Evidential Reasoning in Semantic Networks: A Formal Theory and its Parallel Implementation

Lokendra Shastri
Department of Computer Science
The University of Rochester
Rochester, NY 14627

TR 166
September 1985

This report reproduces a dissertation submitted in partial fulfillment of the requirements for the degree Doctor of Philosophy in Computer Science at the University of Rochester, supervised by Jerome A. Feldman.

This research was supported in part by the National Science Foundation under Grants MCS-8209971, IST-8208571, and DCR-8405720.

Curriculum Vitae

Lokendra Shastri was born on August 14, 1956, in Indore, India, He received his early education at Campion School Bhopal, and graduated from High School in 1971. Shortly thereafter, he moved with his family to Jammu and Kashmir, and studied at the Gandhi Memorial College, Jammu. There he passed the Pre-University examination of Jammu University in 1972, securing the first rank. The same year he left home to join the Birla Institute of Technology and Science at Pilani, to pursue a Bachelor's degree in Electronics Engineering. In 1977 he received the BE(hons) degree with distinction from the Birla Institute of Technology and Science- Next, he decided to pursue his studies in the field of Computer Science. After spending six months at the School of Automation, Indian Institute of Science, Bangalore, Lokendra transferred to the Indian Institute of Technology, Madras, where he received a Master's degree in Computer Science in February 1980. While awaiting admission to a graduate program, he worked for Hinditron Computer Systems as a Software Engineer. Subsequently, in January 1981, he joined the Computer Science department at the University of Rochester to pursue a doctoral degree. During his stay at Rochester he was a research assistant and a teaching assistant from 1981 to 1985. This provided him an opportunity to participate in and contribute to many significant projects. During the summer of 1981 he was a member of the natural language understanding project. The next summer he wrote ISCON, a simulator for connectionist networks. Since 1982, he has been working on knowledge representation and evidential reasoning within a massively parallel (connectionist) framework. He has published several papers on this subject in collaboration with Jerry Feldman. He has also published a paper on connectionist parsing with Steven Small and Garrison Cottrell. In July 1985, he joined the faculty of the Computer and Information Science department at the University of Pennsylvania.

Acknowledgements

rould like to thank my advisor, Jerry Feldman for guiding me during renticeship at Rochester, and exposing me to the wonderful world of connectioni Lve learned a great deal from him and this thesis bears testimony to that; but for dance, motivation and support this thesis would not have been possible-1 would *i* \ to thank him for creating an unencumbered and intellectually stimulat ironment at Rochester and being a constant source of inspiration.

Duld like to thank my committee members: James Allen, Dana Ballard, Gary I [Patrick Hayes, for lending me their ear and encouraging me these last two ye \y read numerous rough drafts and provided valuable feedback. I will 1 lember the heated arguments with Pat. He too taught me a lot and it was a pleas jracting with him. I am grateful to Dana for his criticism and advice. To a gi gnt he was responsible for pushing me into seeking a theory of evidence for para works. His criticisim paid rich dividends. James shared his insights into sema; works, offered many helpful, suggestions and asked many crucial questions. I D grateful to Henry Kyburg for his comments on my paper on evidential reason] 1 toSteven Small for helping me in numerous ways.

Duld like to thank Subhash Gupta for convincing me that I should pursue a cares earch at a time when I doubted my own abilities. He is responsible for arousing erest in artificial intelligence and has remained a constant source of inspiration as.

stay at Rochester was made pleasant by Mark Brucks, fellow connectionist G ttrell, Diane Litman, Jim Mayer with whom I shared an office for four ye :hard Newman-Wolfe (Nemo), Richard Pelvin, and M. Srinivas. Special thank D Hartman who always seemed to be around to lend a helping hand. Most of

ıres appearing in the thesis are his handiwork. Many thanks to A
ndyopadhyay and Giannis Aloimonos for their hospitality and material support.

und the staff at the Department of Computer Science exceptionally warm, frien
l helpful. Special thanks to Jill Orioli, Rose Peet, Peggy Meeker, Suzanne E
l Peggy Frantz.

n grateful to my parents for a happy childhood, an abundance of warm memor
l for inculcating in me the desire to excel.

ny thanks to my wife Sadhana for her loving care and understanding. I fin
possible to imagine how anyone could be more supportive and helpful than she
is thesis would have taken many more months to finish were it not for her help
rifice.

ally, thanks to Anjali, who is too young to realize that she was also a great sourc
tivation and promises to be an even greater source of joy