about a visit to a dentist and a story about a visit to a doctor they were confused in their later recognition of which events were in which story. They found that such confusions were situated within similar scenes across different scripts. This led Schank to reform the notion of scripts so that they describe smaller units, such as the paying scene, which would apply to various situations, or the waiting room scene which would apply to visits to both doctors and dentists. Schank (1980) proposed that, instead of scripts, we have many general scenes in memory called MOPS (Memory Organization Packages) which are dynamically assembled into higher level structures. These are built into structures that resemble scripts, but also account for the memory confusions found experimentally. However, the main limitation on scripts remains that they only specify knowledge about events that are stereotyped, whereas knowledge is also used to understand events and discourse which are *not* stereotyped.

One area of knowledge where stereotyping is less problematic is the representation of word meaning. Berlin and Kay (1969) asked native speakers of twenty different languages to select coloured chips which represented the best examples of each of their language's basic colour terms. They also asked their subjects to select chips which delineated the boundaries between colour terms. There was very little consistency in the choice of boundary chips; however, there was a reliable consensus about the choice of the best exemplars of a colour. Indeed, this agreement extended across many of the twenty languages. This, and subsequent studies by Rosch (e.g. Rosch, 1976), have been used to argue that many natural categories are mentally represented by *prototypes*. These prototypes are schemata of a category's most characteristic members: in the way that a robin is a prototypical bird, whereas other birds have a greater distance from the schema, e.g. a chicken. Although this approach has been developed for 'kind' notions (like *dog, bird,* and *animal*), 'artifact' notions and simple descriptive notions (e.g. 'triangular'), it has not been extended to intricate concepts such as *belief, desire,* and *justice* and it is an open question whether or not the theory can be extended to cover these cases. The second problem with prototype theory lies in the mechanism for conceptual combination. Methods such as fuzzy logic have been suggested for the combination of prototypes, but these seem to result in as many new problems as they solve (see Osherson and Smith, 1981).

Schemata offer high level representations and rules about representation-specific processes. It is argued by their proponents that these are required to supplement the descriptions of processes and representations which are derived from a small number of general principles. The experimental evidence from psychological studies of schemata supports the view that representations are constructed which appear to be schematized. However, there is little evidence for the existence of abstract schemata themselves. Although there are effects which are best explained by stereotyping and the use of defaults, the size of the units that are stereotyped is not certain. Although such problems still exist with the suggested human use of schemata to represent knowledge they do not detract from the potential of schemata as a form of representation for machine use. The problem of the combination of concepts which are represented as schemata or stereotypes and the use of stereotypes to represent non-stereotypical concepts are problems both for the psychological representation of stereotypes and machine representation. These issues are discussed in Chapter 4 of this volume.

Schemata and frames have been suggested as top-down mechanisms to represent general high level knowledge, and prototypes have been used to represent concepts. A second class of representations which use bottom-up processes to represent concepts and word meanings includes the semantic net, and semantic feature models.

6.2.4 Semantic Nets and Semantic Feature Models

The 'semantic net' was developed by Quillian (1966) and others both as an exercise in artificial intelligence and as a possible psychological model of human associative memory. Semantic nets are an extension of the well established idea in psychology of associations. In behavioural psychology these took the form of associations between stimuli and responses, but the best known example is that of word association. Quillian argued that among the properties of concepts were several special property relations that are commonly found. They are special because they permit certain kinds of inferences to be made. A frequently used kind is the *superset* or superordinate relation (e.g. a mammal is an animal). These superset relations will chain like: dog → canine → mammal → animal → living thing → object. Each item in such a structure is termed a 'unit' which can have properties attached to it. A property represents some descriptive feature of a unit, such as would be represented by an English verb phrase. Thus the unit MAMMAL might be linked with a number of properties, such as *has hair, provides milk* and so forth. Each property is stored in the highest level to which it applies. Hence *has hair* is stored with MAMMAL rather than with each individual instance, thereby reducing the amount of storage space. All properties of a superset can also hold for the instances of that superset unless otherwise indicated. In the case of an exception, the fact that the property does not apply is stored with the unit itself. Therefore, to determine a property of a concept a simple three-step procedure can be applied which avoids conflicts which could arise from inconsistent data in a network:

Step 1. In determining properties of concepts, look first at the node for the concept.

Step 2. If the information is not found, go up one node along the relation and apply the property of inheritance.

Step 3. Repeat step 2 until either there is success or there are no more nodes.

This work attracted little attention until a series of experimental studies by Collins and Quillian (1969). The model was tested by presenting subjects with a series of sentences and measuring the time taken to decide whether they were true or false (reaction time). The model predicts that reaction time should depend, first, on the number of levels of the hierarchy that must be traversed (e.g. *a dog is an animal*) and, second, on whether or not a property must be retrieved (e.g. *a dog has hair*). As predicted, reaction time increases linearly with the number of levels of the hierarchy that must be traversed, in that it takes longer to decide that *a canary has feathers* than that *a canary is yellow*. Although this evidence supports the network representation, there is evidence against this simple view of processing. In a hierarchical representation of the sort suggested, the decision that *a pine is not a flower* would be made by finding that there was no path joining the two items. This would imply that the nature of the negative instance should be unimportant, so long as no legitimate path exists. However, as Schaeffer and Wallace (1969) and Wilkins (1971) showed, it takes longer to decide that a *pine* is not a flower, than that a *chair* is not a flower, suggesting that some sort of discrimination takes place, even though no permissible path exists. The more features a negative instance has in common with a category, the longer it will take to reject. A second body of evidence against a hierarchical model is that it fails to predict differences within categories, while such differences have often been found. Subjects can verify a dominant or typical member of a category consistently more rapidly than they can in the case of a less typical one. That is to say, it takes less time to verify that a *robin* is a bird, than that a *chicken* is. A third problem for the simple hierarchical model comes from a study by Rips, Shoben and Smith (1973), who showed that subjects took longer to decide whether some items were members of the class MAMMAL than to decide if they were 'animals' despite the fact that MAMMAL is a subset of the class ANIMAL. This probably reflects subjects' greater familiarity with the concept of an ANIMAL than the concept of a MAMMAL, since these judgements correlate with the rated semantic distance between an instance and its category.

Two alternative classes of representations arose to account for this evidence against the early hierarchical networks as models of human representation. Firstly, more complex theories of processing were developed for semantic network representations themselves which would be capable of representing sentence meaning as a network of labelled associations rather than just being a simple hierarchy supporting inheritance. Secondly, Smith, Shoben and Rips (1974) proposed a 'feature comparison' model as an alternative to a hierarchical structure, using inheritance as a model of the representation which supports performance in the class of tasks investigated.

The basic representational assumption of this model is that words representing categories can be represented by a set of features that vary in their relationship to the formal definition of the category. Features are of two types: 'defining features', must be true if an item is a member of a category; 'characteristic features' usually apply, but are not necessary for a definition. Thus 'has feathers' is a definitional feature for the concept BIRD, whereas 'can fly' is a 'characteristic feature' (although most birds fly, it is not part of the definition since some birds do not show this characteristic). They suggested that category membership was not a pre-stored attribute, but was computed by a comparison of a set of features. They proposed a two-stage model of the verification of category membership. The first stage involved a quick comparison of all the features, both definitional and characteristic. If the comparison was good enough, the statement would be confirmed. If the comparison was poor enough, the statement would be rejected. Intermediate comparisons would result in a slower comparison process using only definitional features. This model accounts for the basic experimental results in that: true statements using items typical of categories are quickly confirmed; false statements involving typical items are quickly rejected; decisions on less typical items take longer.

Although feature models offer good accounts of the experimental data, they are almost always limited to nominal concepts, and it is not clear how such models could represent propositions. Although semantic feature models were intended to account for propositions, this inability is a limitation when compared to recent semantic networks.

The second development following the early hierarchy representations of concepts was to add more complex processing mechanisms to semantic networks in order to enable them to be able to represent sentence meaning as a network of labelled associations. The major processing mechanism proposed was a development of Quillian's notion of 'spreading activation' (Collins and Loftus, 1975).

The semantic network is a highly interconnected structure with relations connecting together nodes, very much like the transport links connect together towns and cities. 'Activation' is an abstract quantity which represents how much processing is taking place in the structure. If a network representing the structure of animals were used to answer the question "Does a dog have hair?", the nodes for DOG and HAIR would both become activated. The activation could then spread down the links connected to these nodes, and onto the nodes at the ends of these relational links. Activation would then spread on down the links from these nodes, and so on. If one imagines spreading rings of activation originating from each starting point, like the ripples extending away from the sites where two pebbles are dropped in a pond, these rings will eventually meet. When the activation patterns meet, a path has been established between the two nodes. The path can be found by following the activation traces, and given the nature of the path, the question can be answered.

There are several properties of activation theories of semantic network processing which have led to empirical investigation. Most research has focused on the time course of activation (e.g. Neely, 1976), or have used one aspect of activation called 'priming' as a tool to examine the details of representation (e.g. Meyer and Schvaneveldt, 1971). The theoretical assumption behind priming is that, once a node has been activated, it will take some time for that activation to decay. Therefore if a second node is accessed the spread of activation will be faster than if the first item had not been activated. This has given rise to many studies where items are presented together or in succession and the reduction in time for some decision on items is used as an indicant of the relation between them. For example, subjects may be asked to read two strings and decide if they are words or not (e.g. "nurse" "plame"). When the two words are related, (e.g. "bread" "butter"), the judgements are considerably faster than if they are not related (e.g. "bread" "nurse"). Sophisticated experiments have used stimuli with multiple meanings (e.g. "bank" with "money" or "river") or embedded words with related meanings (e.g. "cot" "ton" "cotton" "wool" "cottonwool") to investigate the interrelationships of items in memory. Using a similar method, Collins and Quillian (1970) showed that decisions are made more quickly when they require the traversal of recently used paths across the network than when the required paths have not been recently used as evidence for their hierarchical model of memory.

A second aspect of the spreading activation model which has received much investigation has been termed the 'fan effect'. This has been particularly investigated in relation to the detailed model of cognition proposed by Anderson (1976, 1983). This model makes the processing assumption that the activation that crosses a link is inversely proportional to the number of links that "fan out" from, or leave, that link. This results in the prediction that the more nodes that are connected to an item, the harder it will be to retrieve information about that item. A series of experiments (summarized in Anderson, 1983) show that when subjects are shown a number of sentences to learn, and then tested on their ability to recognize test sentences, they are slower to recognize sentences involving concepts about which they have learned other information, than those which contain items which do not occur elsewhere. These studies support the prediction derived from the processing assumption: that the more the facts, the slower the recognition time.

When semantic networks are used to represent sentences, a distinction has to be drawn between 'tokens' and 'types'. That is to say, between BOOK representing any book (the type) and 'book' referring to a particular book (the token). This distinction is required to prevent a confusion about which book is being referred to when two books occur in the same text. For example, to represent the sentence 'John picked up a book and Mary threw a book at John', there must be a node representing each of the books. A variety of mechanisms have been devised to overcome this problem, but a common one (e.g. Norman and Rumelhart, 1975) is to use general identifiers as nodes (e.g. numbers) with links to the type concept. However, this mechanism does not overcome a fundamental problem with semantic networks as models of human memory: that they deal with the connections between concepts rather than their connections with the world. This problem was well summarized by Johnson-Laird, Herrman and Chaffin (1984: 306): "Any psychological theory of meaning should account for these phenomena [the relations among intensional relations, ambiguity, anomaly, instantiation and inference]; semantic networks contain mechanisms designed to do so, but nevertheless fail to deal with them adequately, a failing that also applies to theories based on semantic features or on meaning postulates."

A variety of semantic network theories has been developed (e.g. Quillian, 1968; Anderson and Bower, 1973; Norman and Rumelhart, 1975; Glass and Holyoak, 1974; Collins and Loftus, 1975; Anderson, 1976; 1983; Sowa, 1984). Each of these theories makes specific predictions, some of which have been empirically investigated, and a few of which have been described above. But these studies relate to specific aspects of individual theories. What can be said of semantic network theories as a class? There are few features which all network theories share: they are designed to elucidate the relations between words (intensional rather than extensional relations); they assume that the evaluation of intensional relations can be considered separately from those of extensional relations; they are based on a framework composed of: a parser, a semantic memory consisting of a network of links and nodes, and a set of processes that operate on, and interpret, the network; they have a general commitment to parsimony. These four features in themselves restrict the class of network theories very little. Anderson (1976) has shown that his ACT network system is equivalent in power to a Universal Turing Machine (i.e. the processes it invokes are capable of computing anything that can be computed at all), and it is likely that this equivalence could be proved for other network theories. Although individual network theories may have testable properties, and when used in conjunction with other forms of representation and processing may overcome the problems of extensionality, as a class, they offer no theory which can be assessed in terms of psychological validity. An extreme expression of this position is provided by Johnson-Laird, Herman and Chaffin (1984: 305): "We have no quarrel with the formalism or notation of networks: A commitment to them is little more restrictive, and no more open to criticism, than is a commitment to a particular programming language such as LISP." Consequently, semantic networks remain only a form of computer implementation, as discussed by Mac Randal in Chapter 3 of this volume.

6.2.5 Analogical Representations: Imagery

The representations discussed so far employ symbolic representations of the world. In contrast, there is a class of analogical representations which are much closer to the world being represented. One of the major sources of support for the existence of this class of representation comes from the phenomenon of mental imagery. The study of *imagery* has had a chequered history in psychology, but unlike the representational forms discussed so far, the analogical representation suggested to support it has hardly been addressed as a computational form of representation in AI.

6.2.5.1 What is Imagery and Why Study it?

People, when asked to describe what they experience when they think, often say that they have the sensation of "pictures in their heads". These images are reported as varying in intensity, in the degree of detail present and in their manipulability. Generally there is no difficulty in distinguishing them from the reality of perception; i.e. these images are distinct from hallucinations. If asked to imagine say a cat, specific characteristics of the images (such as colour of fur, length of tail, size relative to an image of a mouse, whether the cat is sitting upon a mat? ...) can often be given. Frequently there is the impression of "zooming-in" on part of the image to obtain more detail.

This human ability to form images has been known and utilized for many years: for example it is the basis of the method of loci as an aid for orators (formalized by the Greek poet Simonides: where a speech is remembered as a trip through the rooms of a familiar building, and the objects in the various rooms act as cues to the next topic). Various mnemonic techniques rely upon imagery ability; and some of these have been explored by psychologists

(Paivio suggested that this accounts for the advantage of concrete words over abstract words in memory tasks, and Bower has investigated the use of bizarre associations as a memory aid). It is also likely that imagery is used routinely, and perhaps unconsciously, in problem-solving. In some of these problem-solving situations there may be practical implications: consider studies of "common-sense" or naive assumptions about physical processes (for example: Caramazza *et al.* (1981), diSessa (1982), Shanon (1976)).

However there seem to be wide individual differences in imagery ability: from spectacular examples such as those claimed for Nikola Tesla (e.g. O'Neill (1980)) to those people who report not having experienced images at all. There are also developmental complications arising chiefly from the slippery area of eidetic imagery ('photographic memory', see Haber, 1979) and its apparent relationship to verbalization (e.g. Glanzer and Clark (1964)).

Other internal vehicles for thought (such as some form of internal speech, symbolisms related to mathematics) are also reported but imagery is currently of great interest within cognitive science because:

(i) it seems to suggest a form of (knowledge) representation that is **analogical** rather than propositional in nature (and thus presents an interesting problem to workers in artificial intelligence as well as to psychologists), and

(ii) it suggests, to some psychologists, that there may be **more than one central representation** underlying cognitive processing.

6.2.5.2 Does Imagery Exist as a Real Process?

To the extent that people report such experiences, "imagery" exists as a psychological phenomenon, and is therefore worthy of investigation. It is, however, difficult to experiment on such mental processes and to produce acceptable behavioural data for sceptical colleagues.

At one stage in psychology it was hoped that physiological approaches to psychology might come to the rescue by providing correlates between the electroencephalogram (EEG) and various psychological traits. This followed closely from the hopes of such pioneers of the EEG as Berger (1929), in one of the first papers on the human EEG, that the techniques would prove useful for psychiatric diagnosis.

One exploration attempting to relate "imagery types" to EEG alpha rhythm (electrical activity in the range 8-12Hz: tending to dominate the occipital scalp when the subject is relaxed with eyes closed) was carried out by Golla *et al.* (1943). They classified their subjects into groups as (self-reported) visualizers or non-visualizers and into three EEG-categories: persistent-alpha (those subjects who produced alpha when relaxed with their eyes open or closed), responsive (subjects producing the 'normal' responses of alpha when their eyes closed but not when their eyes were open), and alpha-minus types (those subjects who produced no alpha during the experiment). The report indicated a large number of visualizers in the alpha-minus category and non-visualizers in the persistent-alpha category. However (as shown by Oswald (1957)) very few people produce no alpha (if allowed sufficient time to relax in the experimental situation). Oswald's visualizers, when allowed to settle down, produced almost as much normal (responsive-category) alpha as his non-visualizers. Further, there is a close relationship between alpha rhythm and the activity of the visual system (as suggested by Lippold (1970) and Wertheim (1974, 1981)). This might suggest that any correlations found between some aspects of EEG and differences in cognitive style might be more correctly attributed to differences in habits of visualization.

At present psycho-physiological measures offer little evidence which can be drawn upon to support or reject hypotheses concerning high-level cognitive representations and processes: in the same way as data concerning the voltages across components in a digital computer are not appropriate to describe the current (high-level) program operation.

Psychologists are chary of subjective (introspective) reports as primary evidence, which leaves the major experimental attack upon imagery relying upon the behavioural consequences of tasks in which subjects may use "imagery" (and perhaps subjects pre-selected for such ability, trained and encouraged in its use in the experimental situation).

6.2.5.3 Pre-history of Recent Work on Imagery

In the early days of scientific psychology, imagery was a major area of study for those interested in cognition. For Wundt (1904) **introspection** was seen as the "*sine qua non* of any psychology"; although this was introspection as a controlled experimental technique with highly trained observers. However, a major problem for the introspectionist position was the description and theoretical justification of the distinction between thought involving images and 'imageless thought'.

However with the emergence of **Behaviourism** (e.g. Watson (1913)) and the predominance of a logical-positivist inspired methodology, at least in the United States, through to the 1960s, the study of such mental phenomena declined. Imagery was viewed essentially as epiphenomenal to visual processing, in much the same way that 'imageless thought' was reduced to sub-vocal movements of the throat and larynx. An interesting example of the change in approach at this time is Warden's (1924) report on the relative efficiency of using imagery, verbal coding or motor memory in human maze learning. It is also an early example of **protocol analysis** (the analysis of verbal accounts of tasks while they are being performed) which has since become an important technique in many 'ecologically-valid' (real-life) experiments. Against this general background there were only a few examples of a 'mentalistic' approach to the area before the "Cognitive Revolution" of the early 1960s. However, imagery did feature (as Spatial Visualization) in a number of aptitude tests (e.g. Army Air Force test battery constructed by Guilford *et al.* (1952)).

With the re-emergence of a cognitive psychology (ushered in by such books as Miller, Galanter and Pribram (1960)) it again became respectable to experiment on such mental processes as imagery. This was backed up in 1964 by Holt's paper in the American Psychologist - "Imagery: The return of the ostracized." Within ten years of this paper a number of major texts on imagery had been produced (e.g. Horowitz (1970), Paivio (1971), Sheehan (1972)) and a new range of experimental strategies to tackle these difficult problems had been developed (see, for example, Chase (1973)). With the emphasis within cognitive science on **representation** it has assumed an importance perhaps equal to that it had achieved around 1900. A number of earlier experiments have essentially been re-interpreted within a more cognitive frame, for example, the study by Carmichael *et al.* (1932) where there were distortions in the reproduction of line-drawings from memory after the drawings had been associated with a verbal label. From this it is suggested that images held in memory might be more malleable than perceptions. Interestingly, some of the same problems are recurring: imageless thought, the relationship to self-awareness and the problem of infinite (mental) regress.

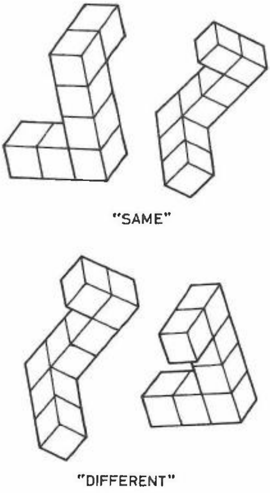
6.2.5.4 Experimental Findings: What Do We Know of the Nature of Imagery?

The following are brief descriptions of a selection of psychological studies on imagery. They have been selected as typical examples of the work in the area. For more comprehensive reviews of the area: see Kosslyn's (1980) "Image and Mind" or Pinker's (1985) "Visual Cognition" (which integrates considerations of the mechanisms supposedly underlying imagery with work on visual perception).

6.2.5.5 Mental Rotation

Shepard and his colleagues have been performing experiments on mental rotation from around 1970. In one of their early studies (Shepard and Metzler, 1971) subjects were presented with two drawings of three-dimensional objects (examples of the type of drawing are shown in Figure 1). The experimental task was to report if the represented objects were identical except for orientation. Subjective reports suggested that subjects attempted to match by mentally rotating one of the shapes until it fitted with the other. The decision-time for matching pairs suggested that the process being used seemed to be an analogue of physical rotation of the object because the greater the angular disparity between objects the longer it took subjects to give a decision. The speed of rotation seemed uniform at about 50°/second (although there are differences between subjects). A similar finding was found by Cooper and Shepard (1973) using rotated letter and mirror-image letters (see Figure 2). The subjects' task here was to decide if the letter was well-formed or backward. Again the result suggested that a mental analogue of rotation was being performed. A faster, though still uniform, rotation speed of 300°/second was found, possibly because less complex and more familiar shapes were being used. There is some suggestion from Metzler (1973) (also discussed in Metzler and Shepard, 1974) for the process being continuous. She used an image as an aid to a subsequent perceptual match task. However, there are some difficulties in her technique owing to subject variability.

Schwartz (1979), using a version of a technique pioneered by Cooper and Podgorny (1976), produced some results that, whilst generally supporting the earlier rotation studies, suggest that some refinements may be needed. The experimental design is shown in Figure 3. The results again showed that a greater rotation needed a longer time to carry out. With larger angles of rotation larger patterns needed more time to rotate than small ones but it takes no longer to rotate complex patterns than to rotate simpler ones.


"SAME"

"DIFFERENT"

Figure 1

Example configurations for one stimulus: after Cooper & Shepard, 1973.

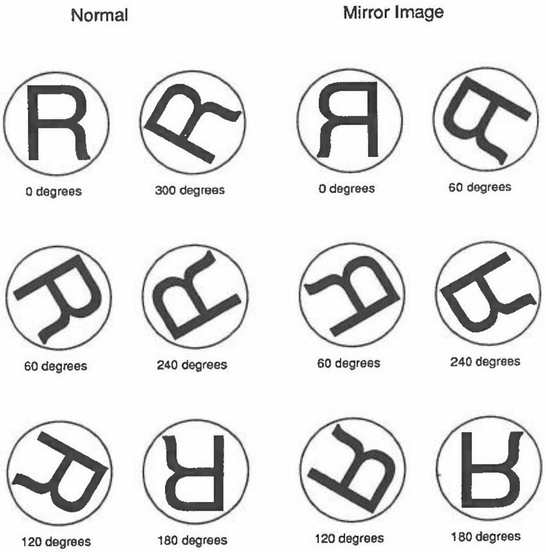


Figure 2

There was some confirmation, that subjects were actually rotating an image, from results of a template-matching task where subjects had to match their images against an actual "probe" pattern: in that responses were faster when the probe and image had the same orientation.

6.2.5.6 Mental Paper Folding

Shepard and Feng (1972) used a task in which subjects were required to make judgements about paper cubes which had been unfolded to make patterns of six squares. Some examples are given in Figure 4. The task was to determine from the two-dimensional pattern if the heads of the two arrows marked on the pattern would or would not meet if the squares were folded into a cube. Shepard and Feng found that there was an approximately linear function between the number of folds required to test for a meeting and the time taken to make a decision. That the function was linear rather than exponential in nature would be expected from the nature of the search space. This was congruent with their subjects' reports that they were mentally refolding the squares in order to solve the problem.

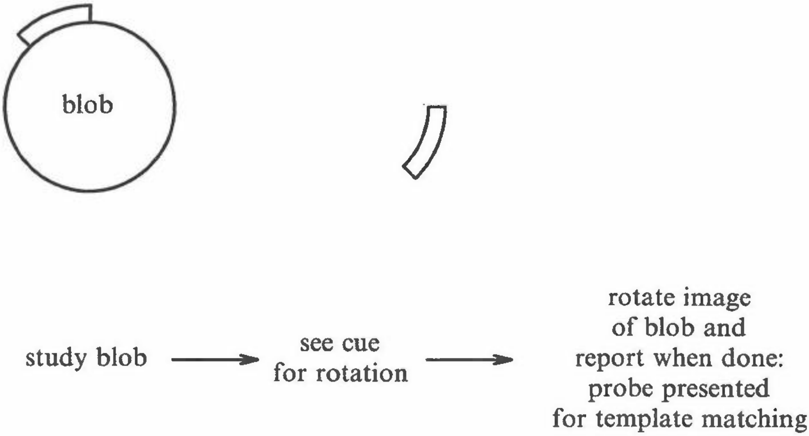
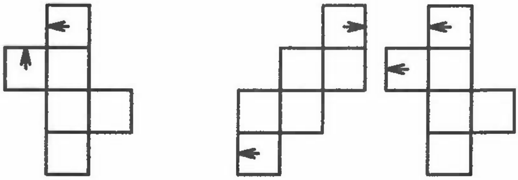

blob
study blob
see cue for rotation
rotate image of blob and report when done: probe presented for template matching

Figure 3 Schwartz (1979) experimental procedure

Do the heads of the arrows meet when the patterns are folded into cubes?



(2 folds) (5 folds) (Non-match)

Figure 4 Examples of diagrams used by Shepard and Feng (1972)

6.2.5.7 Scanning Mental Images

Kosslyn, Ball and Reiser (1978) attempted to investigate the amount of movement on, or of, an image. Subjects were given a map of a fictitious island. The island had a number of features (including a hut, a rock, areas of sand). Subjects were trained on the map until they could reproduce the drawing with great accuracy. The main experimental task consisted of the following sequence. An object on the map was named. Subjects were asked to imagine the map and focus upon that object. A second object was named five seconds later. Subjects were instructed to scan the map for this second object and to press a button when they had mentally focused upon it.

Analysis of the data supports the view that the time needed to perform this task increases linearly with the distance apart of the objects (on the real map and presumably on the image). Again this suggests that imagery is a process similar to a physical operation: an analogue.

6.2.6 The Uses of Imagery?

The examples of processes described above relate specifically to imagery rather than to the use and possible relationship of imagery to other forms of cognitive activity. Imagery can take the form of photographic recall of images, but it can also involve the mental construction, representation and manipulation of images of diagrams, and figures. With the acceptance of imagery as a phenomenon and of processes which operate on imagery, two parts of the argument that imagery is an example of an analogical representation being used in human reasoning have been established. The third part of this argument is to attempt to establish the representation upon which imagery is based and to confirm that this representation is indeed analogical in nature. To do this we need to consider the relationships between this form of representation and others which have been described in earlier sections of this chapter.

6.2.6.1 Analogue Representation, Imagery and Inference

There are examples in the psychological literature of tasks which may be helped by the formation of appropriate images. There are also examples where an incorrect image can be misleading. Additionally some operations which may be easy to perform on a picture seem hard for most people to perform on an image. (McKim (1980) is a useful source of examples in this area.)

As an example of the former:

A Buddhist monk visits a shrine at the top of a sacred mountain. It is a difficult climb. He leaves on Monday at 08.03h, reaching the top at 16.30h (with a 15 minute lunch break sharp at noon). At 08.03h the next morning he descends the mountain using the same path. He makes good time in the descent arriving at the bottom of the mountain at 13.05h on Tuesday afternoon.

The question is: Is there any time of day (you do not need to say what time) when the monk was at exactly the same point on the path on Monday and Tuesday?

In this case one can attempt to solve the problem by imaging a saffron robed monk climbing a hill, or a time-distance graph of the movement. The former image does not aid the solution whereas the second image of a graph rather than a picture can lead to a solution.

A number of psychologists working on problem solving in the 1930s and 1940s made extensive use of imagery-like processes as explanatory concepts. For example, a typical study in Maier's (1931) experimental series is the "two-string problem". Two strings hanging from the ceiling have to be tied together, but the strings are positioned so far apart that the subject cannot grasp both at once. The room contains a number of objects, including a chair and a pair of pliers, that may be used to find a solution to the problem. Maier found that subjects tried various solutions involving the chair but these did not work. The suggested solution is to tie the pliers to one string and set that string swinging (like a pendulum), then to get the second string and bring that to the centre of the room, to wait for the first string to swing close enough to grasp and then to tie the strings together. Only 39% of Maier's subjects were able to see the solution within ten minutes. In this problem the difficulty does not arise because subjects view a picture of the situation rather than constructing a diagram, but when they construct either, they do not perceive the pliers as a weight that can be used as part of a pendulum, but as a tool for gripping objects. This difficulty is called functional fixedness (fixity): so named because subjects are fixed on representing the object according to its conventional function and fail to represent the novel function.

Similar results can be found in Duncker's (1945) studies. One task that he posed for subjects was to support a candle on a door, ostensibly for an experiment on visual perception. Materials supplied for the task were a box of drawing-pins, matches and a candle. The suggested solution here is to fasten the box to the door with drawing-pins and then to use the box as a platform for the candle. This task is apparently difficult for subjects because they see the box as a container not as a support or platform. Subjects have greater difficulty with the task if the box is filled with drawing-pins, reinforcing the perception of the box as a container.

The following problem, it is suggested, is relatively difficult to solve using an image but is easy using a picture (adapted from Simon (1978)):

Imagine (but do not draw) a rectangle 2" wide and 1" high, with a vertical line cutting it into two 1" squares. Imagine a diagonal from the upper-left-hand corner to the lower-right-hand corner of the 2" x 1" rectangle. Call this line Diagonal A. Now imagine a second diagonal from the upper-right-hand corner to the lower-left-hand corner of the right hand square. Call this line Diagonal B. Consider where Diagonal A cuts Diagonal B.

What is the relationship of the length of B above the cut to the length of B below the cut?

The above problems also give good examples of the wide range of individual differences found in this area. Some people report using images ("like a drawing"), to others it is "just obvious". A number of people find it difficult to solve such problems without resort to external means (such as pencil-

and-paper). It is not clear that these differences relate well to measures of intelligence, personality or 'cognitive style'.

6.2.6.2 Imagery and Interference

Brooks (1968) conducted a series of experiments on processing of visual images contrasting performance here with performance on non-visual tasks reckoned to be of equal difficulty.

For the Visual task subjects had to scan imagined diagrams such as Figure 5. Scanning round, starting at the * and moving in the direction of the arrow, subjects were asked to categorize each corner as a point at the extreme top or bottom ("yes") or a point in between ("no"). For the figure below the correct sequence of responses should be YYYNNNNNNY. The Non-visual task was to hold a sentence, with the same number of words as there are corners in the diagram, in memory (such as " A bird in the hand is not in the bush "), and then to classify each word as a noun or not. The correct sequence of response for the example sentence is NYNNYNNNNY. Three response conditions were used (a) say "yes" or "no", (b) tap with the left hand for "yes" and the right hand for "no", and (c) point to successive Ys and Ns on a sheet of paper.

The following table gives the mean classification time in seconds:

	OUTPUT CONDITION		
	Pointing	Tapping	Vocal
Visual Imaging task	28.2	14.1	11.3
Non-visual task (sentence)	9.8	7.8	13.8

This pattern of results suggests that scanning a sheet for the responses interfered with the scanning of a mental image. The interpretation, again, is that subjects are scanning a representation that is an analogue of a physical display.

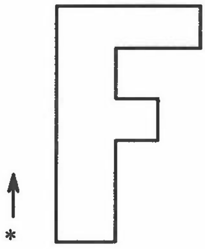


Figure 5 Example of diagram to be imagined and scanned: after Brooks (1968)

It has been suggested that Brooks' result was caused by the interference generated from having to do a visual pointing task and at the same time scan a visual image. Later results (such as Baddeley and Lieberman (cited in Baddeley, 1976)) would suggest that the problem was at a more abstract level than a visual one: that is, the interference is spatial. This latter result also points to the image being an analogue of a physical structure.

6.2.6.3 Distortions of Cognitive Maps

One topic of interest to people from a variety of areas has been how mental (cognitive) maps of the environment are formed and used. Many people report having some form of mental map, and the geographic information that would be encoded in such maps is obviously drawn on in everyday navigation around the world. However, when people are asked to draw such maps, systematic distortions of reality are often produced. For example, Boston Common is usually drawn as a square although in reality it has five sides; the standard New Yorker's view of the United States or Londoner's view of the United Kingdom distort the geography so that the part of the world the author inhabits is overly large, and other areas are reduced. This anecdotal evidence is supported by an experimental study (Milgram and Jodelet, 1976) which shows that Parisians' view the Seine as making a rather gentler arc through Paris than it does. This results in some Right Bank districts being placed (mentally) on the Left Bank. These effects could be due to a greater familiarity, recency or frequency of experience with one area than another (a useful summary of some of the work on mental maps can be found in Gould and White, 1985).

Many mental maps are based on information derived from printed maps rather than directly from experience. The persistence of distortions found in such sources accounts for some people's belief that Greenland is much larger than it actually is, because the Mercator projection used to produce flat maps of the earth distorts its size (a belief usually dispelled when Greenland is viewed on a globe). The persistence of such information is in itself of little interest, however, yet more pronounced distortions can be found in diagrammatic maps such as those of the London Underground. The London Underground map has been found to be very satisfactory for its role, and has been emulated in many similar situations. Since these descriptions are spatial in nature, they are often assumed to represent geographic distance relations. However, they actually represent the connectivity in a network. Consequently, Bayswater station is a third of the map away from Queensway on the London Underground map, whereas, spatially, it is only 50yd away. Here a spatial representation is being used to represent information which is itself not spatial. A similar translation may underly many of the distortions found in cognitive maps.

Stevens and Coup (1978) collected a set of misconceptions about American geography. They analysed their results in terms of the influence from more abstract facts about the relative locations of large physical bodies (such as individual states) producing distortions on the finer detail. They then repeated their experiment and analysis with a set of simplified maps and found the same error patterns, thereby supporting a view that the knowledge of abstract facts results in the distortions in mental maps.

Thorndyke and Hayes-Roth (1978) investigated how mental maps are developed through experience rather than from paper maps. They studied secretaries working in the maze-like Rand Building (Santa Monica, California). They found that 'route-maps', for example how to get from one's office to the photocopying room or to the lunch room, were acquired fairly easily but that it took much longer to develop 'survey-maps', which would enable accurate decisions as to the direction of the lunch room from the photocopying room (a potential route that was not used). It was suggested that secretaries had typically to have ten years' experience of the building before they developed such 'survey-map' knowledge! This finding is supported by developmental evidence (Hart and Moore (1973)) which suggests that children develop from using *route-maps* to *survey-maps*.

Although many of these studies appear to suggest that subjects are using an image (equivalent to a paper map) and can produce a drawing of such an image, backed by subjective report: there are again differences of opinion. For example: Hintzman *et al.* (1981), investigating the way that people orient themselves in somewhat familiar environments, suggest that subjects use propositions not mental images. Reed (1974) has examined the relationship between structural descriptions and mental images, emphasizing the limitations of visual images. Consequently, when information is held as a route-map, a route may be mentally represented and not a two- or three-dimensional representation of geography. These are only produced as drawings when requested by experimenters. Although maps may be viewed as images, they may be constructed from propositions rather than being stored as images themselves. Further, although mental maps are viewed in a spatial manner, the information represented is not always itself spatial in nature.

6.2.6.4 Imagery and Visual Perception: Constructed Analogues or Recalled Pictures?

As described above, a number of authors have investigated the relationship of visual imagery to other cognitive processes: e.g. problem solving (Kaufman, 1979), learning (Fleming and Hutton, 1973), and memory and cognition (Richardson, 1980, Yuille, 1983). These have shown that images that have been seen can be recalled, but also that representations can be developed to aid problem solving and reasoning. Since imagery is easiest to characterise as a phenomenon similar to vision, the most obvious process to compare it to is perception. Although there is a similarity between imagery and perception, some of the evidence as to the nature of imagery is persuasive that images are rather different from perceptions.

For example: Yuille and Steiger (1982), using a modification of the Metzler and Shepard (1974) mental rotation task, described above, have been able to find effects resulting from the complexity of the concepts imaged. This is in contrast with earlier investigations such as Cooper and Podgorny (1976), and suggests a piecemeal (feature analysis) processing system rather than an holistic process.

However, a number of workers in the area still suggest that images take on some of the properties of objects. The idea that images are just (faint) echoes of perceptions has been around for years. There have been attempts to investigate this directly. For example, Perky (1910) found that her subjects confused a faint projection with their own images although there are some methodological problems with this study.

One attempt at a direct test as to the dependence of imagery upon access to 'visual perception processes' was carried out by Mamor and Zaback (1976). They investigated mental rotation by blind subjects using a tactile analogue of the Shepard and Metzler (1971) task. One inspiration for this work appears to be an interview study of dreaming by Jastrow (1888). Mamor and Zabach describe Jastrow's study as indicating that those blind after seven years old reported experiencing visual imagery in their dreams, whereas those blind before five years old made no such claim. The Mamor and Zabach task used tear-drop-shape wedges with a "bite" taken out of the left- or the right-hand side of the wedge. The left-hand wedge was always presented upright, the right-hand wedge was presented at a rotation of 0°, 30°, 60°, 120° or 150° in a clockwise sense. The subjects had to decide if the wedges presented were the same or different. Analysis of these decision times were interpreted by Mamor and Zabach as indicating rotation rates of 54°/sec for their early-blind (before five years old) subjects, 114°/sec for their late-blind subjects, and 233°/sec for the control group of blindfolded, sighted subjects. This study would seem to offer some support for the facilitation of imagery-based tasks, such as mental rotation, by visual processing. However, there is some suggestion in the paper that a verbal strategy may have been used by some subjects.

As a supporter of the view that imagery closely resembles visual perception, Finke has carried out a number of experiments that attempt to use the formation of an image as a potential help or hindrance to a subsequent perceptual task. His experiments tend to support the position that there are some mechanisms that are used both by imagery and by perceptual processes. Some of his studies indicate that imagery can influence perceptions. (This is perhaps not unreasonable if one's view of perception is close to the idea of the brain forming hypotheses about the world outside (one held by Richard Gregory or by the late David Marr and his co-workers).) Finke has a paper in the Psychological Bulletin (1985) which discusses these issues and the related theories: a more accessible source for the non-psychologist reader might be Finke (1986).

In direct contrast to the position that imagery resembles visual perception, Pylyshyn (1984) offers an account of the data from imagery experiments that does not involve analogue representations by drawing on four assumptions. Firstly, that the instructions in imagery experiments lead subjects to recreate as accurately as possible the perceptual events that would occur if they were observing the situation. Secondly, subjects draw on their tacit knowledge of both the environment and human perceptual processes in order to decide how to behave in the experiment. Thirdly, that subjects have the skills necessary to simulate the performance that would arise if they were using a spatial medium. Fourthly, that no special processes are recruited to perform 'as though one were using a spatial medium' when simulating such performance using a representation composed only of propositions. This account is not experimentally falsifiable since tacit knowledge "could obviously depend on *anything* the subject might tacitly know or believe concerning what usually happens in the corresponding perceptual situations" and that "the exact domain knowledge being appealed to can vary from case to case" (Pylyshyn, 1981: 34). Consequently, an assessment of this position depends on whether one can accept that subjects could both possess and apply the tacit knowledge capable of simulating visual perception to the extent experimentally observed.

The influential interpretation of the psychological evidence on imagery favoured by Kosslyn (and his co-workers) seems to be as follows. Imagery (to include the generation, inspection, and transformation of mental images together with their role in fact retrieval) is more than a simple mirror of the equivalent perceptual or physical processes. Because the underlying psychological processes are analogue in nature they are easily performed on an analogue representation and are not easily realized for a propositional representation. He considers it unlikely, given the limited degree of conscious control and access to such processes, that subjects are merely trying to behave as if they were carrying out the equivalent physical tasks. The level of theory specified (for example in Kosslyn and Schwartz, 1978) to emphazise the (cognitive) representation is considered both appropriate and better specified than alternatives such as those offered by Pylyshyn (1984) and Johnson-Laird (1983). He feels it inappropriate, at this stage, to specify the processes at the level of nerve-cell activity. He assumes that these analogue mental representations are actually processed when we have an experience of mental imagery and that processing differences are reflected in external behaviour. These analogue processes are better suited to certain types of data manipulation than propositional forms of representation: and it is in this context that Kosslyn suggests the usefulness of psychological work on imagery for workers in the area of artificial intelligence.

There are still differences of opinion as to the interpretation of many of the experimental results in the area of imagery. A number of authors (e.g. Johnson-Laird, 1983) still see the results as not discriminable from the predictions derived from a propositional model, whereas others (e.g. Kosslyn, 1980) interpret the results as convincing evidence for a unique, analogical processing system - visual imagery. (Discussion on these points can be found in Kosslyn *et al.*, 1979, and in Hilgard, 1981.) It should be remembered that most of the work on imagery has been in the area of 'visual imagery' and not concerned with equivalent processes related to other sensory modalities. One problem that still remains to be solved in the area of imagery, and one in which computer science and other disciplines may be of assistance, is the lack of a suitable formalism for representation.

6.3 Reasoning with Concepts

The use we associate with human action that gives it an advantage over those of machines is that of thinking, or reasoning. The process of reasoning depends on principles that establish some sort of relation between premises and conclusions. There have been several suggestions as to the identity of this set of principles.

Logic specifies the principles of valid reasoning (in certain domains). Most psychologists have assumed that there is a form of mental logic that enables us to reason. According to this *doctrine of mental logic* an inference is made by translating its premises into a mental language, drawing on the relevant general knowledge from long-term memory, and then applying formal rules of inference to these conclusions to derive an inference from them. The doctrine suggests that valid inferences are encountered by children, in the same way as well-formed sentences, and the formal rules of inference are derived in the same way as rules of syntax. The question that follows from this doctrine is: *what logic does the mind contain, and how is it represented there?*

Jean Piaget argued that the formal reasoning which children are supposed to master in their early teens is "nothing more than the propositional calculus itself" (Inhelder and Piaget, 1958: 305). (For an explanation of the propositional calculus and other terms from logic used in this section, see Chapter 2.) One of the major difficulties with this suggestion that the mental logic is the propositional calculus is that the rules of inference would hold true no matter what the content of the propositions. However, there is a body of psychological research to support the argument that the difficulty with which inferences are drawn by humans is dependent on the content of the propositions in the premises. Some of the most dramatic examples of this dependence occur in a task which has been used by a number of researchers. The task seems quite simple; the experimenter lays four cards in front of a subject displaying the following symbols:

E	K	4	7

The subject already knows that each card has a number on one side and a letter on the other. The experimenter then presents the following generalization:

If a card has a vowel on one side then it has an even number on the other side.

The subject's task is to select those cards which have to be turned over to determine if this statement is true or false. The order in which cards would be turned over is not considered, merely the question as to which would determine the truth of the generalization. The problem seems easy; try it before continuing.

In a study by Wason and Johnson-Laird (1972) nearly every subject chose to turn over the card showing a vowel; if it reveals an even number, the generalization is unaffected, if it reveals an odd number then the generalization is false. Similarly, most subjects appreciate that turning over the card with a consonant on would be irrelevant to the generalization. Some subjects chose to turn over the card showing an even number; if it shows a vowel, it would be consistent with the generalization, but would not either prove it true or false; if it shows an odd number it would not affect the generalization since nothing is stated about what should be on the other side of cards that have an even number. To select this card is an error of commission, since it is not relevant, but it does no harm. However, very few subjects select to turn over the card which presents an odd number. If this were to have a vowel on the other side it would disprove the generalization; therefore this error of omission is more serious. Selecting this card, as in selecting the card showing a vowel, might demonstrate a card bearing a vowel and an odd number which would refute the generalization.

If the same test is performed with cards showing a different content by using more realistic materials, there is a change in subjects' selections. With cards showing the modes of transport and place names:

Manchester	Train	Sheffield	Car

and the general rule:

Every time I go to Manchester I travel by car

over 60% of subjects chose to turn over the card with 'car' on it, whereas with the abstract materials only 12% did (Wason and Shapiro, 1971). The finding that performance on this task is different when abstract materials are used from when realistic materials are, has been consistently replicated by many researchers (see Evans, 1982, for a review of studies using this task). This task illustrates that the content of the premises to an inference has an effect on the difficulty of a deduction. This is inconsistent with the principles of the propositional calculus, which suggests that the semantics of the propositions is irrelevant to the inference. This is strong evidence that the propositional calculus is not the mental logic used by humans to perform inferences.

There are two further arguments against the propositional calculus being the mental logic. The first is that it requires the inference of all valid conclusions when it is obvious that only non-trivial conclusions are drawn by people. The second is that inferences that hinge on quantifiers such as 'all' and 'some' cannot be captured within this calculus. They require a quantificational calculus, which includes an additional apparatus for quantifiers.

This requirement suggests first order predicate calculus as a candidate for the mental logic. Johnson-Laird (1983) has pointed out five problems with this suggestion.

The first difficulty that such a suggestion must overcome is that some inferences are more difficult than others. This is easy to illustrate. In a study by Johnson-Laird and Steedman (1978), subjects could readily formulate a valid conclusion that follows from premises of the type:

Some of the children are scientists
All of the scientists are experimenters

whereas hardly any could formulate a valid conclusion from the following:

All of the bankers are athletes
None of the councillors are bankers

From the first problem there are two equally valid, if converse, conclusions: *Some of the children are experimenters; Some of the experimenters are children.* For the second problem the only valid conclusion is *Some of the athletes are not councillors.* For this problem, the converse (*Some of the councillors are not athletes*) is not valid, as would be the response that 'there is no valid conclusion of interest'.

Several theorists have suggested sets of rules of inference within a first order logic which attempt to capture differences in the difficulties of inferences (e.g. Rips, 1983). However, such attempts still suffer from the four other reasons which Johnson-Laird argues prevent first order logic from being the method of human reasoning.

Firstly, that although there are algorithms that will determine that an inference is valid for first order logic, there can be no such procedure to discover that an inference is invalid (Boolos and Jeffrey, 1980). This lack of a decision procedure prevents it from being used to produce the response often given to inferential problems that 'there is no valid deduction that can be made'.

Secondly, that as for the propositional calculus, the semantic content of premises should be irrelevant from processing, which it is not in the case of humans (as was demonstrated above).

Thirdly, that in first order predicate calculus all valid conclusions should be drawn whereas humans only draw non-trivial conclusions. This problem could be overcome by some form of relevance testing filter operating on the products of the logical inference, but this would be sufficiently complex to construct to leave the logic as a minor part of the inferencing system.

The fourth problem for first order predicate calculus is that, although it accounts for quantifiers which the propositional calculus does not, there are still quantifiers such as 'more than one' across which it cannot be used to draw inferences, while humans easily can.

This last problem could be overcome by a higher order calculus that can cope with more complex quantifiers. Although second order predicate calculus can operate over such quantifiers, there is no way to specify the formal inference rules by which the complete set of valid deductions can be derived with it. Because of this lack of specification, it would appear that appealing to higher and higher order calculi will not provide the mental logic required.

Looking elsewhere than logical calculi for a mental logic, there are several candidates. Euler circles have been suggested. However, although there is a complexity in expressing different problems in Euler circles, this complexity does not relate to the difficulty people have in making inferences. Another difficulty with them is that, like the propositional calculus, they cannot give rise to the answer 'no valid conclusion'. A second graphical alternative would be Venn diagrams, but these offer no way of predicting the errors that are made in human inference. There are also several candidates available in goal-directed programming languages. These allow rules of inference to be formulated with a specific content, with every general assertion taking the form of such a rule (such as the production rules discussed above). Although these overcome the problem of the effect of propositional content on inference, they go too far, and provide absolutely no machinery for general inferential abilities. There is a need for the sensitivity to content which they offer, in conjunction with the general inferential ability offered by the logical calculi.

One suggestion which attempts to combine this sensitivity and ability into a single approach is that of *mental models,* proposed by Johnson-Laird (1983). In contrast to the syntactic method of the formal rules of inference this method is semantic in nature. The general spirit of the suggestion is that the reasoner imagines a situation which would be described by a set of premises. Then, after drawing a conclusion from the situation, which would not be stated in the premises, an attempt is made to construct another situation from the premises in which this conclusion would be false. The reasoner can reach any deduction by applying a three-step procedure for this process (from Johnson-Laird and Bara, 1984: 5):

Step 1: construct a mental model of the premises, i.e. of the state of affairs described.

Step 2: formulate, if possible, an informative conclusion that is true in all models of the premises that have so far been constructed. An informative conclusion is one that, where possible, interrelates terms not explicitly related in the premises. If no such conclusion can be formulated, then there is no interesting conclusion from syllogistic premises.

Step 3: if the previous step yields a conclusion, try to construct an alternative model of the premises that renders it false. If there is such a model, abandon the conclusion and return to step 2. If there is no such model, then the conclusion is valid.

What complicates this procedure is that there are usually alternative situations which are compatible with the truth of the premises. Given a premise, such as:

All the scientists are experimenters

how is one to build a single model that captures its content? The answer is to draw on some simple assumptions which can be revised later. Therefore, one can imagine a set of scientists which will be consistent with the word ‘all’ but as small as possible; for example, three. The information that all the scientists are experimenters can now be added to the model. This would give the resulting model :

```
scientist =   experimenter
scientist =   experimenter
scientist =   experimenter
             (experimenter)
```

where the item in brackets represents an experimenter who is not a scientist. Although the premise and the reasoner's general knowledge do not require such an individual, they allow for the possibility of one.

If this were turned into a syllogism with the addition of another premise:

None of the children are scientists
All the scientists are experimenters

the deductive procedure can be applied to yield the conclusions. Firstly, the model would be expanded by the use of the first step of the procedure, to include this premise too, incorporating a barrier to represent set boundaries:

```
    child
    child
    child
---------------------------
scientist =   experimenter
scientist =   experimenter
scientist =   experimenter
             (experimenter)
```

Applying the second step of the procedure to this model suggests the conclusion: *None of the children are experimenters,* or its converse that *None of the experimenters are children.* Most subjects erroneously report these conclusions without continuing to apply the third step of the rule (Johnson-Laird and Bara, 1984) which results in a second model:

```
  child
  child
  child =    (experimenter)
---------------------------
scientist =   experimenter
scientist =   experimenter
scientist =   experimenter
             (experimenter)
```

which falsifies these conclusions. Applying the second step of the procedure again, the two models together yield the conclusions: *Some of the children are not experimenters* and *Some of the experimenters are not children* which by the application of the third step again, gives rise to a third model:

```
    child =    (experimenter)
    child =    (experimenter)
    child =    (experimenter)
----------------------------
scientist =     experimenter
scientist =     experimenter
scientist =     experimenter
               (experimenter)
```

which eliminates the first of the previous pair of conclusions, leaving the valid conclusion that: *Some of the experimenters are not children.*

The solution of this syllogism has required three mental models. The theory suggests that the greater the number of models required to draw a valid deduction, the harder the task will be. The results of experiments which show the comparative difficulty of reaching valid conclusions from syllogisms in both adults (Johnson-Laird and Bara, 1984) and children (Oakhill, Johnson-Laird and Bull, 1986) can be accounted for by a combination of the number of models called for by this theory and effects of the ordering of the items in the syllogisms (or more formally, the figure of the premises).

It should also be noted that this theory can also account for the types of quantification that the first order predicate calculus could not. For example, given the two premises:

more than half the experimenters are scientists
more than half the experimenters are children

a mental model can be constructed:

```
scientist =   experimenter
scientist =   experimenter    = child
              experimenter    = child
 scientist                      child
```

which yields the valid conclusion: *at least one scientist is a child.*

These examples of the use of mental models are limited to logical syllogisms involving quantifiers of various forms. However, models can also be used to represent and draw inferences across spatial relationships. This use is best illustrated by a particular problem. Given a problem where 12 individuals are seated equally spaced around a circular table, and a description of the relationships between them:

A is on B's right
B is on C's right
C is on D's right
...
K is on L's right

the transitive inference that A is on F's right is unacceptable, since A is close to being opposite F. The criterion for the validity of a conclusion to such a problem is purely semantic, depending on the impossibility of constructing a model of the premises and their context in which the conclusion is false. The application of syntactic inference rules as formalized in the quantificational calculi would not yield it. This example illustrates how spatial problems can be solved by the construction of a model, where other techniques would fail. It has been argued that similar examples of the usefulness of mental models for reasoning can be found not only for quantificational and spatial relationships but also for temporal and other continuous relationships.

6.3.1 The Representation of Mental Models

This evidence suggests that mental models are a realistic mechanism for human reasoning, but how is the information they draw on represented? Johnson-Laird suggests that "discourse can be represented either in a propositional form close to the linguistic structure of the discourse, or in a mental model that is closer to the representation of the state of affairs [...] than to a set of sentences" (Johnson-Laird, 1983: 160). The evidence to support this claim comes from a series of experiments which show that subjects tend to form mental models of a spatially determinate descriptions, while relying on propositional representations for indeterminate descriptions consistent with more than one spatial layout. In one study (Mani and Johnson-Laird, 1982), subjects heard a series of spatial descriptions, such as :

The spoon is to the left of the knife
The plate is to the right of the knife
The fork is in front of the spoon
The cup is in front of the knife.

After each description they were shown a diagram such as:

spoon	knife	plate
fork	cup	

and they had to decide if the diagram was consistent or inconsistent with the description. Half the descriptions presented to the subjects were spatially determinate (in that they described only one possible arrangement of the objects) and half were indeterminate (in that they described more than one possible arrangement). After the subjects had judged the descriptions and diagrams, they were given an unexpected test of their memory for the descriptions. On each trial, subjects had to rank four alternative descriptions in terms of their similarity to the original. These four were: the original description itself; an inferable description; and two descriptions with a different meaning as confusion items. The inferable description for the example contained the sentence:

The fork is to the left of the cup

in place of the sentence relating the spoon and knife. The inferable description is therefore not a paraphrase of the original, but it can be inferred from the layout of the original description. Mani and Johnson-Laird argue that this inference is only likely to be made if subjects construct mental models, and not if they maintain a propositional representation of the sentence. Further, the model they create would have to be symmetrical, since if they construct an asymmetrical model they will probably fail to consider the fork to be on the left of the cup. An asymmetrical model of the above example will illustrate this point:

spoon knife plate
fork

cup

The results show that subjects remember the gist of the determinate descriptions much better than that of the indeterminate ones, but they tend to remember the verbatim detail of the indeterminate descriptions better than that of the determinate descriptions. This cross-over effect requires the existence of at least two sorts of representation. Models do not encode the surface linguistic form of the sentences they represent, and when using them, subjects confuse inferable descriptions with the original. Propositional representations, however, do encode the surface form of the sentences. This result therefore suggests that subjects use a representation such as mental models to represent determinate descriptions and a propositional representation for indeterminate ones. One motivation that has been suggested for this change in representation is that, because models require a greater amount of processing for their construction than propositions, they are easier to remember, although they cannot express indeterminacy when it is noticed. It is possible that the introduction of propositional elements into mental models (such as the bracketing notation used in the models above) could also be used to represent alternative models of indeterminate descriptions, but it does not appear to be drawn upon in this study.

Further evidence for the construction of mental models is provided by a study of continuous and discontinuous descriptions by Ehrlich and Johnson-Laird (1982). In this study, subjects listened to three sentences describing the spatial relations between four common objects, e.g.:

The knife is in front of the spoon
The spoon is on the left of the glass
The glass is behind the dish

and then attempted to draw a diagram of the layout of the objects thus described. If subjects attempt to construct a mental model of the layout the task should be easier if a single model can be progressively constructed from the assertions (as in the above example), than if the description is discontinuous where the first two statements refer to no item in common, e.g.:

The glass is behind the dish
The knife is in front of the spoon
The spoon is on the left of the glass

In this case subjects may either construct a mental model for each of the first two assertions which must then be combined, or else represent the information in propositional form until some point after all the sentences have been heard. If the effect were one caused by the continuity of the sentences themselves rather than the construction of a single mental model, then a 'semi-continuous' description where the third sentence had no items in common with the second, but did with the first, should also prove difficult to recall, e.g.:

The spoon is on the left of the glass
The glass is behind the dish
The knife is in front of the spoon.

However, if a mental model is being constructed from a description such as this, it should prove no more difficult than the continuous example, since there is no requirement to construct two models because the third assertion refers to the spoon which has already been introduced in the first assertion. The results confirm this prediction of difficulty, since significantly more of the diagrams based on continuous descriptions were correct than those based on discontinuous descriptions, while the semi-continuous condition was not reliably different from the continuous.

These experiments support the view that mental models are used to represent various spatial knowledge, and may be used to represent the premises used to reason without the rules of logic, as described above. Although mental models have an analogical structure, they are not the same as images. The relationship between images and mental models can be seen as one where images can be described as views on the represented mental model. It is this dimensional nature of the representation of mental models that allows them to be manipulated in ways that can be controlled by dimensional variables and give rise to the performance described.

However, there are classes of representation which are currently problematic for mental models. One case is that of infinite regresses which can be exemplified by the representation of the mutual knowledge (Lewis, 1969; Schiffer, 1972) of two participants in a conversation or event. For example, if two people are standing in full view of one another amidst a downpour, then the fact that it is raining is mutual knowledge between them. In general, if the observers are X and Y, and the fact p, then it follows that X and Y mutually know p. This may be described in terms of the concept 'know', by the following series of statements:

X knows p
Y knows p
X knows Y knows p
Y knows X knows p
X knows Y knows X knows p
Y knows X knows Y knows p
.
.
.
etc. *ad infinitum*

There are two classes of solutions to the apparent paradox of this infinite regress. The first class requires the choice of a stage in the series at which to cut it off, either at a fixed point, or at a point which is selected as a function of the inferential or memory limits of the agents. The second class involves the use of a recursive notation in possible worlds or other advanced logics (e.g. Cohen, 1978). The same apparent paradox can exist in a mental model notation. Then, there will be an infinite number of models representing the situation, as there are an infinite number of statements in the series above. It would be possible to select a cut-off point in the sequence, but the decision as to where to place that point is no easier in a mental-models representation than for the series above. Although it is possible to represent recursiveness in diagrammatic or model form without using an infinite number of models (see Power, 1984), this representation requires the introduction of more special conventions and notations (like the horizontal bar and brackets in the models above) that make the models look more like expressions in higher order logic. This consequently reduces at least the face validity of mental models as representations to support reasoning which can be attributed to the simple models given above.

6.4 Conclusion

Representations and the processes which operate on them must both be specified for a system to be testable (Anderson, 1978). As a class, semantic networks and possibly the other major classes of representation (e.g. schema, frames) do not specify the processes which operate in them in sufficient detail to be testable. Individual models employing different representations are testable, and there has been found evidence which supports different aspects of many of them:

> People do appear to produce schematized representations of events, although there is little evidence for the existence of base schemata themselves. People do appear to produce successive records of events they experience, of which the most recent is retrieved (unless the description used for searching is specific enough to locate earlier records), as described by the headed records framework. People do seem to use a two-stage checking process based on the typicality of concepts to determine if statements such as 'a dog is an animal' are true as the semantic feature models suggest. People do find it easier to verbalize some knowledge (declarative) than other (procedural), as is accounted for by the use of production systems. People do retrieve information in regard of its connections to other information so as to give rise to the counter-intuitive 'fan effect', as predicted by some semantic networks (e.g. Anderson, 1976; 1983). People do solve some syllogisms with greater ease than others, as is consistent with the use of mental models.

However, there are few data that can be used to decide which of these phenomena arise because of the representation used to store information, rather than the processes which operate on them. There has been a debate as to whether information is stored procedurally or declaratively. The outcome for most theorists (e.g. Anderson, 1983) has been to produce models which contain procedural representations (production rules) for skilled procedures and declarative representations (semantic networks) for other factual information with processes to translate declarative information into procedures to model skill acquisition. There has been a debate about whether information is represented analogically (see Kosslyn, 1981) or whether it is always represented propositionally, but is processed to give rise to the effects of imagery and analogical reasoning (see Pylyshin, 1981). Again, most theorists (e.g. Johnson-Laird, 1983) have compromised and proposed the use of propositional representations to account for the surface effects found in recall and analogical representations (i.e. mental models) to account for analogical reasoning effects.

This chapter has summarized various phenomena and models composed of representations and processes which have attempted to account for them as parts of theories of human knowledge representation and reasoning. A model which claims to be a model of human processing must account for all of these phenomena. AI models which claim to offer the range of representation and reasoning exhibited by humans must also use these phenomena as tests. But what have these psychological studies to offer as criteria by which to assess and select representations for AI systems which are not intended to have any psychological validity? Simon (1978) has suggested that the informational and computational equivalence of representations should be the core criteria. *Informational equivalence* is achieved when the information inferable from one representation is also inferable from another. *Computational equivalence* is only achieved if two representations not only have informational equivalence but also the processes used to draw inferences for a task from one representation can be performed as 'easily' as for another. This introduces two important notions: firstly, that the computational equivalence of two representations can only be judged in the light of a particular task, or set of tasks; secondly, that there is some measure of 'ease' of drawing inferences which may be measured in time and effort to program, as well as time and resource usage at point of computation.

Models of human performance must be capable of accounting for the phenomena listed above, and use representations as close to those that humans do. They must also use the range of processing available to the human, to perform the large range of tasks that humans can. Most recent theories attempt to include such a range (e.g. Anderson, 1983; Johnson-Laird, 1983; Sowa, 1984). However, when producing computational AI programs, it should be possible to define both the tasks to be performed and criteria for 'ease of computation' which are independent of the tasks performed by humans. If the task is one involving the use of images, the computation may be 'easier' if both the processes operate on spatial units and the representation is more analogous to a picture than is a set of propositions. We know that humans have processes which can quickly draw inferences from pictorial representations since they quickly respond to changes in their visual environment (e.g. when driving). It is therefore reasonable to suggest that analogical representations may be more computationally efficient for the performance of some tasks by humans - especially when people draw external diagrams as aids to solve problems (see Larkin and Simon, 1987). However, for the computation to be more efficient using an analogical than a propositional representation for computational AI, there must be machine processes which operate on such representations. Without these, if the task is to solve propositional syllogisms, the 'easier' representation from which to compute may be one which is itself propositional despite the evidence supporting the applicability of the mental-models approach to human syllogistic reasoning.

In a similar way, for different tasks with different criteria for 'ease of computation', any of the representations discussed in this chapter may be the most suitable for computational AI (see Sloman, 1985). What psychological studies have to offer the writer of computational AI programs is an example of one (or more) ways in which the task can be achieved. The other chapters in this book describe specific tasks and the representations which appear most suited to them for the computational purposes of AI.

6.5 Further Reading

Although some aspects of cognitive psychology have been described in this chapter, many have been omitted. The interested reader may wish to consult a standard introductory text to cognitive psychology which covers more of these (e.g. Lindsay and Norman 1977; Anderson, 1985). The details of experiments which have led to the contemporary view of human memory are particularly well described in Baddeley (1976). There are also two more recent books which present individual authors' views of the cognitive system from a cognitive science perspective (Anderson, 1983; Johnson-Laird, 1983). Two issues have been deliberately omitted from this review since their inclusion would have considerably lengthened it: firstly, connectionist theories of representation which are currently a focus of much research (McClelland, Rumelhart and the PDP Research Group, 1986) and secondly, the psychology of language (see Garnham, 1985).

7 Conceptual Graphs

Michael Jackman and Cliff Pavelin

7.1 Introduction

The *Conceptual Graph* is a graph based notation for the representation of knowledge. It was developed by Sowa in his encyclopaedic work (Sowa, 1984) and subsequent papers (Sowa and Way, 1986; Sowa and Foo, 1987; Fargues *et al.*, 1986) and investigated by numerous workers in knowledge engineering applications (Garner and Tsui, 1985). As a representation scheme it draws on and integrates ideas from much previous work and, although there may be little new, the result is arguably a more flexible, more extensive and more precisely defined knowledge representation system than any of its predecessors. The notation gives the full representational power of first-order logic and the mapping onto logic is precisely defined. The notation can also cope with higher order and modal statements.

It is not enough simply to *represent*; the aim of a representation language is to be able to permit computer-based *reasoning* also. Sowa defines operations on conceptual graphs which are useful in reasoning. The most important is the "maximal join" which looks for the greatest match (appropriately defined) between two conceptual graphs; it is a substantial generalization of the unification operation under a suitable mapping. A methodology for performing first-order deductive reasoning on conceptual graphs is developed at length in (Sowa, 1984). There are also proposals about how conceptual graphs can be used in 'common-sense' reasoning although the ideas are far from fully developed.

Clancey (1985) writes: "every AI and cognitive science researcher should study the conceptual graph notation and understand its foundation in logic, database, and knowledge representation research." We are following that advice in this book and giving a summary of the essential facts of the conceptual graph formalism as an example of a contemporary knowledge representation scheme whose essentials can be grasped fairly easily.

The theory as originally described in (Sowa, 1984) begins from a psychological model of perception, but an understanding of this model is not at all necessary to appreciation of conceptual graphs and the scheme is described here purely in analytical terms. It is emphasized that this account can only be a selective summary and the interested reader should consult the substantial works referenced.

7.2 Types, Concepts and Relations

The primitives of the theory are *concept-types* (which comprise a *type-hierarchy*), *concepts* which are individuals (instantiations of concept-types) and *conceptual relations* which relate one concept to another.

7.2.1 Concept-types

Concept-types represent classes of entity, attribute, state and event. Examples may be: CAT, SIT, READ, PRICE, JUSTICE - they broadly correspond to nouns, verbs, adjectives etc. in language. It is assumed that in any conceptual graph (cg) system there is a pre-defined set of such types. A relation < is defined over the set to embody the notion that some concept-types are wholly subsumed in others. (Technically this is a *partial ordering* relation - like the set inclusion relation or the 'less than' in arithmetic, it is transitive and antisymmetric.) For example if PHYSICAL-OBJECT, ANIMAL, MAMMAL and CAT were concept-types, the relations

CAT < MAMMAL < ANIMAL < PHYSICAL-OBJECT

would exist. The meaning would be that every cat (i.e. instance of CAT) is also a mammal, every mammal is an animal etc. CAT is said to be a *subtype* of MAMMAL, mammal a *super-type* of CAT. The *type-hierarchy*, as it is called, need not be tree-like. For example, one might have a hierarchy in which:

ELEPHANT < MAMMAL
ELEPHANT < WILD-ANIMAL
RATTLESNAKE < WILD-ANIMAL
TIGER < MAMMAL
TIGER < WILD-ANIMAL

The hierarchy must form the mathematical structure known as a *lattice* - this implies that every two types must have at most one maximal common sub-type and one minimal common super-type. In the above example the fact that ELEPHANT and TIGER are sub-types both of MAMMAL and WILD-ANIMAL would necessitate the definition of a somewhat artificial WILD-MAMMAL as a separate concept in order to maintain the lattice. The lattice property is difficult to defend in cognitive terms but essential to some of Sowa's cg algorithms.

Although a type-hierarchy is taken as pre-existing in a system, there are also facilities for new type definitions to be given in conceptual graph form - for example, one may define a type ‘WILD-CAT’ to be a sub-type of CAT which has certain specified qualities, in this case those of being wild. Sowa calls this an ‘Aristotelian’ definition of a new type. But many concept-types will not have such a precise definition: Wittgenstein (1953) in a well known example, pointed out that the concept of ‘game’ has no precise definition; various types of game have family resemblances to each other. The concept-type GAME might well be a sub-type of ACTIVITY in a type-hierarchy but it would be impossible to give it a type definition which would specify the properties which define a game. The cg representation supports such problematic concepts as well as Aristotelian type definitions.

The type-hierarchy represents the subsumption relation between concept-types, sometimes represented by IS-A links in semantic networks. Sowa objects to mixing the higher order relationship represented by IS-A, a relation between *types* of individuals, with other relations such as ‘agent of’ etc. between *individuals* themselves.

7.2.2 Concept

A concept is an instantiation of a concept-type. In the cg notation it is written as a rectangle with the name of the concept-type inside. Thus Figure 1 represents an (unnamed) object of type CAT. A concept on its own like this forms the simplest type of conceptual graph. It has a meaning ‘there exists a cat’.

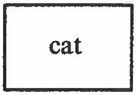


Figure 1

To refer to specific individuals a *referent field* is added. So Figure 2 represents the particular cat #123 (the referent is a name unique in the system). Interpreted as a conceptual graph this would mean 'the individual #123 is a cat'. Referent fields can also indicate much more complicated instantiations, e.g. a set of, one of a set of, etc.

For clarity we will denote referents here by names in quotes (e.g. cat:'fred') although names actually have a special treatment in the conceptual graph scheme.

7.2.3 Conceptual Relations

Conceptual relations show the roles that concepts play in relation to each other. Typical examples are as follows.

ATTR	BIG is an *attribute* of MAN
AGNT	MAN is an *agent* of DRINK
OBJ	WHISKY is an *object* of DRINKing
MANR	SLOW is a *manner* of DRINKing
LOC	An EVENT takes place in a *location*

In language, conceptual relations are indicated by word-order, case endings, prepositions etc. As with concepts there will be a pre-defined set of relation-types in any given system.

Normally a conceptual relation specifies the link between *two* concepts although there are some unary relations (see sections 7.3.3, 7.3.4) and Sowa defines ternary examples like BETW; this links three concepts which are physical objects, one of which lies between the other two.

Each relation will be constrained as to the concepts it can connect to. Thus an AGNT - which is a relation connecting the instigator or agent of an action to that action itself - will link to two concepts one of which is a subtype of ANIMATE and one which is a sub-type of ACT.

Figure 2

7.3 Conceptual Graph

7.3.1 Definition

A conceptual graph is a connected graph formed from concept and relation nodes. Each relation is linked (only) to its requisite number of concepts, each concept to one or more relations - apart from the special case of a graph consisting of a single concept.

Figures 3 and 4 give examples of simple cgs, with an English interpretation. Note that the arcs have a direction but its significance is minor - the most important use is to increase computational efficiency. There is an elaborate 'linear' notation to facilitate input and output of cgs on alphanumeric devices, but all examples here are given in graphical form.

7.3.2 Assertions

A basic cg, such as those given above, represents an *assertion* about individuals which exist in the domain being described. A precise mapping is defined from a cg into first-order logic; it gives a conjunction of predicates, one corresponding to each node of the graph. A concept C with no explicit referent will map onto the assertion that there exists an individual of type C:

$$\exists\ x\ C(x)$$

while if there is an explicit referent 'fred', it simply maps onto the assertion C('fred'). A binary conceptual relation R linking the concepts C(x) and D(y) would map onto R(x,y). Thus, for example, Figure 4 would correspond in logic to:

$$\exists\ x,y\ \ man(\text{'john'})\ \wedge\ agent(\text{'john'},y)\ \wedge\ look(y)$$
$$\wedge\ object(y,x)\ \wedge\ foot(x)\ \wedge\ partof(x,\text{'john'})$$

x and y respectively denote a foot and a 'looking event'.

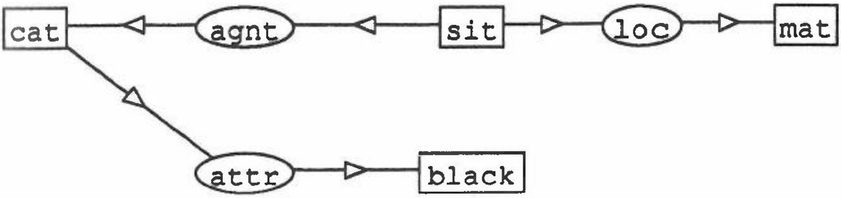


Figure 3 A black cat sits on a mat

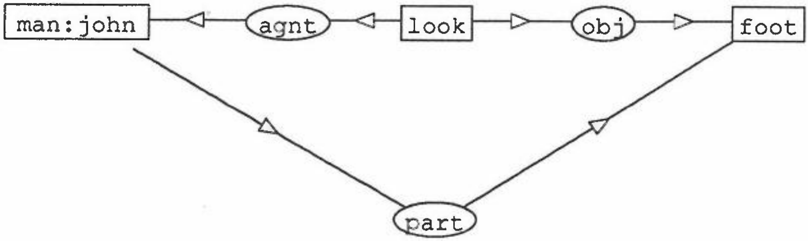


Figure 4 The man 'John' looks at his foot

It should be noted that the type hierarchy embodies obvious logical implications. If MAN is a sub-type of PERSON, then by definition the following is implicitly assumed:

$$\forall \text{ x man(x)} \rightarrow \text{person(x)}$$

(in particular the above 'john' is a person.) If the type hierarchy is regarded as complete, it is possible to make other inferences; for example if concept-types A and B have *no common sub-type*:

$$\forall \text{ x } A(x) \rightarrow \neg B(x)$$

Because these relatively complex logical assertions are implicit in the type hierarchy, the reasoning operations on conceptual graphs (see section 7.4 below) are likely to be more efficient than theorem proving methods based directly in logic (see Chapter 2, section 2.3.5 where a similar general point is made).

7.3.3 Negation and Quantification

To give assertions the full power of first-order logic (and allow extensions to higher order and modal logics) demands an extension of this notation above. Effectively a new concept-type 'PROPOSITION' is introduced. PROPOSITION can take one or a number of conceptual graphs *as a referent*. A concept of this type then asserts the conjunction of the graphs in the referent. In the graphical notation, a concept of type PROPOSITION is denoted by a box drawn round all the graphs in the referent as shown below.

The simplest use of PROPOSITION is the notation used to specify negation. A NOT operator (regarded as a unary conceptual relation) is applied to the proposition. The graph inside the box in Figure 5 asserts 'there exists a person with a mother'. Applying the NOT relation denies this proposition.

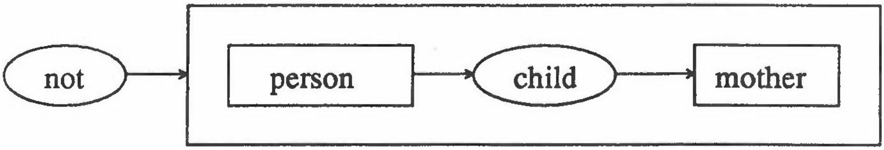


Figure 5

Propositions can be nested indefinitely in this way, and concepts at different levels in this nesting can be identified as referring to the same individual (they are joined by a dotted line known as a *coreferent link*). This enables full first-order logic to be represented. An example is given in Figure 6 - it corresponds to the denial of ‘there exists a person and this person does not have a mother’, i.e. the graph asserts ‘every person has a mother’. However, Sowa uses an extension of the referent notation to make such universally quantified statements more simply (see Figure 7). The graphical notation with boxes indicating ‘negative context’ etc. derives from one originally introduced by C.S.Peirce who devised an elegant and complete deduction system for first-order logic based on simple operations on graphs of this type.

7.3.4 Modalities and Tense

An advantage of the above notation is the ease by which it can be extended by a range of relations to indicate possibility, necessity, tense, knowledge, etc. These are equivalent to the *modalities* of case grammars (e.g. Simmons, 1973). An example is given in Figure 8.

7.3.5 Abstract and Definition

We have seen that a conceptual graph represents an *assertion* - generally about individuals in the world. There is another class of information to be represented - information about typical objects or classes of objects in the world. (There is a rough correspondence between this information and the Tbox of KRYPTON (Brachman *et al.*, 1983b) which is used for constructing structured definitions as distinct from the Abox, the assertion language - see Chapter 10.) The type hierarchy is one element of this information, but conceptual graphs themselves are also used to define new concepts in terms of old, give default information about a concept, give the constraints on concepts which relations can attach to, etc. It is assumed that, like the type hierarchy, these ‘definition’ graphs are pre-existing in any system - they are said to form a *canonical basis* for the domain.

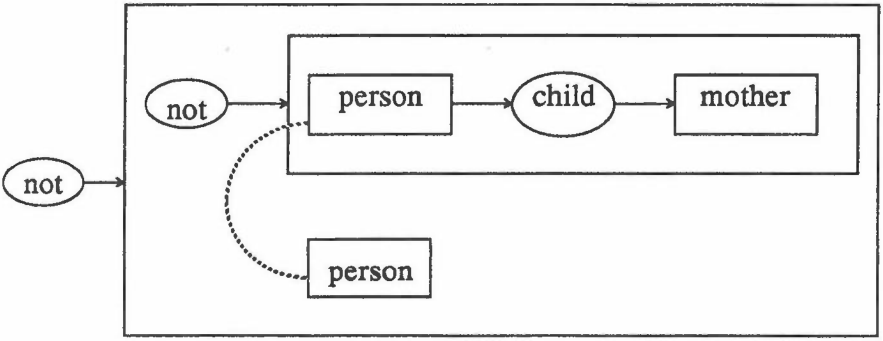


Figure 6

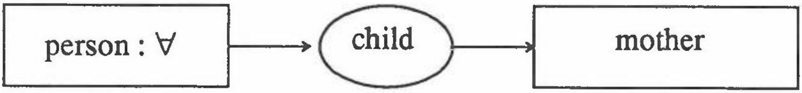


Figure 7

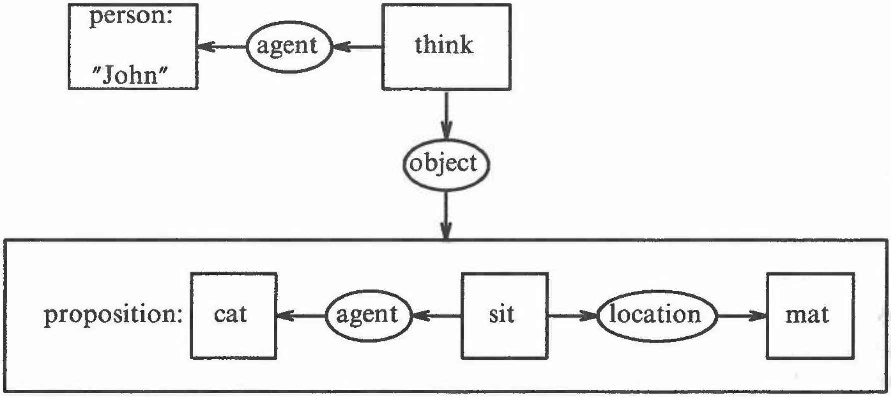


Figure 8 John thinks that a cat sits on a mat

A *Canonical Graph* is an example of one of these. It is a template for a concept or conceptual relation; it defines and puts constraints on the sort of links that can occur. For example a canonical graph associated with the concept-type TEACH may be as Figure 9. This says that the TEACH concept may be associated with AGNT (the agent), RCPT (the recipient, i.e. whoever is being taught) and OBJ (the object or subject matter); if so the

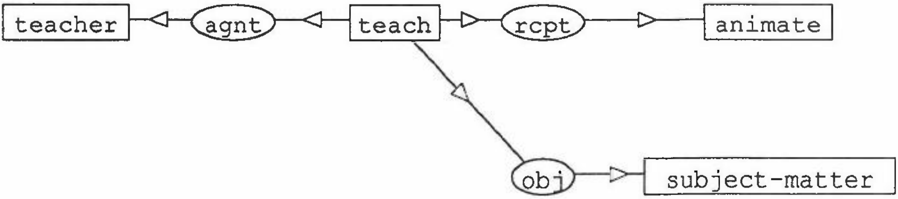


Figure 9

attached concepts must be the same as, or sub-types of, those given in the conceptual graph. The assumption is that all assertions must be derived from a starting set of canonical graphs according to certain rules (see section 7.4.1).

Canonical graphs are incorporated into a number of special types of definition known as 'abstractions' (Sowa gives a mapping from them onto lambda expressions in logic). Examples of two important types are given here.

A *Type Definition* defines a new concept-type in terms of an existing one with the additional properties which characterise it being expressed in conceptual graph form. This is the 'Aristotelian' definition of section 7.2.2. Thus a concept of a KISS may be defined as a sort of TOUCH done by a person with their LIPS in a TENDER manner; this would be expressed in cg form as shown in Figure 10. The 'generic' referent x-x is used to link the defining concept TOUCH with the new one KISS. If KISS appears in another cg, it can be 'expanded' by an operation described below.

A *Schematic Definition* of a concept-type is a canonical graph which gives *plausible* or *default* information about that concept. An example (from (Sowa, 1984)) is given in Figure 11. The set of all such schema for a given type is called a *schematic cluster*. A schema is supposed to act rather like a generalized frame giving typical properties and default values, but its use is far from precise in the current documents. A concept like GAME, impossible to define precisely (see section 7.2.2), would exist in the system as a schematic cluster giving a set of typical usages.

Various other categories of definition (e.g. prototypes, individuals) using conceptual graphs are proposed in (Sowa, 1984).

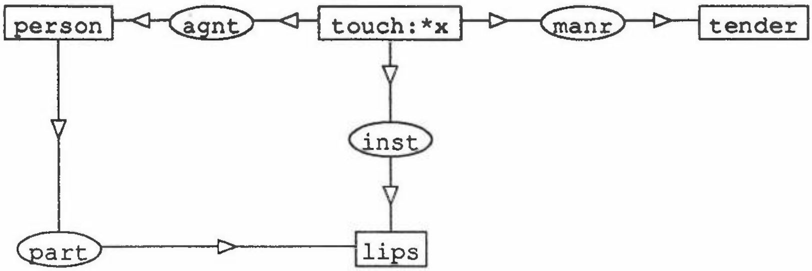


Figure 10

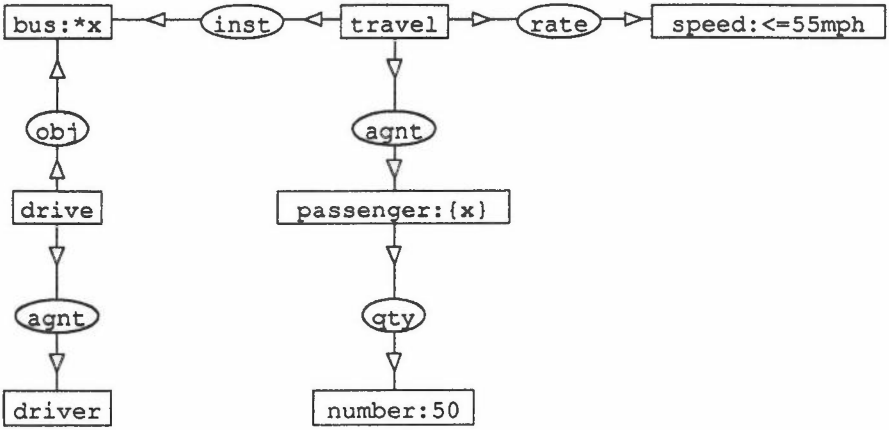


Figure 11

7.4 Fundamental Operations

A number of operations are defined on conceptual graphs. They are all *formation* rules by which one can derive allowable (not necessarily meaningful) conceptual graphs from a canonical basis.

7.4.1 Canonical Formation Rules

Canonical formation rules act as a generative grammar for allowable cgs from the canonical basis. Such graphs will not necessarily have any meaningful interpretation but they will at least obey certain selectional constraints. The important rules are as follows.

Restriction takes a graph and replaces any of its concept nodes either by changing the concept-type to a sub-type or adding a referent where there was none before. Thus ANIMAL may be restricted to CAT, which may be restricted again to CAT: 'fred'.

Note that the system assumes the existence of a predefined set of individuals whose conformance with concepts must be checked on such operations. For example if 'rover' were an individual DOG, the restriction of ANIMAL to CAT: 'rover' would be disallowed.

Joining takes two graphs with a common concept, and joins them over this concept, linking up the arcs from both graphs to form a single graph. Joining may also join a graph to itself, i.e. merge two concepts within the graph.

Simplifying removes any duplicate relations between two concepts - these can arise after a join.

Deduction

A graph that is canonically derived from others in this way is termed a *specialization* of any of the originals. In logical terms, existential variables may have been instantiated, predicates replaced by more constraining ones (i.e. sub-types) or additional constraints added by conjoining with further predicates. If graph gl is a specialization of g2, then g2 is a *generalization* of gl. It should be fairly obvious that a graph representing an assertion when translated in logic will *imply* any generalization of it (i.e. generalization preserves truth). If the girl Susan eats soup quickly, then certainly a girl eats soup.

It is possible to use these operations and properties of conceptual graphs to perform logical deduction using methods similar to resolution (see Fargues, 1986; Rao and Foo, 1987). In particular making two graphs identical by restriction is equivalent to unification in sorted logic.

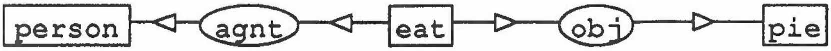


Figure 12(i)

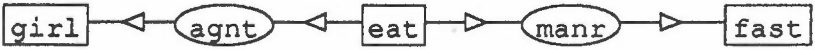


Figure 12(ii)

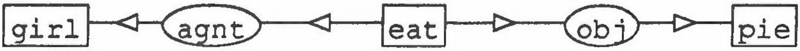


Figure 12(iii) Restriction of (i)

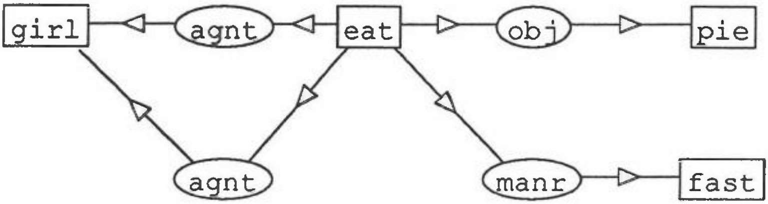


Figure 12(iv) Joining (ii) with (iii)

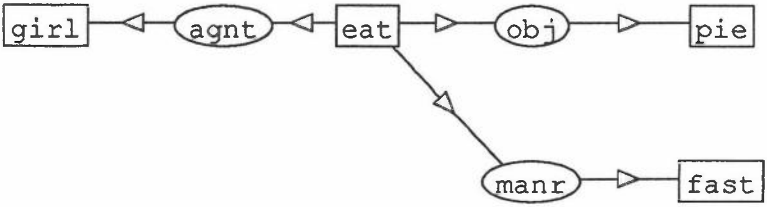


Figure 12(v) Simplification of (iv)

7.4.2 Maximal Join

The sequence of examples in Figures 12(i) to 12(v) form what is known as the *maximal join*: a join of two graphs followed by a sequence of restrictions, internal joins and simplifications so that as much matching and merging of the original graphs as possible is performed. The maximal join is used in a number of operations; the following is an example of the process of *type expansion* where an assertion containing a concept defined by a type definition is expanded by incorporating the type definition. Figure 13(i) is an assertion containing the concept KISS which is expanded using the definition given in Figure 10.

We equate a concept KISS with its super-type TOUCH in the type definition (Figure 10) and then, working out from here, match relation and join concepts restricting the common sub-types if possible. In this case MAN: 'john' joins with person after restriction to give Figure 13.

A *schematic join* is very similar but uses the maximal join to link in default information defined in a schema. This could be a first stage in a common-sense reasoning process.

If two graphs g1 and g2 have a maximal join G, then it is clear that g1 and g2 are generalizations of G, i.e. if g1 and g2 represent assertions, then each is implied by the assertion represented by G. G itself cannot be deduced from g1 and g2 but in certain circumstances it may be a plausible deduction based on matching concepts which are compatible, i.e are the same or have some common restriction.

Thus if a graph declares that Mary loves a man John and Mary loves a Scots person, the maximal join will result in the assertion that Mary loves a Scotsman John. The extent to which the default assumptions of typical common-sense reasoning can really be mapped into this form can only be guessed.

As has been said above, there are similarities between matching graphs by making appropriate restrictions and the process of unification familiar in automated theorem proving and Prolog. The maximal join can be regarded as a generalized unification operation (see Jackman, 1987, 1988).

7.5 Summary

The conceptual graph notation, many features of which could not even be touched upon in this account, is a flexible, consistent and precisely defined notation for the representation of knowledge. A number of operations are defined which are related to the sort of inferences that can be made by using the 'type hierarchy', the conformance of concepts with the names of individuals, and the standard principles of first-order logic. Potentially this may be useful for efficient deductive reasoning and (perhaps more important) methods of plausible reasoning in real-world problems.

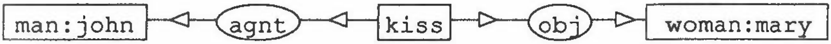


Figure 13(i)

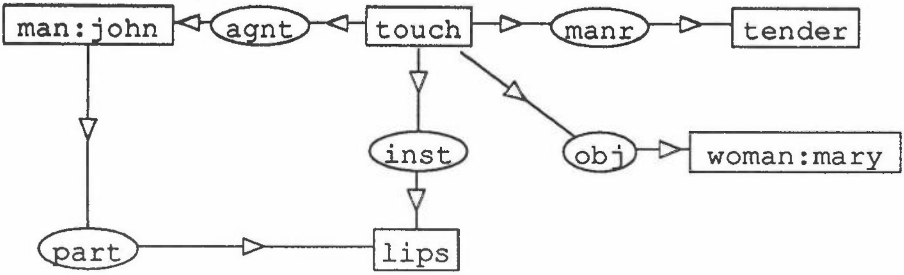


Figure 13(ii) Type expansion of KISS in (i) using type definitions in Figure 10

Clancey (1985) was perhaps going too far when he described the cg scheme as embodying 'the unification of logic, plausibility, and meaning constraints, setting a formal notation with four definitions, proofs, and algorithms for plausible reasoning'. The representation notation is very fully worked out but the reasoning processes require much research. However, there is sufficient world-wide interest in the cg formalism that the ideas are being developed and tested in real systems.

8 The Explicit Representation of Control Knowledge

Brian Bainbridge

8.1 Introduction

It has often been suggested (Bundy, 1983, Jackson, 1986) that a suitable strategy in knowledge-based systems research is to view some working program from a higher level of abstraction in order to see what has been learned from its implementation. A way to proceed is then to perform a 'rational reconstruction' to achieve its ends in a more principled way and to increase the performance of the original program.

The history of the MYCIN experiments of the Stanford Heuristic Programming Project provides good material to illustrate this approach. Chapter 5 gives an overview of the programming formalism employed. In this chapter, some features of expert systems control knowledge which MYCIN well exemplifies will be discussed.

Originally MYCIN formed the subject of E. Shortliffe's Ph.D. thesis (Buchanan and Shortliffe, 1984). The program was designed to aid a physician on the diagnosis and treatment of blood infections. It decides on the basis of clinical and laboratory tests:

(1) whether the infection is significant;

(2) what organisms are involved;

(3) what are the potentially useful drugs;

(4) what drug regime is best for the given patient.

These goals are expressed as antecedents in the top-level rule used by the backward-chaining MYCIN system, viz.

RULE 092

IF 1) There is an organism which requires therapy, and
2) Consideration has been given to the possible existence of additional organisms requiring therapy

THEN 1) Compile the list of possible therapies which, based upon sensitivity data, may be effective against the organisms requiring treatment, and
2) Determine the best therapy recommendations from the compiled list.

The main objects (contexts) to be reasoned about in the MYCIN domain are:

(1) the patient;

(2) cultures prepared in the laboratory from samples taken from the patient;

(3) organisms identified as present in these cultures;

(4) drugs suitable for dealing with these organisms;

(5) prior operations, and drugs associated with these operations.

These are organized into a data structure termed the context tree (see Figure 1). The reasoning process in MYCIN instantiates the context tree by exhaustive backward-chaining of an AND/OR tree, dynamically generated by the application of the inference engine to the rules whose top goals are expressed in rule 092 (above).

As well as obtaining a list of recommended therapies, the user can ask general questions about the knowledge base, for example 'What rules mention meningitis?'. This facility is not a full 'natural language' front-end - it is implemented by key-word scanning. It is also possible for the user to ask 'how' and 'why' at points in the consultation when the system is requesting information. 'Why' is interpreted as 'why is it important that you have this information?', which inevitably results in the printing-out of the rule currently being considered. Another 'why' will produce information about rules referencing the current rule, and so on - up to the top-level goal. The goal-tree is ascended. Similarly, a 'how' question asked in response to the system's statement of some conclusion produces a trace of how the

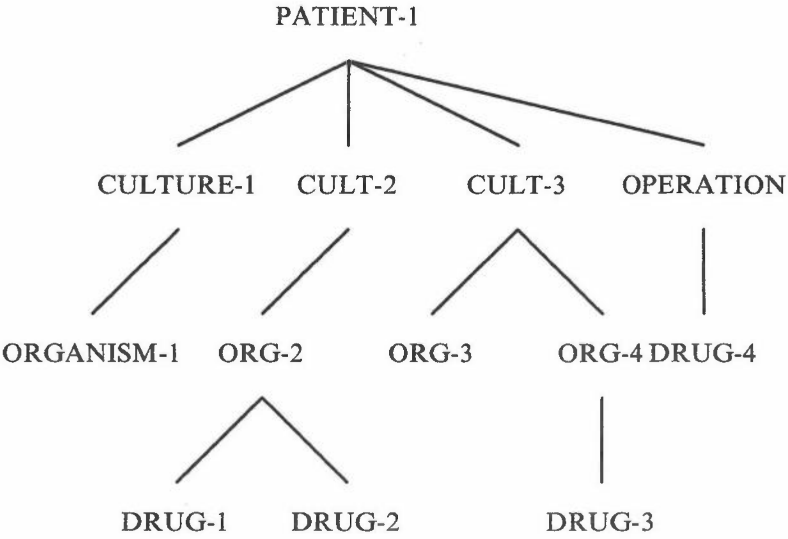


Figure 1 Domain of Mycin (Context Tree for Sample Patient)

information was inferred - it involves a descent of the goal tree.

The simple backward-chaining used in MYCIN was found to give rise to a reasoning process at expert level. The chaining mechanism focuses requests for data, is simple to implement and is also easy to explain to the expert involved in the knowledge elicitation process. It also has the advantage that the description of this line of reasoning can form the basis of an explanation subsystem.

The context tree, in the first place, literally provides rule context, in that it enables the system to relate one object to another. For example, in Figure 1, the tree indicates that organism-4 came from culture-3 and not from culture-1 or culture-2. However, the need to build up the context tree also provides constraints which focus the dialogue with the user, and gives an extra degree of focus to that provided by the depth-first search of the AND/OR goal tree. A fairly natural dialogue results ((Buchanan and Shortliffe, 1984) has examples).

Figure 2 is an attempt to generalize MYCIN's control structure, and could also be used to describe a variety of expert systems, e.g. a blackboard system as in (Aeillo, 1983).

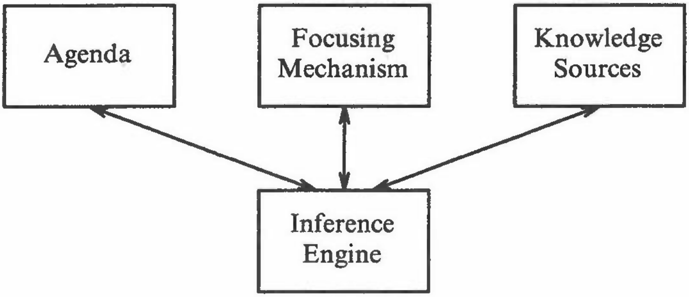


Figure 2

Control is mediated through the agenda - a list of tasks to be done. In the case of MYCIN, this is a first-in last-out queue. To illustrate its operation, we can consider what happens when the value of a parameter has to be inferred. To do this, a list of rules which can be used to deduce the value is retrieved, and the planned execution of these rules is posted as a new task on the agenda.

The way in which the agenda is manipulated is 'wired-in' to the inference engine as a procedure. Similarly, the way in which the context tree is instantiated is represented procedurally. There is no way in which the action of the inference engine can be changed (unlike other systems such as OPS5 (Brownston *et al.*, 1985)). The strategy for rule use is, as noted above, 'wired-in' and is not available for examination, changing or reasoning by the consultation, explanation and knowledge elicitation subsystems. As mentioned in Chapter 5, knowledge about knowledge - metaknowledge - is needed for higher performance expert systems.

8.2 Metaknowledge

To illustrate this point, let us consider the problem of knowledge acquisition in MYCIN. In the original system, the knowledge acquisition system was little more than an editor which could be used to edit the Lisp data structures representing the rules, lists and tables of the knowledge base. R. Davis in his Ph.D. thesis expanded this system and made it knowledge-based (Buchanan and Shortliffe, 1984). The system uses some of the methods used by the knowledge engineer when modifying the knowledge base. When the knowledge engineer peruses the knowledge, perhaps with a view to adding a new rule, he already knows a lot about the form and contents of the rules and facts. He is able to criticize any suggested new rule and to ensure that it can be incorporated into the knowledge base with no unforeseen side-effects.

TEIRESIAS uses such knowledge about knowledge - metaknowledge. It could be said to 'know what it knows'. This enables the program to make multiple uses of its knowledge. The domain knowledge is not only used directly by the system, but can be examined and generalized about (abstracted), and the system can direct precisely how it is used.

Because the knowledge acquisition system has to be able to add new data structures to the knowledge base, it needs knowledge about syntax. Davis' approach involves a data structure schema which provides a framework in which representations can be specified. Taking rules as an example, the antecedent and consequent clauses of the internal representation of a rule are coded as Lisp functions. Function templates are provided which indicate the order and generic types of the arguments in a typical call of that function. For example, the function SAME has as template

(object attribute value).

An instance of its use might be

(SAME CNTXT INFECT PRIMARY-BACTEREMIA).

The system is thus able to examine its own data structures.

Besides representation-specific knowledge about data structure syntax, i.e. about encoding, TEIRESIAS also has knowledge about the contents of rules. This knowledge is specific to the domain of application. Examples would be information about the possible uses of a piece of knowledge (e.g. information about the seriousness of an illness) and its requirements for time and space. Thus information about patterns and trends in object-level knowledge can be represented and used. This metaknowledge is held as rule models - abstract descriptions of rulesets built from empirical generalizations about the rules. The system examines the ruleset and builds up clusters of knowledge about rule patterns. The central idea is the characterization of a typical member of the ruleset (a prototype). This idea is, of course, used in other systems, such as the CENTAUR system of (Aikins, 1980), and usually involves some sort of frame representation.

A use of such a rule model could be when a new rule to categorize an organism is being formulated by the knowledge engineer. If the engineer suggests a rule with no clause concerning the morphology (shape) of organisms, the system can offer some useful criticism, since the rule models suggest that most rules of this type would include such a clause. TEIRESIAS actually offers to write such a clause, and will even include a plausible value for the type of morphology, e.g.

'The morphology is rod'

since it also knows that rod is a typical value of morphology.

Besides these metadata (function templates and rule models), TEIRESIAS also has metarules which guide the use of knowledge and decide what rules and methods are to be applied. At the implementation level, their effect is to modify the list of relevant rules retrieved when an attempt is made to evaluate an antecedent of a rule.

There are two types of metarules. First, a pruning metarule can fire. The effect of this is to exclude particular rules from consideration. In terms of the goal tree, this amounts to a decision not to explore a given branch. It amounts to a judgement on the overall utility of a rule, as to whether it is any use at all in a specific context. The other type of metarule used by Davis encodes knowledge on the relative importance of object-level rules. At the implementation level the metarule acts to reorder object-level rules relevant to some goal before invoking them.

An example:

METARULE 004

IF 1) There are rules which are relevant to positive cultures
AND 2) There are rules which are relevant to negative cultures

THEN It is definite that the former should be done before the latter.

This amounts to a less drastic decision about restructuring the goal tree. The branches are reordered rather than pruned.

At any node expansion, Davis' system chooses complete expansion (exhaustive search), reordering of goals or pruning of goals, and thus allows several types of metaknowledge. Together with the prototypical knowledge of rule models and function templates, it gives rise to a pattern of search which is not 'blind' but which is guided by heuristic knowledge. Only one level of metaknowledge has been described, but the scheme could be extended to indefinite metalevels. The inference engine used is still simple and a number of extensions are possible. For example, metalevel rules could select different types of inference mechanism (such as forward- or backward-chaining) at appropriate points in the search process.

In the past the system performance has been enhanced by adding large quantities of domain-specific knowledge. However, there seems no reason to believe that performance is linearly related to the number of rules. Indeed, it might well be that performance will 'flatten out' - although there are more rules, there is also a harder problem of search. Building the high performance systems of the future could need strategies for acquiring metalevel knowledge which would guide the use of other, lower-level, knowledge. TEIRESIAS is an important implementation which gives guidelines as to how such additional knowledge and mechanisms can increase functionality.. It also demonstrates how knowledge can be reused - the MYCIN medical knowledge base is used as a knowledge source from which function templates, for example, can be abstracted.

8.3 Classification of Metaknowledge

Davis' work indicates that metaknowledge is not uniform. It can, for example, be held as rules or data. It can be applied at various levels. It can be used by different subsystems. Clancey (1983, 1986) has suggested an interesting classification of metaknowledge. He has certainly raised the level of abstraction in this research field by providing a useful metaknowledge taxonomy and suggesting how we can elicit and use such knowledge.

Clancey points out that MYCIN-style systems are often described in terms of the language of graph search - we use terms such as rules, goals and chaining. He argues that we need a vocabulary which is independent of the implementation language, whatever it may be. The description language should be at the knowledge level and should embody a more psychological and human-oriented approach.

This interest in metalevel description derives from Clancey's Ph.D. work, in which he developed a tutoring system called GUIDON by using the MYCIN knowledge base together with additional knowledge about teaching. It was hoped to develop a tutoring 'shell' which could be used to teach a variety of subjects by using different domain knowledge bases.

By using system-derived knowledge about rules in a similar way to that developed by Davis, Clancey's system abstracts patterns from the domain rules. These rule models are descriptions of typical groups of factors in the rules. By doing this, it becomes possible to annotate a rule with a reference to the corresponding rule model. At a slightly higher level, rule schemas were used to represent abstractions from the object-level rules of descriptions of different kinds of rules. For example, a rule schema description could designate a rule to a type that provided identification ('covers') for a specific disease (say meningitis) and could describe the context of its application (say clinical). Knowing what is typical for a given rule, the system can then determine what is untypical by 'subtracting' the rule antecedents common to all rules of this type, leaving a 'key factor' description. This key factor forms another annotation which is of use to the explanation and tutoring system. The knowledge engineer can provide other annotation, e.g. literature citations, which can prove useful as support knowledge for explanation.

Some knowledge is made explicit and therefore available by these methods. However, Clancey found many glaring examples of implicit knowledge.

Consider the following example :

RULE 123

IF 1) The age of the patient is greater than 17, and
2) The patient is an alcoholic

THEN Diplococcus might be causing the infection.

The medical knowledge contained in this rule is that the diplococcus organism is associated with alcoholism. What, then, is the function of the first clause? It is to guide the application of the rule in that it prevents the system asking a patient whose age is 17 or less about alcoholism. A 'hidden' rule is being applied, viz.

IF The age of the patient is 17 or less
THEN The patient is not an alcoholic.

The problem is that in the course of a tutoring dialogue, the student user might proffer the relation between alcoholism and diplococcus, but since this item of medical knowledge is not recorded explicitly, the system will not be able to record that the student has offered some possibly relevant and valuable evidence. It seems that as well as knowledge about rule use and form, we also need to unpack a whole range of knowledge which has been encoded into the knowledge representation language. Further examples abound - for instance, consider the effect of rule order. A purported advantage of production rule systems is that each rule is an independent 'chunk' of knowledge. Rule order is seen to be unimportant because the rules are essentially uncoupled. In fact, rule order can govern the order in which rules are applied, and this can cause the focus of the questioning to jump around, to be defocused. If we start tuning rule order to achieve a more satisfactorily focused dialogue, we are effectively embedding a strategy of rule selection and of knowledge use. Clancey terms this embedding *proceduralization* and points out how this makes knowledge unavailable to the system.

This proceduralization enters many areas of computing. Consider what happens when a programmer using a low level language, say assembly code, decides at some point in the program to initialize a variable to zero, and at some later point to increment that same variable and if the result is less than 100 to perform a branch to a prior address. The programmer's intent is to loop 100 times, although this is not explicit in the program.

A higher level language might allow the programmer to achieve the same effect by stating :

```
repeat 100
  .....
  .....
  .....
end-repeat.
```

Here the 'repeat' is explicit - the programming style has become slightly more declarative (and the execution possibly slightly less efficient). As in the other examples, the writing of an efficient procedure tends to hide the intent of the programmer. Clancey advocates unpacking or decompiling procedural knowledge. In particular, he wishes to represent explicitly domain-independent problem-solving knowledge in the medical and tutoring domains and thus to reveal the bases of medical diagnostic strategy. He has developed a framework that (as so often happens) seems to be useful not only in description of medical knowledge but also in the process of eliciting knowledge from the domain expert.

The divisions of his taxonomy are:

(1) Heuristic knowledge, e.g. associations between patient data and therapies or diagnoses.

(2) Strategic knowledge - control knowledge - how to apply rules.

(3) Structural knowledge - the taxonomy of domain objects, e.g. therapies, diagnoses, cultures, organisms.

(4) Support knowledge - knowledge used in justifying rules, ranging from 'deep' (causal) knowledge to reference citations.

In NEOMYCIN, a reimplementation of GUIDON developed with these principles in mind (Clancey, 1983, 1986), Clancey concentrates on structural and strategic knowledge to supplement the object-level knowledge. He points out that there can be problems with the use of support knowledge in explanation. If we have rules which encapsulate a causal model, for example, we need to use these rules in appropriate situations. For example, it might be useful to explain the action of a virus by following the causal chain associated with the viral infection. However, there are instances where this is an inappropriate way to explain a conclusion. For example, a system might decide not to prescribe tetracycline for a young patient because this drug can cause permanent blackening of developing teeth. If the user of the system queries this decision and asks for some justification, an explanation of the causal chain which leads from the administration of the drug to the blackened appearance of the teeth is probably *not* appropriate. It would be better to form an explanation based on the general medical principle that therapies which have socially undesirable side-effects should be avoided. If the system is able to reason about general principles, it will also be able to override them when necessary. For example, if no other drug were available or usable in the case described and the patient's life were at risk, it might be expedient to prescribe tetracycline, since saving life is a high order goal for all medical systems.

In NEOMYCIN, the strategic metarules encode general diagnostic strategy. The structural knowledge is organized as explicit representations of disease taxonomies, taxonomies which represent patient types (old, young, alcoholic) and taxonomies of goals relating to patients (save life, restore to normal physical state).

Clancey claims that his implementations decouple the perceived high-level inferencing procedure from the system's reasoning with domain knowledge and data. The original MYCIN program used, as stated above, exhaustive depth-first search. Clancey's later work seems to involve a more data-directed style. Clearly, the guidance provided by metalevel knowledge is sufficient to focus the consultation as well as providing a good basis for explanation facilities.

8.4 The CENTAUR Implementation

It is of interest to examine in some detail some work which illustrates in a particularly lucid way the abstraction of strategic and structural knowledge suggested by Clancey. Aikins (1980) has implemented a 'rational reconstruction' of PUFF, a system used to diagnose pulmonary (lung) disease which was originally implemented in the EMYCIN shell (i.e. the inference system remaining when the domain knowledge is removed from MYCIN). CENTAUR was discussed by Ringland in Chapter 4. Here we will examine it in more detail.

CENTAUR uses prototypical knowledge, viz. descriptions of typical lung diseases and a typical consultation, to guide the consultation and explanation process. Effectively, the prototypes choose which rules to use - they provide the broad context of action and the rules themselves provide the finer detail. Input data are matched to the prototypical data and this enables the system to classify the data and identify untypical patterns. The prototypes provide the focusing, facilitating clear control and grouping of data.

Control knowledge is represented by rules attached to prototype 'control' slots. This gives the knowledge context and separates it from other system knowledge. Aikins' general strategy is to represent explicitly the various types of knowledge possessed by the system. She criticizes the 'flat' rules of the original MYCIN. The problem is that their structural uniformity and seeming independence hide groupings which exist. Rules have different uses in different stages of a consultation, depending on the particular situation which has come about in response to input data. Domain rules to infer new information should not look similar to rules which control other rules and rules which set default values.

Frame systems (Minsky, 1975) have been put forward as an appropriate way to provide the required grouping, and are used by CENTAUR to organize rule-groups and hence to bring rules into play where appropriate.

The prototypes are organized into a hierarchy whose structure is related to the taxonomy of lung diseases (Figure 3). At the root of the tree is the consultation prototype. This has control slots whose values represent the various consultation stages. A consultation starts by the choosing of this prototype, which represents a primitive plan of the consultation. The user is allowed to set built-in options, for example the selection strategy for the current list of prototypes. A strategy which could be chosen is to pursue the disease prototype with the highest certainty measure (a scoring factor) first. Aikins comments that the explicit representation of the consultation stages means that they can be reasoned about by the system and, for example, could be re-ordered. (This was not actually done by the implemented system.)

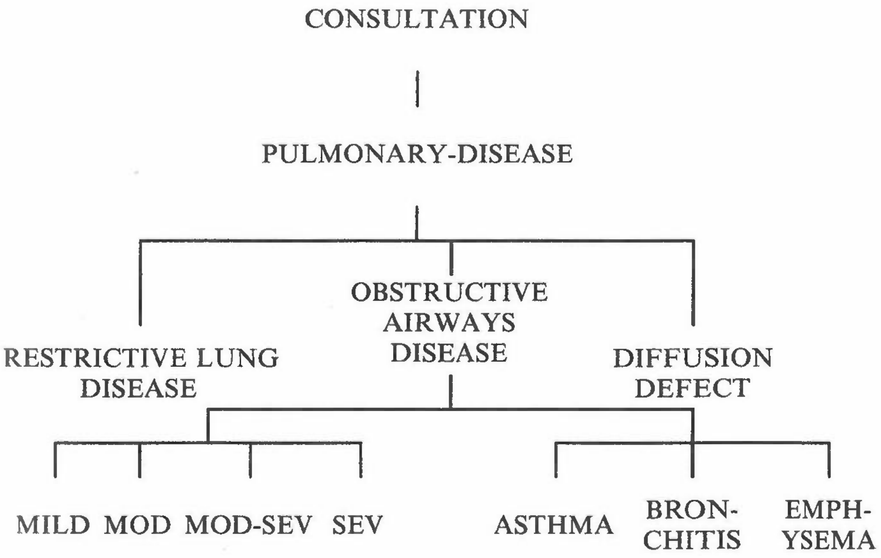


Figure 3 CENTAUR Prototype Tree

The next prototype chosen is the Pulmonary Disease prototype, which controls the acquisition of initial data. This data entry triggers other prototypes. Each triggered prototype is given a certainty measure, similar to the certainty factors used in MYCIN.

A summary of the prototypes which have been triggered is now printed out and the system chooses the ones to be followed up. In considering these active prototypes, the system takes into account all the data (including negative evidence) and so modifies the certainty measures. Since plausible values and possible error values are stored, other values are regarded as 'surprise' values and are printed out by the system for the information of the user.

After considering all the triggered prototypes, the system orders the hypothesis list and indicates the current best prototypes. The next stage is one of refinement, instigated by refinement rules stored with the relevant prototypes, and further questions are asked.

Next the 'summary' rules of the relevant prototypes are executed, which prints out a summary of the information that has been obtained in filling in the prototypes, i.e. the main inferences the system has made.

Lastly, the 'actions' slots of the confirmed prototypes are executed, which prints out the main data and conclusions in the same style as the reports produced by the human pulmonary disease experts.

Explanation facilities are provided by the Review prototype, which produces a review of the instantiated data associated with any given prototype. The system uses an agenda which is not only used as a mechanism for control by providing the system with a way to post tasks for execution, but also stores verbal descriptions of the tasks, their origins, and the reasons for placing them on the agenda. This information is thus available for explanation purposes.

CENTAUR is a research system specifically constructed to explore issues of representation and control. Compared with its precursor systems, it has access to a great deal more knowledge. The system knows what sort of stages occur in a consultation, knowledge nowhere explicitly represented in PUFF (Kunz *et al.*, 1978). It can respond to incomplete and inaccurate data in a reasonable way. Aikins has not explicitly set out to produce a psychological model of the expert (as does Clancey), but the system produced does proceed in the style of an expert in lung disease. The control mechanism is flexible and understandable to the user, moving between data-driven and model-driven phases. The use of the certainty factor mechanism gives the system the ability to 'change its mind' by, for instance, the arrival of new data causing less weight being given to a previously highly favoured hypothesis.

Certainly a great deal of the knowledge in PUFF has been 'unpacked' and made available, together with some new knowledge elicited from the expert. An indication of the greater knowledge is given by the number of rules, which have risen from 50 in PUFF to 300 in CENTAUR, and this does not take into account the wealth of prototypical knowledge.

8.5 Comments on CENTAUR and Conclusion

Jackson (1986), in his critique of Aikins' work, points out that a larger system built in the style of CENTAUR could have much larger numbers of triggered prototypes and that this could give rise to difficult scheduling and focusing problems, of the type encountered in the INTERNIST internal medicine system (Pople *et al.*, 1975). No attempt is made in CENTAUR to let the system itself derive metaknowledge, as was implemented in TEIRESIAS and GUIDON (Clancey, 1983). Clancey (1983) comments on knowledge of various sorts which was not made explicit by CENTAUR. For example, the control steps that specify on each level what to do next, e.g. "after confirming obstructive airways disease, determine the subtype of obstructive airways disease", are compiled into the prototype hierarchy.

However, CENTAUR can certainly be said to exhibit the explicit representation and use of a wide variety of metaknowledge and to be a significant pointer to what might be called the 'second' generation of expert systems. Whether the specific architecture is of much general use can be argued about. At least one system has been implemented with a similar architecture (Gale, 1985). Possibly it is unwise to take too much account of the specific architecture, but instead we should attempt to abstract the essential feature - the explicit representation and use of a wide variety of metaknowledge. Clancey (1983) comments that the metalevel analysis which the 'second generation' of expert systems will require will impose on the expert (and the knowledge engineer) the extra burden of becoming a knowledge taxonomist. This task will require considerable assistance, patience and tools.

9 Representing Time

Charlie Kwong

9.1 The Need for a Temporal Representation

We understand the world around us with relationships between the entities. When we consider something, an elephant for example, we know that it is an animal. Furthermore, we retrieve other identifying attributes to give a picture of a large four-legged creature with a greyish colour, big ears, etc. When told that John gave Mary a book, there is an automatic attachment of attributes to the entities within the discourse: gender to the named persons, perhaps that the book is made of paper leaves bound within covers.

To build AI systems, representations of these entities and their attributes need to be generated. These representations have to be easily manipulated, stored and retrieved. Other chapters in this book have looked at how it is possible to do this, and have discussed the issues involved. Here we examine temporal representation, the time aspect of relationships between universal entities.

First of all one might ask the difficult question - "What is TIME?" and try to think of an answer in the general everyday context. This is highly philosophical and I should not even try to touch upon the answer here. The reader is referred to an in-depth study of Time by van Benthem (1982).

How is time represented in general? We can take an analogue watch and say that it represents the passage of time by the rotation of its hands at a certain rate and it expresses the current time by the positions of its hands. A more appropriate question we should ask is "How do we humans represent time?". Humans can hold time representations in a variety of ways, by visualizing a pair of clock hands or the numbers of a digital time display to represent an instant or an angle for an interval of some minutes, for example. But our representations are more complex than that. We do not use angles to represent weeks or months. We reason about different time intervals and their relationships, possibly converting between different representations to achieve this. To express different time instances or intervals we use language embedded with ways to express temporal knowledge. In addition to pure temporal representations that we hold, it is more often the case that we associate temporal knowledge with other knowledge, i.e. that they are often intermingled.

The aim of AI is to build systems that will automate processes like problem solving, planning, natural language understanding/ translation and medical diagnosis, to name but a few. One important criterion is that these systems should employ intelligent means to tackle the problems given. A large number of experimental systems built to try out ideas have identified that modelling physical relations between entities is difficult, and have tended to concentrate on solving that problem using different physical models, ignoring the temporal aspects altogether.

In the domain of general problem solving many early implementations concentrated on physical relationships. The laws of 'physics' gave a seemingly better foundation for testing knowledge acquisition, storage and application theories. Certainly this was necessary so that the program controlling a robot did not request it to do physically impossible things. For example, trying to place a spherical object on top of a pyramid shape or to move some shapes from one side of a low wall to the other without lifting them over the wall. However, dealing with the temporal relationships is equally important in many areas of AI applications. Problem solving methods require sophisticated world models that can capture change over time within them.

Early attempts at building a medical diagnosis system examined ways of diagnosing a patient's symptoms to produce a set of possible diseases. Doctors do not stop there. They prescribe medicine to combat the diseases and apply a monitoring procedure to check the effects of the medication over some period. Time is sometimes quite an important factor in this process. If the course of the disease is not significantly stemmed by the medication the doctor may alter the medication to increase the dose or change it. A medical diagnosis system must be able to encapsulate time in a manner that allows it to reason about the progress of diseases and the effects of medication on them.

Temporal reasoning systems, like truth maintenance systems, are limited now by the models of the world that they manipulate. One commonly cited example problem for truth maintenance systems is keeping a fact that there is an ice-cube on a table. Excepting extraordinary conditions, the ice-cube melts at a given rate and hence the fact about the ice-cube being on the table has to be altered into facts about the diminishing size of the cube, its eventual disappearance and an expanding pool of water on the table. If the world model that the truth maintenance system resides in, is a world model without any representation of time the quality of the truth maintenance is doubtful. Temporal representations must be included for there to be an acceptable quality of truth maintenance.

Planning generally involves choosing from a selection of available resources and arranging them so that their use is maximized to solve a given problem. Sometimes a problem can be solved only with certain arrangements of some of the resources, and planning involves finding what that arrangement is. Examples of resources that come under consideration are space, time, money and even people. The earliest planning systems like STRIPS (Nilsson and Fikes, 1971) had no built-in notion of time whatsoever. Actions in the world model took an insignificant amount of time to execute; hence, changes in the world took place instantaneously. Where this has been applied to real robot arms that exhibit the strategy of those planning systems, the time intervals of the actions are modelled by the robot arm receiving a command, executing it and then signalling that it has completed the request. This is all very well when the object of the exercise is to test the planning strategy or the interface to the robot's arm. Later planners used temporal world models (see Allen, 1981; Allen and Koomen, 1983) in which there were explicit time intervals, and the planner tried to fit them together.

Natural language processing has to cope with a lot of tense and temporal information in its input. Linguistics as a subject of its own has produced much formal study into the temporal aspects of language. Dowty (1979) has identified a number of temporal phenomena. From the sentence "John was leaving on Thursday yesterday." there are:

Past/Future relations
Deictics - (now, yesterday,Thursday)
Vague event durations
Alternate worlds and times
Adverbial phrase interval bounds
Expectation/uncertainty

Any true generalized natural language understanding system must be capable of handling all these aspects. Some of them have been quite extensively exercised in "story understanding systems" and "question and answer" systems like "BORIS" (Lehnert *et al.*, 1983) and "CHRONOS" (Bruce, 1972).

Many representations in working systems are a side-effect of modelling the main application domain, although there has been work on producing systems that provide reasoning and representation on time alone. Some of these time reasoning systems were designed to test out methodologies, others were meant as part of other larger applications which needed a 'specialist' to handle temporal information. We shall see examples of these later when we examine the different approaches.

9.2 Characteristics of Temporal Modelling

What characterizes the representation of time? When we are examining a model that uses time what do we have to look for? Temporal information that is received can be absolute or relative to another referenced point in time. It can be an interval that is microseconds long or decades. Separate events can relate to each other in uncertain terms. What about persistence? Let us look at 'now' and what issues this brings up. We also do not want to be limited to retrieving data solely by their temporal references.

9.2.1 Temporal Determinism

Some information is timeless. The statement '3 is a prime number' is an example. It poses no relevant problem as there is no time information that can be meaningfully attached. Any general knowledge representation technique that incorporates a form of temporal representation should be able to handle timeless information with ease.

Statements can have two types of temporal information attached. The first of these is a date or time attached to the non-temporal part of the statement. Instances of these are:

> "The atom bomb was first used in warfare on 6th August 1945"
> "Karl Marx was born on May 5th 1818"
> "At 8:45am this morning, I was driving to work"

These statements all give an absolute reference of time that is either a calendar date or a time. In contrast to the explicitly stated times or dates, a relative reference can be used:

> "Yesterday was a rainy day"
> "The workers will go on strike tomorrow"

In addition to whether the time expression within any statement is relative or absolute, the whole statement can be temporally definite or indefinite.

The time that the statement is made is important in determining whether the statement is temporally indefinite. The truth value of a definite statement is unaffected by the time that the statement is made. The definite statement can be stated at any different time and still hold the same truth value. The following statements illustrate this:

"It is always sunny in Barbados"
"It rains every Sunday in Athens"

Both these statements are false because it does sometimes rain in Barbados and sometimes the sun does shine on Sunday in Athens.

The truth values of temporally indefinite statements are not independent of the time of their assertion. Take the statements

"It is now raining outside"
"It was raining outside yesterday"

The first will hold true if it is factually true that at the time of the assertion it is raining. If it rained on Monday but not on Tuesday then the second statement is true if asserted on Tuesday and false if asserted on Wednesday.

9.2.2 Granularity

Representations of time must be capable of encapsulating large timespans depending on the context. This can range from microseconds in the world of digital computer circuit design to millions of years in the subject of palaeontology. In everyday life the span probably ranges from decades to hundredths of seconds. Within narrower domains this span may be reduced even further. It may well be that not many systems need to be able to hold representations of time of such a wide range granularity. Perhaps it might be sufficient to have a representation that can be easily adjusted to cope with different kinds of granularities.

9.2.3 Points or Intervals

Do we represent time as a set of points or a set of intervals? Perhaps this is one of the most prominent conflicts in representing time. If we see a set of intervals then it is possible that something might be lost within the interval. If we see time as a set of points then we could have very large sets of points to represent long durations, each one representing every instant of the smallest granularity that we care to go down to.

9.2.4 Fuzziness

Often, fuzzy time information is given because it is unnecessary to expand in detail. It would not be useful to do more processing than necessary to try and remove the fuzziness. Words and phrases like ‘yesterday’, ‘tomorrow’, ‘three weeks ago’ often introduce fuzzy intervals.

When uncertain information is given, the representation must be able to handle this. If given two statements

> “Yesterday, I went swimming”
> “Yesterday, I went shopping”

The representation should be able to hold both of these without imposing any order on the events.

9.2.5 Persistence

Many situations in the real world require us humans to model persistence. I switch on a light in a room and then leave the room. I would normally continue to believe that the light was switched on until told or observe for myself otherwise. Several things could have happened. For instance, someone else switched it off or the bulb might have blown. Whether the light is in reality still on, or off, without any reason to believe otherwise I normally hold to the belief that it is as I left it.

The notion of persistence or truth maintenance is currently an area of intense research. Given a sequence of events at times as denoted by the times of their assertion t(n):

> t(0): I (now) pick up a loaded gun.
> t(1): I unload it.
> t(3): I point the gun at my head.
> t(4): I pull the trigger.

Any deductive system should give negative answers to the questions “Was there a noise?”, “Am I alive?” asked at time t(5). The temporal representation must be able to denote that, from that time t(1) onwards, the gun that I picked up is unloaded. It would be no good if, at a time later than t(1), the representation did not contain the information that the gun was still unloaded. The default deduction from the action of pulling a trigger might be that there is a loud noise and a bullet projected at what the gun points at. If this is so, then the deductive system would answer that there was a loud noise at t(5) from the given facts. This is a major problem in AI which has been coined the FRAME problem.

Persistence is usually easily gained in a modelling system. Any facts asserted to the system should remain there until explicitly contradicted or removed. This leads to the point of historical representationality. Consider the gun scenario. The immediate known history of the gun is that before t(1) it is loaded and not after. It is possible to have persistence in that at t(now) the gun is still unloaded, but there is no way of telling what time the gun was unloaded. What is needed is the representation of the history. When the assertion is made at t(1), simply replacing 'fact(the gun is loaded)' by 'fact(the gun is unloaded)' in our world state model does not maintain a history. Replacing it with 'fact(the gun is unloaded, t(1)) & fact(the gun is loaded, < t(1))' will represent the history more accurately.

9.2.6 What is Between the Past and Future

In the last paragraph we saw, briefly, a notation t(now) to indicate a particular instant. The dimensionality of time is such that there is a boundary between what has already happened and what is yet to come. This boundary is not simply a separation of the past and future. It often has a duration of its own right we call the present. Furthermore the present illustrates one of the qualities that makes modelling time difficult: it is a dynamic interval, constantly shifting in the direction of the future.

The present can take up different durations dependent on some context. Perhaps it is our human model of time that gives us the view of the present as it is. Time travel can be seen as simply the ability to shift the present that we exist in freely along the time dimension. But whatever it is, the present has a varied duration as shown by our language. Natural language allows different meanings for the word 'now' and we use it to express different things. In the following two sentences this is illustrated:

"I will clean the floor now."
"I am cleaning the floor now."

The first is normally used to express the intention of starting the action of cleaning the floor. That is, the interval of time which contains this action starts shortly AFTER the utterance of the sentence, whilst the second utterance usually occurs WITHIN this time. It may be used to refer to the instant or small interval in which an assertion using it is made:

"Now, I take the knife and slice through ......"

Also there is often an implied 'now' that is used in reference to events of the past or future. Take the scenario:

> Mary says to me, "I am setting off NOW, so I will see you in twenty minutes".
>
> Later, I meet John at the cinema entrance and say to him, "Mary said she was setting off THEN so she should be here any minute".

In my communication to John I am implying that Mary said something to the effect of "I am leaving NOW ...".

9.2.7 Co-existence of Time and Other Knowledge

So far, we have mostly concentrated on the characteristics that have to be taken into account when modelling time. Many temporal logics and temporal representation schemes have time as the central concept on which other knowledge is hinged. The temporal aspect of knowledge representation should lie orthogonal to the representation of other knowledge. It should not be restricted to knowledge manipulation using the temporal part of a fact only.

To illustrate what I mean here take the statement 'I was watching the 9 o'clock news on BBC1 last night'. The temporal part of the information conveyed here is the time (9pm) of the night before the sentence was uttered. The non-temporal information is the fact that I was watching the news on BBC1, if the only index used to store and retrieve this fact is the time I did it. The query of what channel I was watching cannot be answered directly without reference to the time index.

Very often the temporal factor in some knowledge is very insignificant or irrelevant compared to some other aspect. Some things are done just for the sake of doing them, others out of necessity. Often some other factor takes higher precedence in the order of things and then WHAT rather than HOW LONG is more important.

9.3 The Alternative Approaches

Having looked at some of the characteristics that exist in temporal representations and modelling, let us take a look at how these are exhibited in past and existing systems, whether explicitly or implied.

9.3.1 State Space Modelling

This method of space representation is an example where the temporal representation is implied. In state space modelling, a state is a snapshot of the current world state. Take an early planning system like STRIPS (Nilsson and Fikes, 1971) for example. It planned the sequence of actions necessary to achieve a requirement for a set of blocks to be in given positions. A typical statement giving a world state might be:

```
world-state(on(A,B),on(B,C),on(C,table),on(D,table))
```

Actions are functions which map between states:

```
move(block,dest)
        if clear-top(block)& clear-top(dest)
        then delete(on(block,X)) & add(on(block,dest))
```

Because each state is a snapshot, it is taken to represent the world at that instant in time (i.e. point-based representation).

In general we have a series of world states {(S1), (S2), (S3),...} to represent the passage of time as a sequence of instants. There are different temporal semantics that we can apply here.

One is that each state is an instant and there exists an interval between these instants during which the function which maps from one state to the other is active.

Alternatively, we can say that the states are the intervals during which the facts are known to be true. The functions which map between the worlds now are the instants between the intervals and it is during these instants that some of the world facts change from true to false, some disappear or others appear. What is wrong in this interpretation is that the functions take place instantly and that does not reflect the real world closely enough. In addition, there are events as McDermott (1982) pointed out which are not factual changes. He used the example of a person running around a track 3 times. How would a state space modelling representation cope with this?

Using either of the two interpretations above, there is still the problem that there is no distinction of different time interval durations. In the first one, all the operations take the same length of time to execute, whilst in the second there is one common denomination of time for the facts that exist in any world (the duration of each state, which is identical). Therefore in this weak representation we need to store explicitly, in the state, information about how long the function took or how long a state persists. This makes the temporal representation in state space modelling explicit.

In general, state space systems keep copies of old states as new ones are created. This requires a large amount of storage capacity if the worlds get large. Keeping copies of states, however, facilitates history and allows the answering of questions like

"What was the red block on before it was last moved?"

Because the mapping functions only need to change the relevant world objects, and copy the rest into the new state, this provides a very neat way of handling persistence. It demonstrates causality where "things only change when cause exists to change them".

These models, however, have a big disadvantage when it comes to handling tensed information. Consider the assertion "Sue said yesterday morning that she would come tomorrow". The state that represents the world 'yesterday morning' will have to be altered to fit in the new information. This would then have to be propagated through all the intervening states between then and now. It is arguable that changing the world model in this way is altering the belief model and is actually wrong. Because, if we take the view that a world state model like this represents a person's belief about the world around her/him, then doing modifications like these is akin to changing history. We are changing the fact that that person did not know what Sue said prior to the instant of the assertion of the statement above.

STRIPS was typical of the early AI systems which chose to ignore the temporal aspect of solving any given problem. They concentrated on the 'physical' side of things. Later however, some systems took a more direct approach and just concentrated on the temporal aspect of a problem solution. The system by Kahn and Gorry (1977) employed two different types of representation within the same system. Their system was designed to take temporal statements and then answer questions on the temporal part of the information given.

9.3.2 Date Based Method

Intuitively, every event/occurrence starts at a certain time. Using the date based method, these events are stored using this time as the index. Information with relative temporal content is resolved to a fixed time and this is used as the key. Cross referencing tables are needed here, one to hold the times and dates, another to hold the factual information. This or some other mechanism is needed to facilitate the accessibility from the temporal and non-temporal information.

However, there is a need to distinguish between the end time of an event and the start time of the event immediately following. If a switch is activated to turn on a light for example, the model should not allow there to be an instant when the light is both on and off. Figure 1 shows the analogy to the real number line in mathematics for representing ranges of reals.

There is the question of which time is used as the key. The choice is between the time of assertion or the temporal information in the sentence. If the first is used, any relative temporal information within the assertion can be resolved to an absolute time. However, some information may be changed in this transformation. Take, for example, the sentence "The bullion delivery will take place at dawn tomorrow" asserted at 3pm on December 31st 1988.

December 31	3pm	bullion delivery dawn tomorrow	(1)
December 31	3pm	bullion delivery dawn January 1st 1989	(2)

$$f(x) = 1: x < 3$$
$$f(x) = 2: x \geq 3$$

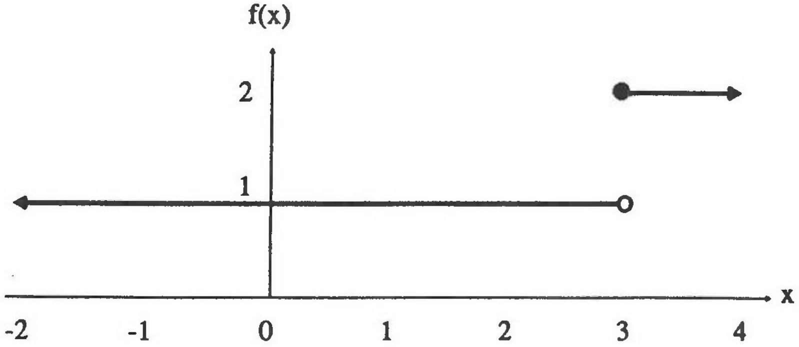


Figure 1 Real line representation of $f(x)$

In the second entry (2), it is not stored that the word tomorrow was actually used in the statement.

If we choose to use the implicit or explicit temporal information as the access/store key, then relative temporal references have to be resolved to a time or date. Again, this loses information (unless the time of assertion is also kept):

January 1   dawn   bullion delivery [3pm 31/12/88]   (3)

The big drawback with this method is that disjunctive temporal knowledge cannot be stored if we choose to index it on the contained temporal data. For example, the two assertions are made on the 1st of January:

> "Yesterday, there was a demonstration of sausage making"
> "Yesterday, there was a jumble sale"

The 'yesterday' will have to be resolved to the date 31st December. This will be placed in the events table.

December 31   sausage making demonstration, jumble sale   (4)

Care must be taken to ensure that the textual ordering or the order of the utterance of the sentences is not taken as an ordering of the events. This however can be interpreted more broadly in that there is an ordering based on the times of the assertions of the statements. However, can we really imply this sort of ordering based simply on the sequence of utterance of two statements? In addition, if we have the statements also asserted on the 1st of January:

> "Yesterday, there was a demonstration of sausage making
> followed by a jumble sale"
> "Yesterday, there was a demonstration of sausage making
> during the jumble sale"

We can have an entry in the events table exactly like that of (4) above for both sentences. Clearly the same representation for two distinctly different statements is not good enough. In the first case the two intervals do not overlap; the implication is that the end of the sausage making interval was the start of the jumble sale interval. In the second, the sausage making interval is contained within the jumble sale interval.

Here we see the illustration that shows that straightforward 'time-stamping' of events is not powerful enough to represent certain kinds of temporal information. Firstly, the entries for the events in the table are simply attached to the 'date' of the event. There is no indication of the duration of the events. The sausage making and sale could have taken 24 hours or 12 hours. They may have different durations. How would it be possible to use time-stamping to obtain default durations for jumble sales? How would that be fitted into the event entry of the table?

So far, some of the relative temporal information like 'dawn', 'yesterday' and 'tomorrow' has been converted to a date. Most often we use general terms to describe a time rather than specifically stating the exact time because there is no need to divulge additional information. Therefore if a system has to be more specific than necessary there is no harm. Sometimes, however, we are uncertain of the exact time of things we are referring to. So we intentionally use fuzzy time intervals or boundaries. This is especially true when talking about the future. In a purely time-stamping system fuzzy time will be extremely difficult to handle:

> "Three and a half weeks ago I was ill"
> "In four months' time I will be a millionaire"

It would be inaccurate to resolve the temporal information in these sentences to a date or time. If the times of assertion of the sentences are used as the indices for storage, there would be no need to resolve the fuzzy time.

9.3.3 Before/After Chains

Many events naturally fall into a sequence. By linking them together using pointers, we have a representation for the temporal relations of these events. The simplest method would just have bi-directional pointers between events (Figure 2):

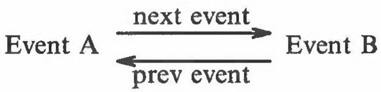


Figure 2 Events connected into a chain

This now forms the basis for building a chain of events (Figure 3a):

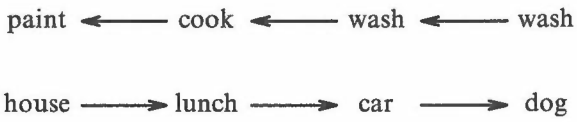


Figure 3a A simple chain

These chains do not explicitly represent the durations or any time units. These have to be incorporated into the nodes which hold the information about the event. If Xh means that associated activity took X hours, we thus have an interval based representation (Figure 3b):

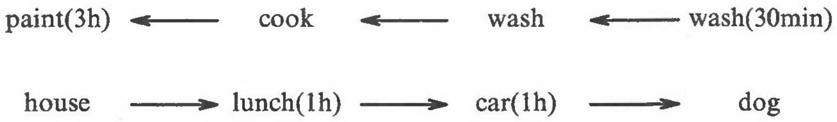


Figure 3b A chain with events containing their durations

If instead we have the start time of the activity associated, then we have a point-based representation. Simply by looking at the time of the activity after the current one, the duration of the activity can be computed (Figure 3c):

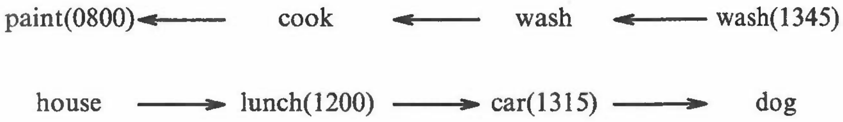


Figure 3c A chain with events containing their start times

This allows the mixing of events of varying duration alongside each other in a chain. Parallel events can be modelled by a splitting node with two pointers to its successor events. At the end of these two separate chains they merge again.

Chains, however, can get long, and to search for an event will require starting at the head of the chain and traversing all the way down the nodes until it is found. This can be overcome by abstractions of time intervals and/or event information. For example, at a higher level abstracted chain, there would be a chain of the main events in a working day and the sub-chains of these main events would carry more detail. There will have to be two different types of information abstraction, one temporal the other non-temporal. If we omit the latter, then to find out what time something happened will still require an exhaustive search of the chain. By abstracting the non-temporal information, it may be possible to search only the relevant sub-chains.

The time specialist was implemented and used to gain experience on the problems of having a time specialist integrated with specialist(s) of other domains for general problem solving and to see if this approach of trying to segment temporal information from non-temporal information is feasible.

The final conclusion of this work depended on the actual implementation of other specialist problem solvers which could interact with the time specialist and testing this out on some real problems. On its own, the time specialist produced interesting results when given stories of time travelling to understand.

More recently, the emphasis in temporal representations and reasoning systems has shifted towards the more formal and rigorous approach.

9.3.4 Formal Temporal Logics

These are often based on standard Predicate Logics. Formal Logics are very powerful and expressive for representations. In their book "Temporal Logic" (Rescher and Urquhart, 1971) Rescher and Urquhart give a clear and understandable introduction to a simple temporal logic based on a topological logic. They then go on and develop this further, including incorporation of more than two truth values.

In topological logic there is a positional realization operator, p. Applied to a proposition P(x), the predicate then reads that P(p)(x) is true at that place p. In Rescher and Urquhart's simple system R of Temporal Logic there is a temporal realization operator, t. This operator says that its associated predicate is true at that time and it is denoted as R(t)(A) where t is a time and A is a statement. For example, R(December 25)(it is Christmas day) says that, on December the 25th, it is Christmas day. To represent the important case of 'now' the symbol n is used. Therefore the statement

R(n)(the sun is shining) reads as "the sun is now shining".

There are axioms of R:

(T1) $R(t)(\neg A) = \neg R(t)(A)$
(T2) $R(t)(A \wedge B) = [R(t)(A) \wedge R(t)(B)]$
(T3) $R(n)(A) = A$
(T4) $R(t1)[(\forall t)A)] = (\forall t)[R(t1)(A)]$
(T5) $R(t1)[R(t)(A)] = R(t)(A)$
(T6) $R(t)(n = t1) = t = t1$
(T7) $R(t)(t1 = t2) = t1 = t2$
(T8) $(\forall t)A > A\,\hat{}\,t/n$

The rules of inference, in addition to *modus ponens*, are:

(R1) If |- A then |- ∀(t)R(t)(A)
(R2) If |- A = B then |-(..A..) = (..B..)

The proof theory of this system of logic is given in (Rescher and Urquhart, 1971).

What is of most interest to us here is the model theory of this temporal logic. The main idea of the development of the model theory is that the truth or falsity of the statements within the logic can be determined for any time.

As temporally definite statements have constant truth value over time they can be set aside. Whilst conventional Propositional Logic utilizes a Truth Table, Temporal Logic uses a Truth Cube. This simply has a third axis to represent a set of times. Then for each 'time-slice' of the cube, we have a table and this is treated in the conventional way. The exception is the treatment of "n". The interpretation of "n" is that it takes the value of the time at which it is being evaluated.

Since Rescher and Urquhart there have been many more approaches to Formal Logic time representation. Of the latest examples, Ladkin is in the process of implementing a system which uses an interval-based formalism. He has developed a formalism (Ladkin, 1986a, Ladkin, 1986b) based on Allen's work (1983). McDermott (1982a) presents a point-based one.

Alongside the development of formal logic as temporal representation, formal logic as temporal reasoning mechanisms is undergoing heavy research. So far there has been no mention of temporal reasoning, mainly because this book concentrates on the representation side of things. Some understanding of the complexity of reasoning with time is needed to see why the more formalized approaches are necessary.

McDermott (1982a) designed a temporal logic to take into account continuous change and the indeterminacy of the future. He shows how it is necessary to be able to model different futures. Just as in applying different functions that alter the physical world model, different events happening cause different possible futures. To be able to model branching futures is crucial in supporting reasoning about planning for the future. If there is only one future then any reasoning mechanism sees only one outcome of anything that it schedules. (If it were really intelligent, it would see the futility of its actions and give up altogether!!)

Continuous change modelling is supported to allow reasoning about things that change over time, for example, the level of water which rises during the action of filling the bath-tub. A more practical need to model a changing value is for the temperature of the bath water. If the water temperature is rising above a comfortable threshold before the required water level is reached then perhaps the cold tap should be turned on more or the hot one turned down. Furthermore, normally some adjustment of the taps is done and we decide upon a rough estimate of a time lapse after which we will check the water.

9.4 Concluding Discussion

As with most issues in computing, there is a question to be asked about the trade-off between expressibility and tractability. How tractable are temporal representation systems? The time specialist described by Kahn and Gorry (1977) is intended as a representation/reasoning system that handles temporal information. It is designed to operate alongside other 'specialists' to be a general reasoning system. How large will the final system be? Is it fast enough?

Sceptics of AI in the computing world are querying the necessity for all the complex techniques for modelling, planning and expert systems, to name but a few. Their arguments point out that many of the so-called AI systems are very large in terms of size, amount of processing power required and manpower to build, and, despite all this, seem to have very little practical impact. Why not use conventional programming languages, conventional computers?

Applying this argument to the more specific domain of time representation, why not just time-stamp something else that is representing the information that we want to attach a temporal reference to? Indeed, why not? As we saw earlier, time/ date-stamping is one of the methods of temporal representation but we also looked at it's drawbacks. In the naive simple way that time/ date-stamping works, it falls short in supporting deep reasoning about temporal relations.

Really, the question that has to be addressed is why do we want a good representation and how important is it to the application that requires some form of temporal reasoning capability. It may be that time-stamping is sufficient for the purposes so "why use a mainframe when a micro will do?". We cannot let ourselves be content that, when a simple temporal representation works, we stop looking for better ones.

10 Functional Approaches to Knowledge Representation

Simon Lambert

10.1 What is a Functional Approach to Knowledge Representation?

The word 'functional' has a number of usages in computer science. In system design, for example, a *functional specification* is a statement of how the components of a system will behave without reference to how they are to be implemented: their interfaces are defined but their internal operations are of no concern outside themselves. A similar idea is found in the development of programming languages, formalized as the *abstract data type* and described in for example (Liskov and Zilles, 1974), 'What we desire from abstraction is a mechanism which permits the expression of relevant details and the suppression of irrelevant details. In the case of programming, the use which may be made of an abstraction is relevant; the way in which the abstraction is implemented is irrelevant... An *abstract data type* defines a class of abstract objects which is completely characterized by the operations on those objects.'

The usual example of an abstract data type is the stack with its operations of Push and Pop. The practical development of abstract data types has perhaps reached its zenith in Ada, with its packages and ideal of reusable code (see for example (Freedman, 1982), 'A package is a collection of data types and allowable operations between objects defined from these types.'). We shall see, however, that in knowledge representation it is a whole knowledge base with associated operations that may be regarded as an abstract data type.

At this point it is worth mentioning that some work has been done on the application of functional *languages* to the needs of knowledge-based systems. Thus SYNTEL (Reboh and Risch, 1986) is a knowledge representation 'programming language' designed to suit the development of expert systems, and is functional in nature (by which the authors mean that there is no variable assignment as such but only function definition). Languages of this sort are not, however, the concern of this chapter. Our interest is in the knowledge base as abstract data type, and the ideas which have become associated with it. Work in the area has centred around the figures of Hector Levesque and Ronald Brachman. Levesque's remarks from (Levesque, 1984) set the scene:

> '... a knowledge base... interacts with a user or system only through a small set of operations... The complete functionality of a KB is measured in terms of these operations; the actual mechanisms and structures it uses to maintain an evolving model of the domain are its own concern and not accessible to the rest of the knowledge-based system.'

Some interesting results and powerful systems have appeared within the compass of this definition. But their roles are not as restricted as the succinctness of the definition might at first suggest, for there are other themes which have become bound up with the functional approach in various ways. They include procedural semantics - in this context, the definition of behaviour by means of programs attached to entities in a knowledge base - and hybrid representation schemes, the latter being an important feature of KRYPTON, the most highly developed of the systems to be considered. The following sections discuss these related topics in roughly the order they were developed; first, though, we consider why a functional approach to knowledge representation should be desirable.

10.2 Why a Functional Approach?

The definition given in the last section of what it means for a knowledge representation scheme to be functional made clear its links with software engineering principles. Here lies some of the motivation for investigating functional approaches. One should not be over-literal and claim that there is any connection with programming 'style', or that thirty years of high level language design are having a direct influence on knowledge representation, for the concerns of the two areas do not coincide. However, the principle behind the abstract data type, that the user should be prevented from interfering where he should not, from making assumptions about implementation and taking advantage of them, is very applicable to knowledge representation schemes, and in quite a precise way.

What is it that the user of a knowledge representation scheme is to be distanced from? The answer, of course, is the ubiquitous frame or semantic network, and for two reasons: firstly that the interpretation of the links in semantic nets (or slots in frames) is not well defined, so that different users are liable to put different interpretations on them, sometimes confusing levels of abstraction or imparting their own private interpretations; and secondly - though related - that frames/semantic nets are susceptible to treatment as mere data structures for manipulation. We turn to a paper by Woods for a starting point in the analysis of these issues.

One of the major contributions of Woods' famous paper 'What's in a link.....' (Woods, 1975) is the distinction drawn between structural and assertional links in semantic nets (for the purpose of this chapter, there is no real difference intended between semantic nets and frames - both are regarded as means of organizing knowledge for representation purposes). Structural links simply describe, whereas assertional links are intended to make a positive statement: Woods' example is of a representation of 'telephone' which has a link to (or slot containing) the colour 'black', of which he asks whether the resulting structure is a description of a black telephone or an assertion that telephones are black. The distinction is an important one, and Brachman takes it up in, for instance (Brachman *et al.*, 1983a), where he observes that Hayes assumed the assertional interpretation when reducing frames to first order logic (Hayes, 1979). (See Chapter 4 in this volume for a fuller discussion of Minsky's original idea of frames and Hayes' response to it.) Brachman goes on to point out that the assertional interpretation is a limited one for two reasons. First, instantiation (i.e. slot filling) is inadequate for representing incomplete knowledge. A slot must be filled with a definite value, and no progress can be made if the value is not known precisely, but is subject to some known constraints. Second, there is no distinction between essential and incidental properties. This point is of great importance because the drawing of such a distinction is a vital feature of the KRYPTON system described in section 10.5. For the moment let us say that an assertional interpretation sees no distinction between sentences such as 'Every triangle is a polygon with three sides' and 'Every car is red [in the restricted world represented]'.

Having pointed out the limitations of the assertional interpretation of frames, Brachman considers the purely structural alternative. In such systems (exemplified by KL-ONE (Brachman and Schmolze, 1985c)) the frames do not state facts but just create descriptions like 'a car with a steering wheel'. It comes as no surprise to find that the great drawback is precisely the inability to make assertions; or rather, because there is a need to do so but no mechanism available, that the structures are often misused and given assertional interpretations. As Brachman says, 'even if the structures of a frame system are taken non-assertionally, their presence or absence can still be misread assertionally and used to encode facts about the world.'

The fact is that frames, especially with the structural interpretation, are crying out to be manipulated as data structures. The standard notation, with its Lisp-like syntax, makes this all too apparent:

```
(CAR
        (IS-A VEHICLE)
        (OWNER person)
        (WHEELS 4)
        ...
)
```

The user (by which is meant whoever employs the frame package to interface to another part of the system or directly to the outside world) may well feel at liberty to treat the slots/links as he likes, pursuing IS-A chains (for instance) with happy disregard for what they really mean. This is another problem that has taxed Brachman: in (Brachman, 1983) he analyses some of the many interpretations that have been put on the overworked IS-A. An example will show just how far frames can go in becoming mere data structures for manipulation, and what effects this can have on the representation of *knowledge*. Consider Figure 1, a classification of students at a hypothetical university. Intuitively, the classes are all on the same 'level', and one feels happy answering the question 'How many kinds of student are there?' with 'Three'. But suppose we throw caution to the winds and attempt to represent students in more detail as in Figure 2. The result is incoherent confusion, and the question of how many kinds there are becomes

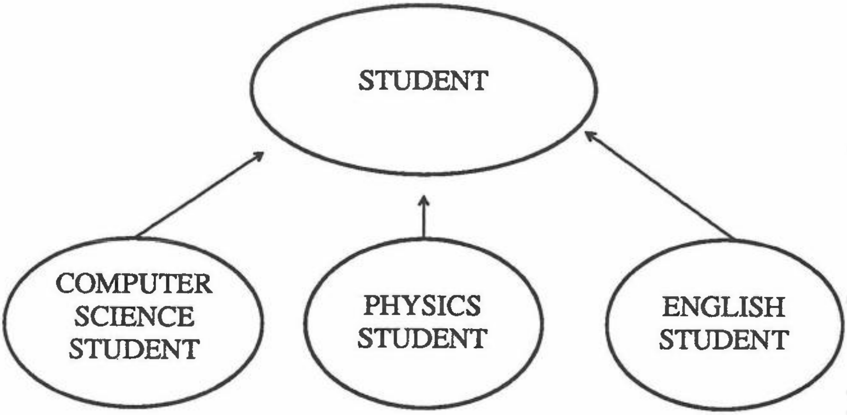


Figure 1 Kinds of Student

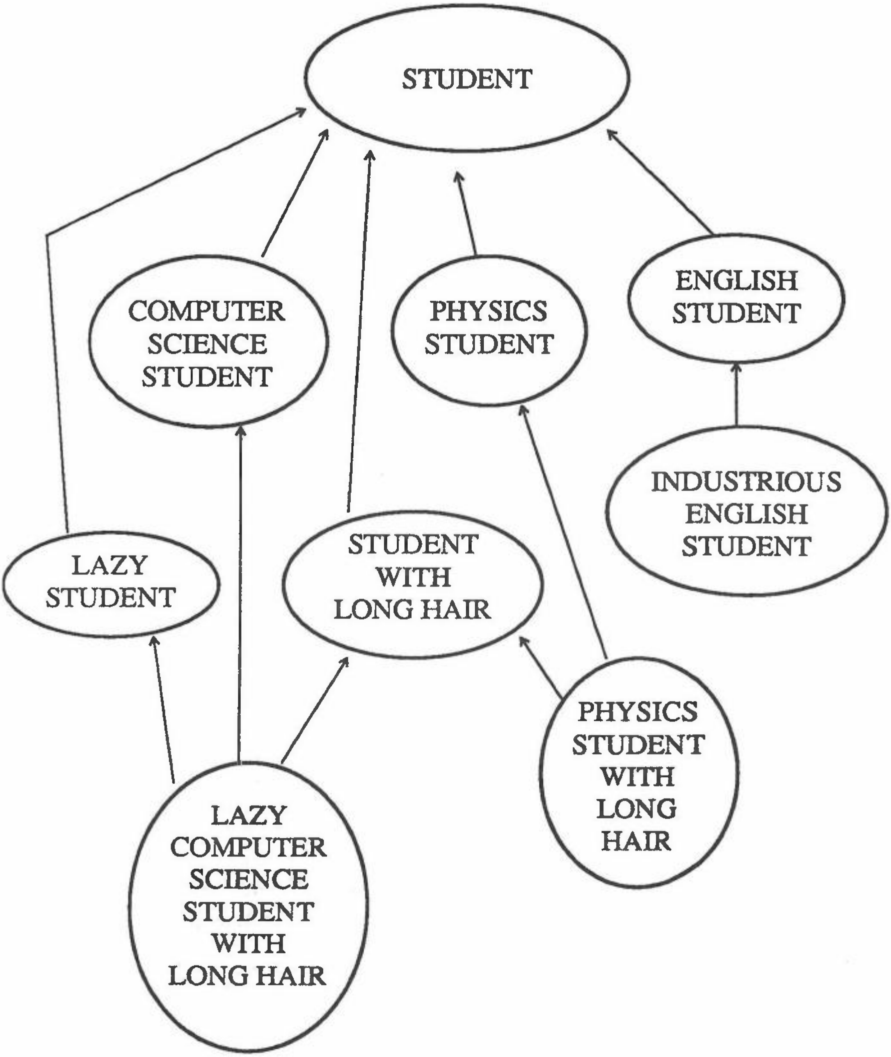

STUDENT
COMPUTER SCIENCE STUDENT
PHYSICS STUDENT
ENGLISH STUDENT
INDUSTRIOUS ENGLISH STUDENT
LAZY STUDENT
STUDENT WITH LONG HAIR
PHYSICS STUDENT WITH LONG HAIR
LAZY COMPUTER SCIENCE STUDENT WITH LONG HAIR

Figure 2 More Kinds of Student

meaningless. Yet one could make a case that all the links shown in the figures are IS-A links. All that seems to have been achieved is an arbitrary encoding without any logical structure. Clearly, many different arrangements of nodes and links could have been chosen, and any inference engine out to use such representations would be on very uncertain ground. In fact, Brachman gives a definite example of the dangers of uncontrolled use of networks in considering two definitions of 'bachelor': one sees a bachelor as being a person, with the qualifications of male-ness and being unmarried (two slot values), while the other regards him as a conjunction of a 'man' and an 'unmarried-person', themselves both kinds of person (Figure 3). As Brachman points out, the 'conceptual distance' from person to bachelor is different in these two representations, and 'spreading activation theories of processing in semantic nets might consider this distance to be significant.'

As well as his analysis of IS-A links, Brachman has also considered the nature of semantic nets in general (Brachman, 1979). He gives a comprehensive overview of the history and development of semantic nets, and provides an analysis of the levels of primitive employed, ranging from 'implementational' up to 'linguistic'. What all his work has done is to show how dangerous it is to allow direct access to the structures representing knowledge, for there are almost bound to be unstated assumptions made about the significance of slots or links, and the cause of knowledge representation in general will not have been advanced. Hence the need is seen for a functional approach.

The above discussion was based on Brachman's work, which led to the development of KRYPTON in an attempt to eliminate some of the problems described. Levesque's more formal work on functional approaches has a more formal justification, in terms of bringing out the full implications of a

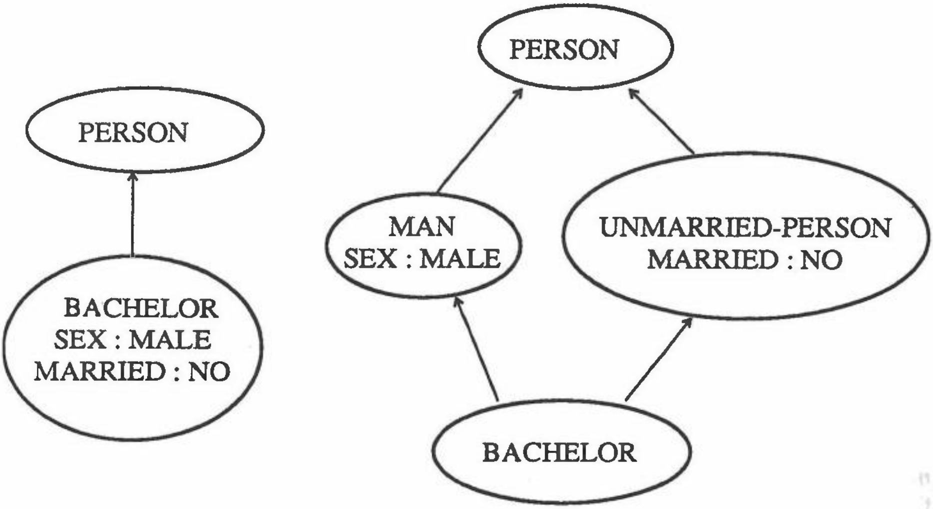


Figure 3 Two Definitions of a Bachelor

set of beliefs - what he calls 'competence'. Ultimately, though, his work needs little justification; like mathematics, its interest lies in the very fact of being able to prove results.

10.3 The Procedural Semantics of Mylopoulos and Levesque

It was mentioned in section 10.1 that one idea associated with functional approaches to knowledge representation is procedural semantics, or at least one manifestation of it, for the term has a wider applicability beyond the scope of this chapter. The time has now come to examine it, for the work of Levesque and others predates the work on functional approaches but possesses its important characteristics. Woods uses the term in his paper (Woods, 1975) introducing the structural/assertional distinction; though he says that his interpretation 'differs slightly from that which is intended by other people who have since used it', his definition, though very vague, conveys some of its later meaning:

> '...a specification of truth conditions can be made by means of a procedure or function which assigns truth values to propositions in particular possible worlds. Such procedures for determining truth or falsity are the basis for what I have called "procedural semantics".'

For something more concrete, we turn to the work of Levesque and Mylopoulos, and in particular their paper (Levesque and Mylopoulos, 1979). This paper has a number of important themes, but essentially springs from an attempt to formalize the semantics of semantic networks by using programs to define behaviour. In the introduction the authors express this motivation, take a swipe at the usefulness of logic as a representation scheme ('there is no distinction between an inference rule that *can* be used and one that *should* be used'), and make two significant statements from our point of view:

> 'To interpret this diagram [of a semantic net] as a model of a data structure within a computer memory simply postpones the problem [of what it really means] since we must now ask what the data structure represents.'

> 'These diagrams [i.e. those used by Mylopoulos and Levesque to illustrate their scheme] should be understood by the reader as convenient visual aids, not to be confused with the representation itself (defined by the operators of a formalism) or a possible implementation of this representation (defined by an interpreter of the formalism).'

The first of these points is precisely the justification for a functional approach that we have seen in the previous section; the second states the essence of that approach.

Mylopoulos and Levesque begin by taking the usual semantic net primitives (in their terminology, *objects* are instances of *classes* and have *relations* between them), and they determine to associate programs with them to permit inferencing in an efficient and modular way. There are eight operations which the programs implement, four on relations and four on classes:

to assert that a relation holds between two objects;
to assert that a relation no longer holds between two objects;
to fetch all objects related to another by a given relation;
to test whether a given relation holds between two objects;
to create an object as an instance of a class;
to destroy an instance of a class;
to fetch all instances of a class;
to test whether an object is an instance of a class.

Here we have our functional interface, the interface to the knowledge base by means of TELL and ASK operators which Levesque was later to formalize.

Mylopoulos and Levesque go on to discuss hierarchies within their model, specifically IS-A (taking account of the assertional/structural distinction) and PART-OF. They introduce metaclasses, unifying the whole representation scheme and making the handling of inheritance more consistent. Finally they consider the nature of the programs attached to classes (which now include classes themselves and relations). There is some divergence of interest from the purely functional approach here, but the idea of having operators defining behaviour is exactly right.

Procedural semantics in the sense of Mylopoulos and Levesque came to fruition in the system unimaginatively called PSN, developed at Toronto since 1976 (Mylopoulos *et al.*, 1983). The system is based very closely on the ideas described above. Instances of classes are now called tokens; relations are defined between classes and are instantiated to form links between tokens (Figure 4). Programs specify operations on classes and relations, exactly as above. There are three primitive relations for creating hierarchies: INSTANCE-OF (relating tokens to their classes), IS-A (with inheritance) and PART-OF ('aggregation', using slots to represent parts of a concept). The attached programs are themselves objects, and have their own class with slots (for parameters, prerequisites and actions). A new idea in PSN is the *similarity link*, which suggests other classes to be tried when a match fails, and may owe something to Minsky's suggestion of 'sharply localized knowledge that would naturally be attached to a frame itself for recommending its own replacement' (Minsky, 1981).

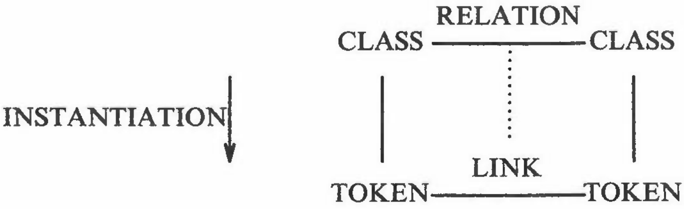


Figure 4 Entities and Relationships in PSN

A number of applications have been developed using PSN. Mentioned in (Mylopoulos *et al.*, 1983) are two in the domain of cardiology.

Before going on to look at Levesque's later work, when he abandoned the procedural aspect and considered the functional interface alone, it is worth mentioning some work by Rich, described in (Rich, 1982). Here we see another theme of importance in our area, that of mixed representations. Rich was working on the Programmer's Apprentice at MIT, for which a representation scheme called *plan diagrams* had been devised to suit the type of knowledge being represented. 'During this period we took a fairly *ad hoc* approach to the semantics of our knowledge representation. This is not to say that we did not know what plan diagrams meant, but just that ultimately the meaning of the representation was implicit in the procedures we were writing to manipulate it.' Rich began to develop a formal semantics for plan diagrams using predicate calculus, and found that the two representations could co-exist in the system implementation. It is a hybrid representation, in which both levels of language are used for various levels of reasoning in the application domain, depending on which is appropriate. A mixed representation scheme, though on a more formal footing, is an important of feature of KRYPTON, described in section 10.5.

Procedural semantics played an important part in the evolution of functional approaches to knowledge representation, but ultimately it is rather limited because it is not suitable for theoretical amplification. Rich makes no bones about this:

> 'Before going any further it is crucial to understand that this paper is about pragmatic rather than philosophical issues in knowledge representation.'

His concluding remarks are couched as advice for representation designers. Similarly, the procedural semantics of Mylopoulos and Levesque, though undoubtedly fruitful, seems unlikely to lead to any new directions. Defining behaviour by means of programs is clearly advantageous for efficiency of search, but there is not much one can say about it. The programs seem to be at too low a level, despite attempts to impose structure on them. It is not clear how the approach can be extended beyond the pragmatic. Indeed, Levesque realized this when he came to investigate the functional interface in isolation, and it is to his work that we now turn.

10.4 Levesque's Formalization

In the preceding section we have seen that Mylopoulos and Levesque's procedural semantics attached programs to classes and relations in a knowledge representation scheme in order to perform basic operations on those classes and relations. It is these operations that constitute the functional interface, and Levesque's next step was to abandon the procedural aspect and investigate what could be said in the abstract about a knowledge base with such an interface. In a lengthy and important paper (Levesque, 1984) he describes his results.

Levesque begins by specifying the interface by means of two operators TELL and ASK, the former asserting that a statement (made in some language L) is true in the knowledge base, the latter querying whether this is the case:

$$\begin{aligned} &\text{TELL: } \mathrm{KB} \times \mathrm{L} \rightarrow \mathrm{KB}; \\ &\text{ASK: } \mathrm{KB} \times \mathrm{L} \rightarrow \{\text{yes, no, unknown}\}. \end{aligned}$$

He goes on to discuss the requirements on the language L, paying particular attention to the need to represent incomplete knowledge. Arguing that 'it must be possible for the KB to find out about the world in an incremental way', he maintains that TELL and ASK must permit weak statements about the world: saying what something is not, for instance, or giving a range of possible things it might be. A formal discussion of the semantics and proof theory of the language L follows; then Levesque outlines some of the problems arising from the use of L (roughly speaking, if it is possible to make weak statements to the KB, it may return unhelpfully weak answers). He introduces an operator which applies to a sentence of L and returns 'true' if the sentence is currently known in the knowledge base, 'false' if not, and he defines an extended language KL incorporating this operator. Making assumptions of 'competence' (that every consequence of what is known is itself known) and 'closure' (that a pure sentence is true exactly when it is known), semantics is defined for the extended language, and the operators TELL and ASK are redefined.

The argument then takes another dive into formalism: Levesque considers what sort of knowledge (in a formal sense) is representable in his language, and having established it, asks what effect a TELL operation has, and how ASK works. The result is a 'Representation Theorem', which states in essence that communication with the knowledge base in the language KL may be achieved completely in first order terms, just as with the original L.

Having surfaced from the proof of his Representation Theorem, Levesque considers possible extensions to the theory he has established. The most immediately interesting concerns default reasoning. Levesque proposes two possible lines of attack by extending respectively the ASK and TELL operators to take defaults into account. He introduces an operator which, applied to a one-place predicate, yields a predicate of being a 'typical example'. It is possible to assert that 'typical birds fly' and 'typical birds have two legs and two wings' using the operator - and these are necessary properties of typical birds, not just typical properties. In Levesque's words, 'all of the "content" of the default is put into knowledge about the properties of typical instances of the predicates'. The 'is known' operator is now the key, for the representation language can express that if an entity is not known to be atypical then it should be treated as typical - hence default reasoning.

As a final aside, Levesque mentions the possibility of defining new terms from existing ones. His formalization is complete, and we can consider the system which embodies much of it. KRYPTON has a functional interface with TELL and ASK operators, a powerful inferencing mechanism, and a separation of definitions from assertions about the world, and we go on to examine it now.

10.5 KRYPTON

KRYPTON is an experimental knowledge representation system, chiefly the work of Brachman and Levesque (Brachman *et al.*, 1983a, 1983b, 1983c, 1985b). Its origins lie in KL-ONE (Brachman and Schmolze, 1985c) which is a highly influential representation system founded on a formalization of the ideas of frames and semantic nets and intended to allow the formation of complex structured descriptions. Manifesting Brachman's interest in the semantics of such terms, KL-ONE pays particular attention to 'concepts', 'descriptions', 'attributes' and the like. It has undergone several implementations in a variety of languages, and has been used in a number of applications. Its emphasis is very much on the description of concepts by structured inheritance networks at the expense of an assertional capability, and although the distinction was gradually recognized and some account was taken of it, the two have never been on an equal footing. We have already seen how a purely descriptive approach is defective, and KRYPTON is an attempt to combine it with an assertional approach, clearly defining the responsibilities of each and their interrelation.

KRYPTON also has a functional interface with TELL and ASK operators like those of Levesque's work described in the last section. Published reports on the work differ in the emphasis they place on the functional interface and the mixed representation scheme: the papers (Brachman *et al.*, 1983a, 1983b), for instance, are entitled 'KRYPTON: a functional approach to knowledge representation', whereas (Brachman *et al.*, 1983c) is 'KRYPTON: integrating terminology and assertion' and (Brachman *et al.*, 1985b) is 'An essential hybrid reasoning system'. In fact the two ideas are orthogonal, but it is perhaps natural that they should be associated, for the interface between assertional and definitional components is certainly the kind of semantic minefield that Brachman would like to see guarded by a firm functional interface.

KRYPTON's representation involves two components called the TBox and the ABox, intended respectively for structured definitions ('terminology') and for making assertions. The ABox assertions refer to terms defined in the TBox in order to make their statements. Before examining the operators available for communicating with a KRYPTON knowledge base, we shall look briefly at the TBox and ABox in turn.

The TBox language is essentially that of frames, with an important difference. There are concept expressions which correspond roughly to frames and role expressions which are the equivalent of slots. Importantly, there is no direct access to the value of a slot, and hence no danger of these frames becoming data structures; rather, new concepts and roles are formed by combining or restricting others. A range of operators is available for this purpose (not to be confused with the operators defining the functional interface to the system as a whole). Some examples will illustrate this. One of the operators is ConGeneric, which yields a concept which is the conjunction of the concepts which are its arguments. Thus a bachelor might be defined as

(ConGeneric man unmarried-person)

where 'man' and 'unmarried-person' are pre-existing concepts. Another operator is VRGeneric, whose arguments are a concept, a role and another concept, and which returns the first concept restricted so that all specified roles are instances of the second concept.

(VRGeneric paper author scientist)

yields the concept of a paper all of whose authors are scientists. A third operator is RoleChain.

(RoleChain child child)

would yield a 'grandchild' role. Other operators are described in the earlier papers, but it appears that only these three have been fully implemented. It can be seen that considerable power is available for making structured descriptions, but in a form very different from usual frame systems.

Another interesting feature of the TBox is in its handling of 'primitive' concepts and roles. Obviously the construction of concepts and roles has to start somewhere, and the basic ones chosen may be entirely independent of each other. However, it is possible to declare a new concept or role to be a primitive specialization of an existing one, meaning that any instance of the new type is necessarily an instance of the old, but there are no sufficient conditions for determining membership of the new type. Thus an elephant might be declared to be a primitive specialization of a mammal: all elephants will be mammals, but the system will not be able to deduce that anything is an elephant unless explicitly told so. This is KRYPTON's rather defensive answer to the problem of 'natural kinds' - it favours safety and simplicity at the expense of expressiveness.

The ABox is a language for making assertions about the world. It is in fact standard first order predicate calculus, but the basic non-logical symbols are not mere atoms but refer to the terms of the TBox. Because the language is a logical one, incomplete knowledge may be expressed using the usual operators of disjunction, negation and existential quantification.

Operators are provided to define the functionality of a knowledge base from a user's point of view. The ABox has the expected TELL and ASK, the former asserting that some sentence is true, the latter querying it. The corresponding operators for the TBox are called DEFINE and SUBSUMES. DEFINE is used for setting up definitions of concepts and roles, as in the examples above, while SUBSUMES queries whether one TBox term is subsumed by another (as for instance 'bachelor' is subsumed by 'unmarried-person'). There are in fact other TBox operators which return sets of symbols rather than just a truth value.

The two representation schemes of KRYPTON are tightly integrated, meaning that there is no simple translation from one to the other (as, for instance, TBox definitions might be re-expressed as logical sentences indistinguishable from ABox statements). Rather, the two are kept separate and their interrelation is closely defined. There is a requirement for *competence* in deriving conclusions from given definitions and assertions, in that using the definitions and their relationships together with its (incomplete) knowledge in the form of ABox assertions, KRYPTON can answer correctly quite general queries. The structure of the whole system is shown diagrammatically in Figure 5. The principle of having ABox sentences refer to TBox terms seems straightforward enough, but as Brachman *et al.* (1985b) observe:

> 'It is not enough to say that KRYPTON has a frame-style description language for forming terms and a first-order predicate language for forming sentences - we must explain how the interpretations of the sentences by the theorem prover depend on the definitions of the terms.'

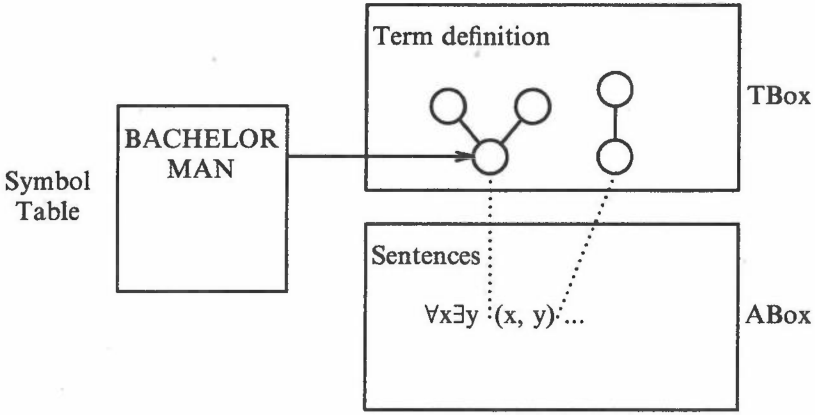


Figure 5 The Structure of KRYTON

Some effort is expended on formalizing the hybrid semantics of the two components, and the TELL and ASK operations are then defined in terms of it. It is possible to prove certain simple results about these operations that are clearly desirable, for instance that a term subsumes any ConGeneric involving it.

The implementation issues of KRYPTON are complicated and this is not the place to discuss them in detail, involving as they do unification algorithms taking account of the hybrid representation scheme. However, there are some points worth making. Because of the functional interface, the way in which reasoning is implemented is irrelevant as long as it yields the required behaviour as defined by KRYPTON's semantics. The ABox incorporates a specialized theorem prover, Stickel's Connection Graph theorem prover, for drawing its inferences, but there is no reason why it could not be replaced by an equivalent mechanism, or why performance could not be improved by preceding use of the theorem prover with an efficient database lookup. In the TBox, the relation of subsumption between terms is of great importance, and a classifier may be used (as in KL-ONE) to place newly defined terms in their correct places in the taxonomy, but the details of how it works are independent of the semantics it implements.

KRYPTON is very much a research tool. It would almost certainly not be possible to develop large applications in its current state. In some areas it is not complete - the theorem prover is only partially integrated with the terminological component. There is no doubt that KRYPTON has been a valuable experiment in representing incomplete knowledge, in functional interfaces and in responding to the fact that 'an intelligent system has more than one kind of representation need'. Its distinction between definitions and assertions, which have no definitional import even if expressed as 'universally quantified biconditionals', is attractive. Yet such a distinction may not always be appropriate for representing certain kinds of knowledge: it may be suitable for an abstract domain like geometry in which one can define a triangle precisely and then state properties of particular triangles, but when one comes to look at natural kinds the situation is less clear. We enter the domain of stereotypes, defaults and redundancy in definitional knowledge, and the structural/assertional distinction begins to look unsure. However, a full discussion is well outside the scope of this chapter, and we can conclude by remarking that KRYPTON is the most advanced implementation of functional ideas, though whether the principle has a future, and how it will resolve with such issues as mixed representation schemes, it is not yet possible to say.

11 Expressive Power and Computability

Tony Williams and Simon Lambert

11.1 Introduction

It should not be forgotten that all the knowledge representation formalisms introduced in previous chapters are intended for implementation on a computing machine, however ulterior their origins. That is how AI proves itself, and is the *raison d'être* of this book. Logic sprang from the head of Aristotle, while Newell and Simon looked into the heads of those around them and saw production rules. Semantic networks too have been proposed as 'models of cognition'. Yet all are amenable to encoding and manipulation within a computer program. The variety of manipulations permitted is of great importance, for a knowledge representation system must do more than just represent; it must be able to respond to queries about what it represents. Algorithms are needed to act upon it: and the study of their properties leads us into one of the provinces of mathematics. It may be that there is no algorithm guaranteed to terminate for a particular task, or that its intrinsic resource requirements are hugely expensive. The complexity of the tasks will depend on how much is expected of the knowledge representation system: there is a trade-off between computability and expressive power, and it has been explored by Levesque and Brachman in their paper (Levesque and Brachman 1985). This chapter serves as an informal introduction to their work, and attempts to relate it to some of the subjects described elsewhere in this book.

11.2 Setting the Scene: What's in a Knowledge Base

Levesque and Brachman start from the Knowledge Representation Hypothesis formulated by Brian Smith (Smith, 1982). It states that an intelligent system has components that:

(a) appear to contain a propositional representation of the knowledge that the system as a whole possesses;

(b) cause the system to behave in a way that manifests that knowledge.

A knowledge-based system satisfies this hypothesis by design. Its knowledge representation component is the subsystem that maintains knowledge in some explicit representation, called the knowledge base. (The separation of the knowledge representation component from the rest of the system is very characteristic of Levesque and Brachman's work. It is the essence of the 'functional' approach to knowledge representation described in Chapter 10.) The knowledge representation subsystem is in general more than just a database manager, for it has inference mechanisms enabling it to answer queries whose results are not explicitly stored as facts in the knowledge base. Logic has its rules of inference, while semantic networks lend themselves to operations which we would think of graphically, such as the pursuit of IS-A links.

According to Levesque and Brachman, the knowledge representation system should be capable of accepting new knowledge and incorporating it into the knowledge base; and they mention too the possibility of having it contain reasoning tactics separate from the declarative domain knowledge. Compare the definition of 'ancestor' as the transitive closure of 'parent' with the reasoning tactic that says that to determine the truth of

X ancestor-of Y

it is better to search up from Y rather than down from X. Levesque and Brachman are unsure how such reasoning tactics may be represented, suggesting that in practice they will tend to be implicit, or else take advantage of the sort of pragmatically motivated features that cause Prolog to differ from first-order predicate calculus. There has, however, been some work on the control of reasoning within rule-based systems, described in Chapter 8.

Given that the knowledge representation component is seen as a subsystem, it should be dependable. That is to say, it should respond to queries with results that are 'correct' according to the knowledge it contains. Furthermore, its resource consumption, such as the time taken to respond to a query, should not grow unmanageably as the size of the knowledge base increases.

11.3 First Order Logic

A good place to start our exploration of expressiveness and computability is with first order logic (FOL, see Chapter 2). It permits statements of very unrestricted scope, and its expressive power lies not so much in the propositions that can be expressed directly as in those that need not be explicitly stated. The rules of inference enable the knowledge representation system to generate the implicit information when it is required. As an example, consider,

$$\forall\ x\ Friend(George, x) \Rightarrow \exists\ y{:}Child(x, y)$$

which states that all George's friends have children, without stating who those friends are or even that there are any. Given that $\forall\ x \neg\ Child(Harry, x)$, one can deduce (in FOL) that $Friend(George, Harry)$ is false, without any explicit knowledge of who George's friends are. Similarly, the sentence

$$Child(Ian,\ Anne)\ or\ Child(John,\ Anne)$$

asserts that Anne is the child of Ian or John but without specifying which. This property of providing expressive power through the use of implicit information is not unique to FOL, but is used (perhaps in weaker form) in any knowledge representation system that will perform inference. As we shall see, it appears to be a major cause of computational intractability.

A fundamental property of first order logic is that the question of whether or not a statement is implicit in the knowledge base is equivalent to whether the corresponding sentence is a theorem. Answering a query becomes theorem proving: the statement to be queried is expressed as a proposition, and the system attempts to prove it from the axioms ('facts') it contains. There is a problem, though, in that provability in FOL is 'semi-decidable', meaning that, although a suitable procedure can always prove the theoremhood of a sentence that does follow from the axioms (see e.g. (Bundy, 1983) for a proof of the soundness and completeness of resolution theorem proving), it cannot be guaranteed to terminate when presented with one that does not. In other words, for some queries the knowledge representation system might simply not respond. The problem of undecidability of FOL is related to Gödel's famous results for formal arithmetic. However, any system whose power is equivalent to or greater than arithmetic is not only undecidable but also incomplete, in that there are sentences that can be neither proved nor disproved. See, for example (Rosser, 1939), or for a more entertaining account including relations to many other phenomena (Hofstadter, 1979).

Even when the query is answerable (i.e. it represents a theorem in FOL), the computational expense of proving it may be too great. An interesting example of a problem which is effectively intractable for a standard FOL system is Schubert's Steamroller. In English, it is stated as follows:

> Wolves, foxes, birds, caterpillars and snails are animals, and there are some of each of them. Also there are some grains, and grains are plants. Every animal likes to eat either all plants or all animals much smaller than itself that like to eat some plants. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not snails. Caterpillars and snails like to eat some plants. Therefore there is an animal that likes to eat a grain-eating animal.

The problem may be easily axiomatized in FOL, using predicates for 'is-wolf', 'is-fox', etc., and 'is-animal', 'is-plant', 'likes-to-eat' and 'is-much-smaller-than'. It is possible to prove the final proposition by hand, but it has utterly defeated all resolution theorem provers because the search space is just too large. Many-sorted logics do permit a solution (Walther, 1985), as does the KRYPTON system described in Chapter 10.

It seems therefore that knowledge representation systems with expressive power equivalent to FOL are not dependable in the sense given above, though it should be noted that the intractability represents worst case behaviour, and that many queries will terminate quickly. One might question whether FOL-equivalent systems are useless: the answer must be that it depends on the problem the system is designed to solve. For example, if one were trying to find a proof for Fermat's Last Theorem, one might be happy to leave a FOL system running for several months, looking periodically to see if it appeared to be making progress and perhaps redirecting it if not. On the other hand, a robot must not get bogged down trying to prove or disprove a low-level subgoal, because it must come to a decision about what to do within a defined amount of time. Aeroplanes and nuclear power plants will not wait.

One could ensure that the knowledge representation system returns some answer within a definite time limit, returning 'unknown' if the decision procedure has not terminated. But if this solution is adopted, it becomes difficult to characterize the class of queries the system can answer. Attempting to make the system dependable in resource consumption by this means will compromise its 'correctness'.

11.4 Limiting Expressive Power

Levesque and Brachman distinguish between queries about the information so stored ('Is Harry included in the list of George's friends?') and queries about the world which the knowledge base is supposed to represent ('Is Harry a friend of George?'). The distinction is important when the knowledge base does not explicitly store complete information about the world; database form is completely incapable of expressing incomplete information, but retrieval of what it does contain is computationally inexpensive. An alternative to the arbitrary termination of queries after a time limit is to achieve termination with a valid answer by restricting the inferential capability of the knowledge representation system. This of course limits the range of queries that can be made of it. It is possible to circumscribe the computational complexity of the inference procedure by limiting the degree of unstated information that can be used in inference. Instead, such information must be explicitly present in the knowledge base. An extreme case is the database form, where all information that is to be retrieved must be explicitly stored.

Databases and full first order logic are two widely separated points on the trade-off between computability and expressiveness. Levesque and Brachman consider three other formalisms in the same context: logic programming, exemplified by Prolog; semantic networks; and frame systems. In each case they are careful to point out their logical foundations, explicit and implicit (Prolog's Closed World Assumption, for example). This preoccupation with clearly defined semantics is very characteristic of Brachman's work; indeed he and Levesque are rather dismissive of inference mechanisms suggested by the knowledge representation formalisms themselves and lacking such a foundation. Of semantics networks, they say (*op. cit.*),

> 'For better or worse, the appeal of the graphical nature of semantic nets has led to forms of reasoning (such as default reasoning) that do not fall into standard logical categories and are not yet very well understood. This is a case of a representational notation taking on a life of its own and motivating a completely different style of use not necessarily grounded in a truth theory. It is unfortunately much easier to develop algorithms that appear to reason over structures of a certain kind than to justify its reasoning by explaining what the structures are saying about the world.'

And of frames,

> 'Like semantic networks, frame languages tend to take liberties with logical form and the developers of these languages have been notoriously lax in characterizing their truth theories'.

Though they mention three of what many people would regard as major features of frame systems - default values, restrictions on slots, and attached procedures - it is only to dismiss them when formalizing their own system for the purposes of exploring expressiveness and computability, as we shall see in section 11.5. This does not affect their argument, for the services they expect from their system would probably have to be satisfied by any frame system, and indeed the restrictions are necessary to allow precision in establishing criteria for comparison. One might feel that frames have lost something in the process; but the important conclusion is that when the semantically doubtful accretions have been jettisoned all the above formalisms can be seen as restrictions of first order logic, exhibiting various degrees of expressiveness and computational tractability.

11.5 An Illustration of the Trade-off

In the context of frame systems, Levesque and Brachman have constructed an example to illustrate the trade-off. They define a 'frame description language' similar to that provided by KRYPTON's TBox (see Chapter 10), which allows the user to build composite type definitions from existing types and attributes, starting from some set of primitives (a 'type' defines a set of frames; an attribute corresponds to a slot in a frame). To be specific, a type may either be an atomic symbol or take one of the following forms:

(AND type-1 type-2 ... type-n)
(ALL attribute type)
(SOME attribute).

An attribute can itself be an atom or have the form:

(RESTRICT attribute type).

(AND type-1 ... type-n) is a type denoting the set of frames which are members of all the types listed. That set is the intersection of the sets denoted by the individual types. (AND doctor male) denotes the set of male doctors.

(ALL attribute type) is a type which denotes the set of things for which, if they have the given attribute, its value is of the given type. For example, (ALL friend doctor) denotes the set of frames whose 'friend' attributes (if any) all have type 'doctor'.

(SOME attribute) is a type which denotes the set of frames which have that given attribute, whatever its value might be. For example (SOME friend) is the set of frames that have any 'friend' slot.

(RESTRICT attribute type) defines a new attribute from the old one by requiring its values to be of the stated type. (RESTRICT friend doctor) defines an attribute 'friend who is a doctor'. Forms using RESTRICT are appropriate for use in constructing type expressions in the above compound forms, particularly ALL. For example, (ALL (RESTRICT friend male) doctor) denotes everyone all of whose male friends (if any) are doctors. Nothing is determined about friends who are not male. (SOME (RESTRICT friend male)) denotes everyone with at least one male friend, irrespective of the types of their other friends. The RESTRICT operator therefore provides a way of qualifying expressions leaving certain information unspecified.

The above constructs can be used to create complex descriptions of frames:

```
(AND person
     (ALL  (RESTRICT  friend  male)
           (AND doctor
                (SOME  speciality)
           )
     )
)
```

denotes the set of frames of type 'person' for which each 'friend' attribute of type 'male' (if any) is of type 'doctor' and has an attribute 'speciality', i.e. every person whose male friends are all specialist doctors. There may be frames in the resulting set that have friends who are not doctors with a speciality, but those friends will not be male.

One might well ask what on earth this little language has to do with the frames of Minsky and those who followed him, as described in Chapter 4. The point is that Levesque and Brachman have to be precise about what they mean by expressiveness and computational complexity. They admit that the frame description language is highly restricted, but at least it meets some of the possible requirements on a general frame system. They are able to furnish it with a formal semantics, and to define the idea of 'subsumption' between two types, which they use in their analysis of complexity. Subsumption is a simple idea, and is essentially set inclusion: one type subsumes another if all instances of the second type are necessarily instances of the first. For example, (AND doctor male) subsumes (AND doctor (ALL friend female) male).

The language defined above with its operators AND, ALL, SOME and RESTRICT is called *FL*. Levesque and Brachman denote by FL^- the language without the RESTRICT construction. Not surprisingly, the loss of RESTRICT means that there are some frame descriptions that can be expressed in *FL* but not in FL^-: so *FL* is more expressive. To show this in some more detail, we can examine the forms in which RESTRICT can be used. To take the earlier example, (ALL (RESTRICT friend male) doctor) would have to be written without RESTRICT as something like

```
(ALL friend  (OR
                  (AND male doctor)
                  (NOT male)
             )
)
```

requiring negations and disjunctions (with suitable definitions). (SOME (RESTRICT friend male)), by contrast, has no obvious representation even using OR and NOT.

But there is a price to pay. For they show that the operation of determining whether one type subsumes another is perfectly tractable in *FL^-* (being $O(n^2)$) but not in *FL*, in which it is technically co-NP hard (for an introduction to the complexity of algorithms, including the significance of NP-complete and NP-hard problems, see for instance (Machtey and Young, 1978)). Levesque and Brachman prove their results by, in the first case, producing an algorithm and analysing it, and in the second case by showing equivalence to the problem of deciding logical implication, whose complexity is strongly believed to be intractable (detailed proofs are given in the augmented paper (Levesque and Brachman, 1987)). These two methods of proof are unrelated and so do not together show why the addition of the RESTRICT operator causes the threshold of intractability to be crossed, but some light is shed on the matter by examining the algorithm for computing subsumption in *FL^-*. (The authors are indebted to Ronald Brachman for discussing this line of work, currently in progress. Any errors in this discussion are the fault of the authors, not of Brachman.)

The algorithm proceeds by converting an expression into a 'flattened' form, by combining nested AND expressions and collecting together ALL expressions that have the same attribute. For example,

```
(AND (ALL  friend  (AND male
                        redhead
                        athlete
                   )
     ) -- people whose friends are all male redheaded athletes
     doctor
     (ALL  friend  (AND ambidextrous
                        blind
                   )
     ) -- people whose friends are all blind and ambidextrous
)
```

denotes doctors whose friends are all blind male redheaded ambidextrous athletes. This can be rewritten as

```
(AND doctor
      (ALL  friend  (AND  male
                          redhead
                          athlete
                          ambidextrous
                          blind
                    )
      )
)
```

We can see that the above expression is subsumed by

```
(AND doctor (ALL  friend  redhead))
```

by determining that the ALL expressions refer to the same attribute and that 'redhead' subsumes the AND expression. It can be shown that the conversion to flattened form can be performed in $O(n^2)$ time, and that subsumption of flattened forms can be determined in the same time complexity. The subsumption algorithm is recursive in the case of ALL expressions, as (ALL a1 t1) subsumes (ALL a2 t2) if and only if a1 = a2 and t1 subsumes t2. The flattened form ensures that, at each level of recursion, the size of the problem is reduced.

We now consider the example modified as follows:

```
(AND (ALL  (RESTRICT   friend  male)
                 (AND  redhead
                       athlete
                 )
      ) -- people whose male friends are all redheaded athletes
      doctor
      (ALL  friend  (AND  ambidextrous
                          blind
                    )
      ) -- people whose friends are all blind and ambidextrous
)
```

This denotes doctors all of whose friends are blind and ambidextrous, but only the male ones need be redheaded athletes. The restriction on the friend attribute means that the ALL expressions cannot be combined, and the subsumption algorithm must examine these attributes separately for each such expression. Subsumption of the modified expression by

```
(AND doctor (ALL  friend  redhead))
```

can only be determined by establishing whether there are any doctors with non-male friends. This type of determination could potentially be as complex as the original problem, and so the problem does not necessarily reduce in complexity with each recursion.

11.6 Conclusion

Knowledge representation formalisms may be viewed as forming a spectrum of varying inferential power. Databases fall at the low end, and the scale goes through frame languages, logic programming and other schemes up to first order logic and beyond. As one moves along this scale the computational complexity of answering queries about the knowledge base increases, and eventually the problem becomes intractable. It is not yet known how to categorize a knowledge representation scheme into its position on the scale until it is completely specified. It appears that the ability to use information not explicitly stored, but inferrable from other information, adds to the expressive power, but is a major contributor to the computational complexity.

There are two implications. Firstly, it remains useful and interesting to develop knowledge representation formalisms which are subsets of FOL in order to explore this dimension of computability. There may be knowledge representation systems which are computationally tractable, and sufficiently expressive to be useful in some domain. Secondly, if such studies show that inference becomes intractable for any useful knowledge base, the Knowledge Representation Hypothesis would have to be reconsidered. It may be that intelligent systems which can operate in real time will be composed of some number of simpler, tractable representation and reasoning components, with some sort of overseer which arbitrates among them and endows the system as a whole with apparently intelligent behaviour.

If the provision of a full inferencing capability is liable to be intractable, perhaps some restricted capability should be offered. In the words of Levesque and Brachman (*op. cit.*),

> 'Instead of automatically performing the full deduction necessary to answer questions, a knowledge representation system could manage a *limited form of inference* and leave to the rest of the knowledge-based system (or to the user) the responsibility of intelligently completing the inference.'

Of course, what is meant by this is not at all clear. Just as they shy away from those features of frames that have not (yet) been given a clear semantics, so Levesque and Brachman are hesitant when faced with this prospect:

> 'First of all, it is far from clear what primitives should be available... Finding such a service that can be motivated *semantically* (the way logical deduction is) and defined independently of how any program

actually operates is a non-trivial matter, though we have taken some steps towards this...'

Collected References

Aeillo, N. (1983), "A comparative study of control strategies for expert systems : AGE implementation of 3 variations of PUFF", *Proceedings of the Third National Conference on Artificial Intelligence*, pp.1-4.

Aikins, J.S. (1980), "Prototypes and production rules : A knowledge representation for computer consultations", Ph.D. dissertation, Stanford University. (Also Stanford Report no. STAN-CS-80-814.)

Aikins, J.S. (1983), "Prototypical Knowledge in Expert Systems", *Artificial Intelligence* **20**(2), pp.163-210.

Allen, J.F. (1981), "An interval-based representation of temporal knowledge", *Proc. 7th IJCAI*, Morgan Kaufmann: Los Altos, CA.

Allen, J.F. (1983), "Maintaining Knowledge about Temporal Intervals", *Commun. ACM* **26**(11), pp.932-843.

Allen, J.F., and Koomen, J.A. (1983), "Planning using a temporal world model", *Proc. 8th IJCAI*, Morgan Kaufmann: Los Altos, CA.

Anderson, J.R., and Bower, G.H. (1973), *Human Associative Memory,* Winston and Sons: Washington, DC.

Anderson, J.R. (1976), *Language, Memory and Thought,* Lawrence Erlbaum and Associates: Hillsdale, NJ.

Anderson, J.R. (1978), "Arguments concerning representations for mental imagery", *Psychological Review* **85**, pp.249-277.

Anderson, J.R. (1983), *The Architecture of Cognition,* Harvard University Press: MA.

Anderson, J.R. (1985), *Cognitive Psychology and its Implications,* Freeman: New York.

Anderson, R.H., and Gillogly, J.J. (1976), "Rand Intelligent Terminal Agent (RITA): Design Philosophy", R-1809-ARPA, Rand Corporation, Santa Monica, CA.

Bachant, J., and McDermott, J. (1984), "R1 revisited: four years in the trenches", *AI Magazine* **5**(3), pp.21-32.

Baddeley, A.D. (1976), *The Psychology of Memory,* Harper and Row: New York.

Baddeley, A.D. (1983), "Working Memory", *Philosophical Transactions of the Royal Society London B* **302**, pp.311-324.

van Bakel, J., and Hoogeboom, S. (1981), "Eksperiment met een Kasus-Grammatika", pp. 1-57 in *Verslagen Computerlinguistiek 2 (Katholieke Universiteit Nijmegen).*

Bartlett, F.C. (1932), *Remembering,* Cambridge University Press: Cambridge.

Bekerian, D.A., and Bowers, J.M. (1983), "Eyewitness Testimony: Were We Misled?", *Journal of Experimental Psychology: Learning, Memory and Cognition* **9**(1), pp.139-145.

van Benthem, J.F.A.K. (1982), *The Logic of Time,* D. Reidel Publishing Co.: Dordrecht, Holland.

Berger, H. (1929), "Uber das Ellektrenkephalogramm des Menschen", *Archiv für Psychiatrie und Nervenkrankheiten* **87**, pp.527-570.

Berlin, B., and Kay, P. (1969), *Basic Colour Terms: Their Universality and Evolution,* University of California Press: Berkeley and Los Angeles, CA.

Binot, J.L., Graitson, M., Lemaire, P., and Ribbens, D. (1980), "Automatic processing of written French language", pp. 9-14 in *COLING 80 (Proceedings of the 8th International Conference on Computational Linguistics).*

Bobrow, D.G., and Winograd, T. (1977), "An Overview of KRL", *Cognitive Science* **1**, pp.3-46.

Bobrow, D.G., and Winograd, T. (1979), "KRL: Another Perspective", *Cognitive Science* **3**, pp.29-42.

Bobrow, D.G., and Stefik, M. (1983), *The LOOPS Manual,* Xerox Corporation.

Bonnet, A. (1985), *Artificial Intelligence: Promise and Performance,* Prentice-Hall: New York.

Boolos, G.S., and Jeffrey, R.C. (1980), *Computation and Logic, 2nd Edition,* Cambridge University Press: Cambridge.

Bower, G.H., Black, J.B., and Turner, T.J. (1979), "Scripts in memory for text", *Cognitive Psychology* **11**, pp.177-220.

Brachman, R.J. (1977), "What's in a concept: Structural foundations for semantic networks", *International Journal of Man-Machine Studies* **9**, pp.127-152.

Brachman, R.J. (1979), "On the epistemological status of semantic networks", pp. 3-50 in *Associative Networks: Representation and Use of Knowledge by Computers*, ed. N. V. Findler, Academic Press: New York.

Brachman, R.J. (1983), "What IS-A is and isn't: an analysis of taxonomic links in semantic networks", *IEEE Computer* **16**(10), pp.30-36.

Brachman, R.J., Fikes, R.E., and Levesque, H.J. (1983a), "KRYPTON: a functional approach to knowledge representation", Technical Report No. 16, Fairchild Laboratory for Artificial Intelligence, Palo Alto, CA.

Brachman, R.J., Fikes, R.E., and Levesque, H.J. (1983b), "KRYPTON: a functional approach to knowledge representation", *IEEE Computer* **16**(10), pp.67-73.

Brachman, R.J., Fikes, R.E., and Levesque, H.J. (1983c), "KRYPTON: integrating terminology and assertion", *Proc. AAAI-83*, Morgan Kaufmann: Los Altos, CA.

Brachman, R.J. (1985), "'I Lied About the Trees' Or, Defaults and Definitions in Knowledge Representation", *AI Magazine* **6**(3), pp.60-93.

Brachman, R.J., and Levesque, H.J. (1985a), *Readings in Knowledge Representation,* Morgan Kaufmann: Los Altos, CA.

Brachman, R.J., Pigman Gilbert, V., and Levesque, H.J. (1985b), "An essential hybrid reasoning system: knowledge and symbol level accounts of KRYPTON", *Proc. 9th IJCAI*, Morgan Kaufmann: Los Altos, CA.

Brachman, R.J., and Schmolze, J.G. (1985c), "An overview of the KL-ONE knowledge representation system", *Cognitive Science* **9**(2), pp.171-216.

Bransford, J., and Franks, J.J. (1971), "The abstraction of linguistic ideas", *Cognitive Psychology* **2**, pp.331-356.

Broadbent, D. (1985), "A Question of Levels: Comment on McClelland and Rumelhart", *Journal of Experimental Psychology: General* **114**(2), pp.189-192.

Brooks, L.R. (1968), "Spatial and verbal components of the act of recall", *Canadian J. Psychol.* **22**, pp.349-368.

Brownston, L., Farrell, R., Kant, E., and Martin, N. (1985), *Programming Expert Systems in OPS5,* Addison-Wesley: London.

Bruce, B.C. (1972), "A Model for Temporal References and its Application in a Question Answering Program", *Artificial Intelligence* **3**, pp.1-25.

Buchanan, B.G., and Shortliffe, E.H. (1984), *Rule-Based Expert Systems : The MYCIN Experiments of the Stanford Heuristic Programming Project,* Addison-Wesley: London.

Bundy, A. (1982), "What is the well-dressed AI educator wearing now?", *AI Magazine* **3**(1).

Bundy, A. (1983), *The Computer Modelling of Mathematical Reasoning,* Academic Press: London.

Bundy, A., Burstall, R.M., Weir, S., and Young, R.M. (Eds.) (1980), *Artificial Intelligence: An Introductory Course: 2nd Edition,* Edinburgh University Press: Edinburgh.

Caramazza, A., Mc.Closkey, M., and Green, B. (1981), "Naive beliefs in 'sophisticated subjects': Misconceptions about trajectories of objects", *Cognition* **9**, pp.117-123.

Carbonnell, J.R. (1970), "AI in CAI: An artificial intelligence approach to computer-aided instruction", *IEEE Transactions on Man-Machine Systems* **MMS-11**(4), pp.190-202.

Carmichael, L., Hogan, H.P., and Walter, A. (1932), "An experimental study of the effect of language on the reproduction of visually perceived form", *Journal of Experimental Psychology* **15**, pp.73-86.

Chomsky, N. (1957), *Syntactic Structures,* Mouton: The Hague.

Chomsky, N. (1965), *Aspects of the Theory of Syntax,* MIT Press: Cambridge, MA.

Clancey, W.J. (1983), "The Epistemology of a Rule-Based Expert System - A Framework for Explanation", *Artificial Intelligence* **20**(3), pp.215-251.

Clancey, W.J. (1985), "Review of *Conceptual Structures in Information Processing in Mind and Machine*", *Artificial Intelligence* **27**(1), pp.113-124.

Clancey, W.J. (1986), "From GUIDON to NEOMYCIN and HERACLES in Twenty Short Lessons: ORN Final Report 1979-1985", *AI Magazine* **7**(3), pp.40-60.

Cohen, P.R. (1978), "On knowing what to say: planning speech acts", Technical Report no. 118, Dept. Computer Science, University of Toronto.

Collins, A.M., and Quillian, M.R. (1969), "Retrieval time from semantic memory", *Journal of Verbal Learning and Verbal Behaviour* **8**, pp.240-247.

Collins, A.M., and Quillian, M.R. (1970), "Facilitating retrieval from semantic memory. The effect of repeating part of an inference", *Acta Psychologica* **33**, pp.304-314.

Collins, A.M., and Loftus, E.F. (1975), "A spreading activation theory of semantic processing", *Psychological Review* **82**(6), pp.407-428.

Cooper, L.A., and Shepard, R.N. (1973), "Chronometric studies of the rotation of mental images", in *Visual Information Processing*, ed. W.G. Chase, Academic Press: New York.

Cooper, L.A., and Podgorny, P. (1976), "Mental transformations and visual complexity processes: Effects of complexity and similarity", *Journal of Experimental Psychology: Human Perception and Performance* **2**, pp.503-514.

Davis, R., Buchanan, B., and Shortliffe, E. (1977), "Production Rules as a Representation for a Knowledge-Based Consultation Program", *Artificial Intelligence* **8**(1), pp.15-45.

Davis, P.J., and Hersh, R. (1981), "Latakos and the Philosophy of Dubitability", in *The Mathematical Experience*, ed. P.J. Davis and R. Hersh, Harvester Press: Chichester, U.K.

De Jong, G. (1979), "A New Approach to Language Processing", *Cognitive Science* **3**(3).

Deliyanni,, and Kowalski, R.A. (1979), "Logic and Semantic Networks", *Commun. ACM* **22**(3), pp.184-192.

diSessa, A. (1982), "Unlearning Aristotelian physics: a study of knowledge-based learning", *Cognitive Science* **6**, pp.37-75.

Dowty, D.R. (1979), *Word Meaning and Montague Semantics,* D. Reidel Publishing Co.: Dordrecht, Holland.

Duncker, K. (1945), "On problem solving (transl. L.S. Lees)", *Psych. Monog*. **58**(5).

Ehrlich, K., and Johnson-Laird, P.N. (1982), "Spatial descriptions and referential continuity", *Journal of Verbal Learning and Verbal Behaviour* **21**, pp.296-306.

Etherington, D., and Reiter, R. (1983), "On Inheritance Hierarchies With Exceptions", *Proc. AAAI-83*, pp.104-108, Morgan Kaufmann: Los Altos, CA.

Evans, J.St.B.T. (1982), *The Psychology of Deductive Reasoning*, Routledge and Kegan Paul: London.

Evertz, R. (1982), "A Production System Account of Children's Errors in Fraction Subtraction", Computer Assisted Learning Research Group Technical Report No. 28, Open University: Milton Keynes, U.K.

Farber, D.J., Griswold, R.E., and Polonsky, I.P. (1964), "SNOBOL, a string manipulation language", *J. ACM* **11**(2), pp.21-30.

Fargues, J., Landau, M.C., Dugourd, A., and Catach, L. (1986), "Conceptual graphs for semantics and knowledge processing", *IBM Journal of Research and Development* **30**, pp.70-79.

Fillmore, C.J. (1966), "Toward a modern theory of case", pp. 361-375 in *Modern Studies in English: Readings in Transformational Grammar.*, ed. D.A. Reibel and S.A. Schane, Prentice-Hall: Englewood Cliffs, NJ.

Finke, R.A. (1985), "Theories Relating Mental Imagery to Perception", *Psychological Bulletin* **98**, pp.236-259.

Finke, R.A. (1986), "Mental Imagery and the Visual System", *Scientific American* **254**(3), pp.76-83.

Flannagan, T. (1986), "The Consistency of Negation as Failure", *Journal of Logic Programming* **2**, pp.93-114.

Fleming, M.L., and Hutton, D.W. (1973), *Mental Imagery and Learning*, Educational Technology Publications: Englewood Cliffs, NJ.

Floyd, R.W. (1961), "An algorithm for coding efficient arithmetic operations", *Commun. ACM* **4**(1), pp.42-51.

Forgy, C.L. (1981), "OPS5 Reference Manual", CMU-CS-81-135, Carnegie-Mellon University : Pittsburgh, Pennsylvania 15213, U.S.A.

Forgy, C.L. (1982), "A Fast Algorithm for the Many Pattern / Many Object Match Problem", *Artificial Intelligence* **19**(1), pp.17-37.

Freedman, R.S. (1982), *Programming Concepts with the Ada Language*, Petrocelli Books: New York.

Friedman, A. (1978), "Framing Pictures: the role of knowledge in automatic encoding and memory for gist", *Journal of Experimental Psychology: General* **108**, pp.316-355.

Frost, R.A. (1986), *Introduction to Knowledge Base Systems,* Collins.

Galambos, J.A., Abelson, R.P., and Black, J.B. (1986), *Knowledge Structures,* Lawrence Erlbaum Associates: Hillsdale, NJ.

Gale, W. (Ed.) (1985), *Artificial Intelligence and Statistics,* Addison-Wesley.

Gallaire, H., and Minker, J. (1978), *Logic and Databases,* Plenum Press: New York.

Garner, B.J., and Tsui, E. (1985), "Knowledge Representation from an Audit office", *Australian Computer Journal* **17**(3), pp.106-112.

Garnham, A. (1985), *Psycholinguistics: Central Topics,* Methuen: London.

Glanzer, M., and Clark, W.H. (1964), "The verbal loop hypothesis: Conventional figures", *Amer. J. Psychol.* **77**, pp.621-626.

Glass, A.L., and Holyoak, K.J. (1974), "Alternative conceptions of semantic memory", *Cognition* **3**, pp.313-339.

Golla, F., Hutton, E.L., and Walter, W.Grey (1943), "The objective study of mental imagery. 1. Physiological concomitants", *J. Ment. Sci. (cont. as Brit. J. Psychiat.)* **89**, pp.216-223.

Gould, P., and White, R. (1985), *Mental Maps,* Allen & Unwin: London.

Guildford, J.P., Fruchter, B., and Zimmerman, W.S. (1952), "Factor analysis of the Army Air Force's battery of experimental aptitude tests", *Psychometrika* **17**, pp.45-68.

Haber, R.N. (1979), "Twenty years of haunting eidetic imagery: where's the ghost?", *The Behavioural and Brain Sciences* **2**, pp.583-629.

Hanks, S., and McDermott, D. (1986), "Default Reasoning, Nonmonotonic Logics and the Frame Problem", *Proc. AAAI-86*, pp.328-353, Morgan Kaufmann: Los Altos, CA.

Hart, R.A., and Moore, G.I. (1973), "The development of spatial cognition: A review", in *Image and Environment*, ed. D. Stea, Aldine: Chicago, U.S.A..

Hasemer, A. (1984), *A Beginner's Guide to Lisp,* Addison-Wesley: Wokingham, U.K.

Hayes, P.J. (1977a), "In defense of Logic", *Proc. 5th IJCAI,* pp.559-565, Morgan Kaufmann: Los Altos, CA.

Hayes, P.J. (1977b), "On semantic nets, frames and associations", *Proc. 5th IJCAI,* pp.99-107, Morgan Kaufmann: Los Altos, CA.

Hayes, P.J. (1979), "The Logic of Frames", in *Frame Conceptions and Text Understanding*, ed. D. Metzing, Walter de Gruyter and Co: Berlin.

Hendrix, G.G. (1975), "Expanding the utility of semantic networks through partitioning", pp. 115-121 in *Proc. 4th IJCAI*, Morgan Kaufmann: Los Altos, CA.

Hilgard, E.R. (1981), "Imagery and Imagination in American Psychology", *Journal of Mental Imagery* **5**(1), pp.5-65.

Hintzman, D.L., O'Dell, C.S., and Arndt, D.R. (1981), "Orientation in Cognitive Maps", *Cognitive Psychology* **13**, pp.149-206.

Hofstadter, D.R. (1979), *Gödel, Escher and Bach: An Eternal Golden Braid,* Harvester Press: Chichester, U.K.

Hopgood, F.R.A., and Duce, D.A. (1980), "A Production System Approach to Interactive Graphic Program Design", in *Methodology of Interaction*, ed. R. A. Guedj, F.R.A. Hopgood, P.J.W. ten Hagen, H. Tucker and D.A. Duce, North-Holland: Amsterdam.

Horowitz, M.J. (1970), *Image Formation and Cognition,* Appleton-Century-Crofts: New York.

Hughes, G.E., and Cresswell, M.J. (1968), *An Introduction to Modal Logic,* Methuen: London.

Inhelder, B., and Piaget, J. (1958), *The Growth of Logical Thinking from Childhood to Adolescence,* Routledge and Kegan Paul: London.

Israel, D. (1983), "The Role of Logic in Knowledge Representation", *IEEE Computer* **16**(10), pp.37-42.

Jackman, M.K. (1987), "Inference and the Conceptual Graph Representation Language", in *Research and Development in Expert Systems IV*, ed. Moralee, S., Cambridge University Press: Cambridge, U.K.

Jackman, M.K. (1988), "The Maximal Join for Conceptual Graphs", in *Conceptual Graphs for Knowledge Systems*, ed. Sowa, J.F., Foo, N.Y. and Rao A.S., Addison-Wesley: Reading, MA.

Jackson, P. (1986), *Introduction to Expert Systems,* Addison-Wesley: London.

Jastrow, J. (1888), "The Dreams of the Blind", *The New Princeton Review* **5**, pp.18-34.

Johnson-Laird, P.N., and Steedman, M.J. (1978), "The psychology of syllogisms", *Cognitive Psychology* **10**, pp.64-99.

Johnson-Laird, P.N. (1983), *Mental Models,* Cambridge University Press: Cambridge.

Johnson-Laird, P.N., and Bara, B.G. (1984), "Syllogistic Inference", *Cognition* **16**, pp.1-61.

Johnson-Laird, P.N., Herrman, D.J., and Chaffin, R. (1984), "Only Connections: A Critique of Semantic Networks", *Psychological Bulletin* **96**(2), pp.292-315.

Kaczmarek, T.S. (1986), "Recent developments in NIKL", *Proc. AAAI-86*, pp.978-985, Morgan Kaufmann: Los Altos, CA.

Kahn, K., and Gorry, G.A. (1977), "Mechanizing Temporal Knowledge", *Artificial Intelligence* **9**, pp.87-108.

Kaisler, S.H. (1986), *INTERLISP: The Language and its Use,* Wiley: New York.

Kant, I. (1963, Originally published 1787), *Critique of Pure Reason 2nd Edition, N.K. Smith trans.,* Macmillan: London.

Kaufmann, G. (1979), *Visual Imagery and its Relation to Problem Solving: A Theoretical and Experimental Enquiry,* Universitetforlaget: Bergen, Oslo en Tromso.

Kieras, D.E., and Polson, P.G. (1985), "An approach to the formal analysis of user complexity", *International Journal of Man-Machine Studies* **22**, pp.365-394.

Klahr, D., Langley, P., and Neches, R.T. (1986), *Production System Models of Learning and Development,* MIT Press: Cambridge, Maryland.

Kolata, G. (1982), "How Can Computers Get Common Sense", *Science* **217**, pp.1237-1238.

Kosslyn, S.M., and Schwartz, S.P. (1978), "A simulation of visual imagery", *Cognitive Science* **1**, pp.265-295.

Kosslyn, S.M., Ball, T.M., and Reiser, B.J. (1978), "Visual images preserve metric spatial information: Evidence from studies of image scanning", *Journal of Experimental Psychology: Human Perception and Performance* **4**, pp.47-60.

Kosslyn, S.M., Pinker, S., Smith, S.E., and Schwartz, S.P. (1979), "On the demystification of mental imagery", *The Behavioural and Brain Sciences* **2**, pp.535-581.

Kosslyn, S.M. (1980), *Image and Mind,* Harvard University Press: Cambridge, MA.

Kosslyn, S.M. (1981), "The Medium and the Message in Mental Imagery: A Theory", *Psychological Review* **88**, pp.46-65.

Kraft, A. (1987), "Artificial Intelligence: Next Generation Solutions", in *Intelligent Knowledge-Based Systems: an Introduction*, ed. T. O'Shea, J. Self and G. Thomas, Harper and Row: London.

Kulikowski, C., and Weiss, S. (1971), "Computer-based models of glaucoma", Report CBM-TR-3, Deptartment of Computer Science, Rutgers University: New Brunswick, NJ.

Kunz, J., Fallat, R., McClung, D., Osborn, J., Votteri, B., Nii, H., Aikens, J., Fagan, L., and Fiegenbaum, E. (1978), "A Physiological Rule Based System for Interpreting Pulmonary Function Test Results", Working Paper Human Perception and Performance-78-19, Heuristic Programming Project, Dept. of Computer Science, Stanford University.

Ladkin, P. (1986a), "Primitives and Units for Time Specification", *Proc. AAAI'86* **1**, pp.354-359, Morgan Kaufmann: Los Altos, CA.

Ladkin, P. (1986b), "Time Representation: A Taxonomy of Interval Relations", *Proc. AAAI'86*, pp.360-366, Morgan Kaufmann: Los Altos, CA.

Larkin, J.H., and Simon, H.A. (1987), "Why a Diagram is (Sometimes) Worth Ten Thousand Words", *Cognitive Science* **11**, pp.65-99.

Laurent, J.-P., Ayel, J., Thome, F., and Ziebelin, D. (1986), "Comparative Evaluation of Three Expert System Development Tools: KEE, Knowledge Craft and ART", *Knowledge Engineering Review* **1**(4), pp.18-29.

Lehnert, W.G., Dyer, M.G., Johnson, P.N., Yang, C.J., and Harley, S. (1983), "BORIS - An Experiment in In-Depth Understanding of Narratives", *Artificial Intelligence* **20**, pp.15-62.

Lehr, T.F., and Wedig, R.G. (1987), "Towards a GaAs Realization of a Production-System Machine", *IEEE Computer* **20**(4), pp.37-48.

Lenat, D.B. (1982), "On automated scientific theory formation: A case study using the AM program", in *Knowledge-Based Systems in Artificial Intelligence*, ed. R. Davis and D.B. Lenat, McGraw-Hill: New York.

Levesque, H.J., and Mylopoulos, J. (1979), "A procedural semantics for semantic networks", in *Associative Networks: Representation and Use of Knowledge by Computers*, ed. N. V. Findler, Academic Press: New York.

Levesque, H.J. (1984), "Foundations of a functional approach to knowledge representation", *Artificial Intelligence* **23**(2), pp.155-212.

Levesque, H.J., and Brachman, R.J. (1985), "A Fundamental Tradeoff in Knowledge Representation and Reasoning", in *Readings in Knowledge Representation*, ed. R.J. Brachman and H.J. Levesque, Morgan Kaufmann: Los Altos, CA.

Levesque, H.J., and Brachman, R.J. (1987), "Expressiveness and Tractability in Knowledge Representation and Reasoning", *Computational Intelligence* **3**(2).

Lewis, C.I., and Langford, C.H. (1932), *Symbolic Logic,* Dover Publications.

Lewis, D.K. (1969), *Convention: A Philosophical Study.,* Harvard University Press: Cambridge, MA.

Lindsay, P.H., and Norman, D.A. (1977), *Human Information Processing: 2nd Edition,* Academic Press: New York.

Lippold, O.W.J. (1970), "Origin of the alpha rhythm", *Nature* **226**, pp.616-618.

Liskov, B., and Zilles, S. (1974), "Programming with abstract data types", *SIGPLAN Notices* **9**(4), pp.50-59.

Loftus, E.F. (1975), "Leading questions and the eyewitness report", *Cognitive Psychology* **7**, pp.560-572.

Loftus, E.F., and Loftus, G.R. (1980), "On the permanance of stored information in the human brain", *American Psychologist* **35**, pp.409-420.

Machtey, M., and Young, P. (1978), *An Introduction to the General Theory of Algorithms,* North-Holland: New York.

Maes, P. (1986), "Introspection in Knowledge Representation", *Proc. ECAI'86*, pp.256-269, Brighton, U.K.

Maier, N.R.F. (1931), "Reasoning in humans: II. The solution of a problem and its appearance in conciousness", *J. Comparative Psychol.* **12**, pp.181-194.

Mamor, G.S., and Zaback, L.A. (1976), "Mental rotation by the blind: does mental rotation depend on visual imagery?", *Journal of Experimental Psychology: Human Perception and Performance* **2**, pp.515-521.

Mani, K., and Johnson-Laird, P.N. (1982), "The mental representation of spatial descriptions", *Memory and Cognition* **10**(2), pp.181-187.

Marcus, M. (1980), *A Theory of Syntactic Recognition for Natural Language,* MIT Press: Cambridge, MA.

Markov, A.A. (1954), "Theory of Algorithms", National Academy of Sciences, Moscow, U.S.S.R..

McCarthy, J., and Hayes, P.J. (1969), "Some Philosophical Problems from the Standpoint of Artificial Intelligence", pp. 463-502 in *Machine Intelligence 4*, ed. B. Meltzer and D. Michie, Edinburgh University Press: Edinburgh.

McCarthy, J. (1980), "Circumscription - A Form of Non-monotonic Reasoning", *Artificial Intelligence* **13**(1, 2), pp.27-39.

McClelland, J.L., Rumelhart, D.E., and The PDP Research Group (1986), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition,* MIT Press: Cambridge, MA.

McDermott, D. (1982a), "A Temporal Logic for Reasoning about Processes and Plans", *Cognitive Science* **6**, pp.101-155.

McDermott, D. (1987), "A Critique of Pure Reason", *Journal of Computational Intelligence*, (In press).

McDermott, J. (1982b), "R1: A Rule-Based Configurer of Computer Systems", *Artificial Intelligence* **19**(1), pp.39-88.

McDermott, D., and Doyle, J. (1980), "Non-monotonic Logic I", *Artificial Intelligence* **13**(1, 2), pp.41-72.

McKim, R.H. (1980), *Experiences in Visual Thinking, 2nd Edition,* Brooks and Cole: Monterey, CA.

van Melle, W. (1981), *System Aids in Constructing Consultation Programs,* UMI Research Press: Ann Arbor, Michigan.

Metzler, J. (1973), "Cognitive analogues of the rotation of three-dimensional objects.", Unpublished Doctoral Dissertation, Stanford University.

Metzler, J., and Shepard, R.N. (1974), "Transformational studies of the internal representation of three-dimensional objects", in *Theories of Cognitive Psychology: The Loyola Symposium*, ed. R.L. Solso, Lawrence Erlbaum Associates,: Hillsdale, NJ.

Meyer, D.E., and Schvaneveldt, R.W. (1971), "Facilitation in recognising pairs of words: Evidence of a dependence between retrieval operations", *Journal of Experimental Psychology* **20**, pp.227-234.

Milgram, S., and Jodelet, D. (1976), "Psychological maps of Paris", in *Environmental Psychology*, ed. Revlin, L.G., Holt, Rinehart and Winston: New York.

Miller, G.A. (1956), "The magical number seven, plus or minus two: some limits on our capacity for processing information", *Psychological Review* **63**, pp.81-83.

Miller, G.A., Galanter, E., and Pribram, K.H. (1960), *Plans and the Structure of Behavior,* Holt: New York.

Miller, G.A., and Johnson-Laird, P.N. (1976), *Language and Perception,* Cambridge University Press: Cambridge.

Minsky, M. (1975), "A Framework for Representing Knowledge", pp. 211-277 in *The Psychology of Computer Vision*, ed. P.H. Winston, McGraw-Hill: New York.

Minsky, M. (1981), "A framework for representing knowledge", in *Mind Design*, ed. J. Haugeland, MIT Press: Cambridge, MA.

Moore, J., and Newell, A. (1973), "How Can MERLIN Understand", pp. 201-310 in *Knowledge and Cognition*, ed. L. Gregy, Lawrence Erlbaum Associates Hillsdale, NJ.

Moore, R.C. (1985a), "The Role of Logic in Knowledge Representation and Commonsense Reasoning", in *Readings in Knowledge Representation*, ed. R. J. Brachman and H. J. Levesque, Morgan Kaufmann: Los Altos, CA.

Moore, R.C. (1985b), "A Formal Theory of Knowledge and Action", in *Formal Theories of the Commonsense World*, Ablex Publishing Co.

Morton, J., Hammersley, R.H., and Bekerian, D.A. (1985), "Headed records: A model and its failures", *Cognition* **20**, pp.1-36.

Mylopoulos, J., Shibahara, T., and Tsotsos, J.K. (1983), "Building knowledge-based systems: the PSN experience", *IEEE Computer* **16**(10), pp.83-89.

Nash-Webber, B., Bobrow, D.G., and Collins, A. (1975), "The role of semantics in automatic speech understanding", pp. 351-382 in *Representation and Understanding: Studies in Cognitive Science*, Academic Press: New York.

Neely, J.H. (1976), "Semantic priming and retrieval from lexical memory: Evidence for facilitatory and inhibitory processes", *Memory and Cognition* 4(5), pp.648-654.

Newell, A. (1973), "Production Systems: Models of Control Structures", in *Visual Information Processing*, ed. W.G. Chase, Academic Press.

Newell, A. (1980), "The Knowledge Level", *AI Magazine* **2**(2).

Newell, A. (1982), "The Knowledge Level", *Artificial Intelligence* **18**, pp.87-127.

Newell, A., and H.A.Simon, (1972), *Human Problem Solving,* Prentice-Hall: Englewood Cliffs, NJ.

Nilsson, N.J., and Fikes, R.E. (1971), "STRIPS: A new approach to the application of theorem proving to problem solving", *Artificial Intelligence* **2**, pp.189-205.

Nilsson, N.J. (1982), *Principles of Artificial Intelligence,* Springer-Verlag: New York.

Norman, D.A., and Bobrow, D.J. (1976), "On the role of active memory processes in perception and cognition", in *The Structure of Human Memory*, ed. C.N. Cofer, Freeman: San Francisco.

Norman, D.A., and Bobrow, D.J. (1979), "Descriptions: a basis for memory acquisition and retrieval", *Cognitive Psychology* **11**, pp.107-123.

Norman, D.A., and Rumelhart, D.E. (Eds.) (1975), *Explorations in Cognition,* Freeman: San Fransisco.

O'Neill, J.J. (1980), *Prodigal Genius: The Life of Nikola Tesla,* Grafton: London.

Oakhill, J.V., Johnson-Laird, P.N., and Bull, D. (1986), "Children's Syllogistic Reasoning", *Quarterly Journal of Experimental Psychology* **38A**, pp.35-58.

Osherson, D.N., and Smith, E.E. (1981), "On the adequacy of prototype theory as a theory of concepts", *Cognition* **9**, pp.35-58.

Oswald, I. (1957), "The EEG, visual imagery and attention", *Quarterly Journal of Experimental Psychology* **9**, pp.113-118.

Owens, J., Bower, C.H., and Black, J.B. (1979), "The 'soap opera' effect in story recall", *Memory and Cognition* **7**, pp.185-191.

Paivio, A. (1971), *Imagery and Verbal Processes,* Holt, Rinehart and Winston: New York.

Parnas, D.L. (1985), "Software Aspects of Strategic Defense Systems", *Commun. ACM* **28**(12), pp.1326-1335.

Perky, C.W. (1910), "An experimental study of imagination", *Amer. J. Psychol.* **21**, pp.422-452.

Pinker, S. (1985), *Visual Cognition: Reprints from Cognition: International Journal of Cognitive Psychology Volume 18, 1984,* MIT - Bradford Books: Cambridge, MA.

Polit, S. (1985), "R1 and beyond: AI technology transfer at DEC", *AI Magazine* **6**(4), pp.76-78.

Pople, H.E. Jr., Myers, J.D., and Miller, R.A. (1975), "DIALOG : A model of diagnostic logic for internal medicine", *Proc. 4th IJCAI*, pp.848-855, Morgan Kaufmann: Los Altos, CA.

Post, E.L. (1943), "Formal reductions of the general combinatorial decision problem", *American J. Mathematics* **65**, pp.197-268.

Power, R. (1984), "Mutual Intention", *Journal for the Theory of Social Behaviour* **14**, pp.85-102.

Putnam, H. (1977), "Is Semantics Possible?", in *Naming, Necessity and Natural Kinds*, ed. S.P. Schwartz, Cornell University Press: Ithaca, NY.

Pylyshyn, Z.W. (1981), "The imagery debate: Analogue media versus tacit knowledge", *Psychological Review* **87**, pp.16-45.

Pylyshyn, Z.W. (1984), *Computation and Cognition: Toward a Foundation for Cognitive Science,* MIT Press: Cambridge, MA.

Quillian, M.R. (1966), "Semantic memory", Unpublished Ph.D. dissertation, Carnegie Institute of Technology: Pittsburg.

Quillian, M.R. (1968), "Semantic Memory", pp. 216-270 in *Semantic Information Processing*, ed. M. Minsky, MIT Press: Cambridge, MA.

Rao, A.S., and Foo, N.Y. (1987), "Congres: Conceptual graph reasoning system", *Proc. IEEE 87*, pp.87-92.

Reboh, R., and Risch, T. (1986), "SYNTEL(TM): knowledge programming using functional representations", *Proc. AAAI-86,* Morgan Kaufmann: Los Altos, CA.

Reed, S.K. (1974), "Structural descriptions and the limitations of visual images.", *Memory & Cognition* **2**, pp.329-336.

Reiter, R. (1978), "On Reasoning by Default", *Proceedings TINLAP-2*: University of Illinois at Urbana-Champaign.

Reiter, R. (1980), "A Logic for Default Reasoning", *Artificial Intelligence* **13**(1,2), pp.81-132.

Reiter, R. (1985), "On Reasoning By Default", in *Readings in Knowledge Representation*, ed. R. J. Brachman and H. J. Levesque, Morgan Kaufmann: Los Altos, CA.

Rescher, N., and Urquhart, A. (1971), *Temporal Logic*, Springer-Verlag.

Rich, C. (1982), "Knowledge representation languages and predicate calculus: how to have your cake and eat it too", *Proc. AAAI-82*, Morgan Kaufmann: Los Altos, CA.

Richardson, J.T.E. (1980), *Mental Imagery and Human Memory*, St. Martin's Press: New York.

Rips, L.J., Shoben, E.J., and Smith, E.E. (1973), "Semantic distance and the verification of semantic relations", *Journal of Verbal Learning and Verbal Behaviour* **12**, pp.1-20.

Rips, L.J. (1983), "Cognitive Processes in Propositional Reasoning", *Psychological Review* **90**(1), pp.38-71.

Rosch, E. (1976), "Classification of real world objects: origins and representation in cognition", in *La Mémoire Sémantique*, ed. E. Ehrlich and E. Tulving, Bulletin de psychologie: Paris.

Rosenbloom, P.S., Laird, J.L., McDermott, J., Newell, A., and Orciuch, E. (1985), "R1-Soar: An Experiment in Knowledge-Intensive Programming in a Problem-Solving Architecture", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **7**(5), pp.561-569.

Rosser, B. (1939), "An Informal Exposition of Gödel's Theorems and Church's Theorem", *The Journal of Symbolic Logic* **4**(2), pp.53-60.

Rumelhart, D.E., and Norman, D.A. (1973), "Active semantic networks as a model of human memory", pp. 450-457 in *Proc. 3rd IJCAI*, Morgan Kaufmann: Los Altos, CA.

Rumelhart, D.E., and Ortony, A. (1976), "The representation of knowledge in memory", CHIP Report 55, University of California: San Diego, CA.

Rumelhart, D.E., and McClelland, J.L. (1985), "Levels Indeed! A Response to Broadbent", *Journal of Experimental Psychology: General* **114**(2), pp.193-197.

Russell, B. (1945), *A History of Western Philosophy*, Simon and Schuster: New York.

Schaeffer, B., and Wallace, R. (1969), "Semantic similarity and the comparison of word meanings", *Journal of Experimental Psychology* **82**, pp.343-346.

Schank, R.C. (1972), "Conceptual Dependency: A theory of natural language understanding", *Cognitive Psychology* **3**, pp.552-631.

Schank, R.C., and Abelson, R.P. (1977), *Scripts, Plans, Goals and Understanding,* Lawrence Erlbaum Associates: Hillsdale, NJ.

Schank, R.C (1980), "Language and Memory", *Cognitive Science* **4**, pp.243-284.

Schiffer, S.S. (1972), *Meaning,* Clarendon Press: Oxford.

Schubert, L.K. (1976), "Extending the expressive power of semantic networks", *Artificial Intelligence* **7**(2), pp.163-198.

Schwartz, S.P. (1977), *Naming, Necessity, and Natural Kinds,* Cornell University Press: Ithaca, NY.

Schwartz, S.P. (1979), "Studies of mental image rotation: Implications for a computer simulation of visual imagery", Unpublished Doctoral Dissertation, Johns Hopkins University. (Mentioned in Kosslyn *et al.*, 1979)

Sergot, M.J., Sadri, F., Kowalski, R.A., Kriwaczek, F., Hammond, P., and Cory, H.T. (1986), "The British Nationality Act as a Logic Program", *Commun. ACM* **29**(5), pp.370-386.

Shanon, B. (1976), "Aristotelianism, Newtonianism and the physics of the layman", *Perception* **5**, pp.241-3.

Shapiro, S.C. (1971), "A net structure for semantic information storage, deduction and retrieval", pp. 512-523 in *Proc. 2nd IJCAI*, Morgan Kaufmann: Los Altos, CA.

Sheehan, P.W. (1972), *The Function and Nature of Imagery,* Academic Press: New York.

Shepard, R.N., and Metzler, J. (1971), "Mental rotation of three-dimensional objects", *Science* **171**, pp.701-3.

Shepard, R.N., and Feng, C. (1972), "A chronometric study of mental paper folding", *Cognitive Psychology* **3**, pp.228-243.

Shortliffe, E. (1976), *Computer Based Medical Consultation: MYCIN,* Elsevier, New York.

Simmons, R.F. (1973), "Semantic networks: Their computation and use for understanding English sentences", pp. 66-113 in *Computer Models of Thought and Language*, ed. K. M. Colby, Freeman: San Francisco, CA.

Simon, H.A. (1978), "On the forms of mental representation", in *Minnesota Studies in the Philosophy of Science. Vol. ix: Perception and Cognition: Issues in the Foundations of Psychology*, ed. W.C. Savage, University of Minnesota Press: Minneapolis.

Simon, H.A., and Newell, A. (1965), *Computer Augmentation of Human Reasoning,* Spartan Books: Washington, DC.

Sloman, A. (1979), "Epistemology and Artificial Intelligence", in *Expert Systems in the Micro-electronic Age*, ed. D. Michie, Edinburgh University Press: Edinburgh.

Sloman, A. (1985), "Why we need many knowledge representation formalisms", in *Research and Development in Expert Systems*, ed. Bramer, M., Cambridge University Press: Cambridge.

Smith, E.E., Shoben, E.J., and Rips, L.J. (1974), "Structure and process in semantic memory: A featural model for semantic decisions", *Psychological Review* **81**, pp.214-241.

Smith, B.C. (1982), "Reflections and Semantics in a Procedural Language", Technical Report MIT/LCS/TR-272, MIT: Cambridge, MA.

Soloway, E., Bachant, J., and Jensen, K. (1987), "Assessing the Maintainability of XCON-in-RIME: Coping with the Problems of a VERY Large Rule-Base", *Proc. AAAI-87*, Morgan Kaufmann: Los Altos, CA.

Somers, H.L., and Johnson, R.L. (1979), "PTOSYS: An interactive system for 'understanding' texts using a dynamic strategy for updating dictionary entries", pp. 85-103 in *The Analysis of Meaning: Informatics 5*, ed. M. MacCafferty and K. Gray, Aslib: London.

Sowa, J.F. (1984), *Conceptual Structures: Information Processing in Minds and Machines,* Addison-Wesley: Reading, MA.

Sowa, J.F. , and Foo, N.Y. (Eds.), *Conceptual Graphs for Knowledge Systems*, (To be published).

Sowa, J.F., and Way, E.C. (1986), "Implementing a Semantic Interpreter Using Conceptual Graphs", *IBM Journal of Research and Development* **30**(1), pp.57-69.

Sowizral, H.A., and J.R.Kipps, (1986), "ROSIE: A Programming Environment for Expert Systems", in *Expert Systems: Techniques Tools, and Applications*, ed. D.A.Waterman, Addison-Wesley.

Sparck-Jones, K., and Boguraev, B. (1987), "A Note on a Study of Cases", *Computational Linguistics* **13**(1-2), pp.65-68.

Stevens, A., and Coup, P. (1978), "Distortions in judged spatial relations", *Cognitive Psychology* **10**, pp.422-437.

Szolovits, P. (1983), "Toward More Perspicuous Expert System Organization", pp. 7-12 in *Report on Workshop on Automated Explanation Production, SIGART Newsletter*, ed. W. Swartout.

Thorndyke, P.W., and Hayes-Roth, B. (1978), "Spatial knowledge acquisition from maps and navigation", Paper presented to the Psychonomic Society Meeting: San Antonio, Texas, U.S.A.

Tulving, E. (1972), "Episodic and semantic memory", in *Organisation of Memory*, ed. W. Donaldson, Academic Press: New York.

Tulving, E. (1984), "Precis of Elements of episodic memory", *The Behavioural and Brain Sciences* **7**, pp.223-268.

Turner, R. (1984), *Logics for Artificial Intelligence,* Ellis Horwood: Chichester, U.K.

Walther, C. (1985), "A Mechanical Solution of Schubert's Steamroller by Many Sorted Resolution", *Artificial Intelligence* **26**(2), pp.217-224.

Warden, C.J. (1924), "The relative economy of various modes of attack in the mastery of a stylus maze", *Journal of Experimental Psychology* **7**, pp.243-275.

Wason, P.C., and Shapiro, D. (1971), "Natural and contrived experience in a reasoning problem", *Quarterly Journal of Experimental Psychology* **23**, pp.63-71.

Wason, P.C., and Johnson-Laird, P.J. (1972), *Psychology of Reasoning: Structure and Content,* Harvard University Press: Cambridge, MA.

Waterman, D.A., and Hayes-Roth, F. (Eds.) (1978), *Pattern-Directed Inference Systems,* Academic Press: New York.

Watson, J.B. (1913), "Psychology as a behaviorist views it", *Psychological Review* **20**, pp.158-177.

Wertheim, A.H. (1974), "Oculomotor control and occipital alpha activity: a review and a hypothesis", *Acta Psychologica* **38**, pp.235-256.

Wertheim, A.H. (1981), "Occipital alpha activity as a measure of retinal involvement in oculomotor control", *Psychophysiology* **18**, pp.432-439.

Wierzbicka, A. (1972), *Semantic Primitives,* Athenäum Verlag: Frankfurt.

Wilkins, A.J. (1971), "Conjoint frequency, category size, and categorisation time", *Journal of Verbal Learning and Verbal Behaviour* **10**, pp.382-385.

Williams, M.D. (1978), "The process of retrieval from very long term memory", Technical Report no. 75, Center for Human Information Processing: San Diego, CA.

Winston, P.H. (1975), "Learning Structural Descriptions from Examples", pp. 157-209 in *Psychology of Computer Vision,* ed. P.H. Winston, McGraw-Hill: New York.

Winston, P.H., and Horn, B.K.P. (1984), *LISP: 2nd Edition,* Addison-Wesley.

Wittgenstein, L. (1953), *Philosophical Investigations,* Blackwell: Oxford.

Woods, W.A. (1975), "What's in a link: foundations for semantic networks", pp. 35-82 in *Representation and Understanding: Studies in Cognitive Science,* ed. D.G. Bobrow and A.M. Collins, Academic Press: New York.

Woods, W.A. (1983), "What's important about knowledge representation", *IEEE Computer* **16**(10), pp.22-27.

Wundt, W. (1904), *Principles of Physiological Psychology (transl. E.B. Titchener),* Swan Sonneschein & Co.: London.

Young, R.M. (1976), *Seriation by Children: an Artificial Intelligence Analysis of a Piagetian Task,* Birkhauser Verlag: Basel.

Young, R.M., and O'Shea, T. (1982), "Errors in children's subtraction", *Cognitive Science* **5**, pp.153-77.

Young, R.M. (1987), "Introduction to Production Systems", in *Intelligent Knowledge-Based Systems: an Introduction*, ed. T. O'Shea, J. Self and G. Thomas, Harper and Row: London.

Yuille, J.C., and Steiger, J.H. (1982), "Non-holistic processing in mental rotation: some suggestive evidence", *Perception & Psychophysics* **31**, pp.201-209.

Yuille, J.C. (1983), *Imagery, Memory and Cognition: Essays in Honor of Allan Paivio,* Lawrence Erlbaum Associates: Hillsdale, NJ.

Zadeh, L.A. (1974), "Fuzzy Logic and its application to approximate reasoning", *Information Processing 1974,* pp.591-594, North-Holland: Amsterdam.

Index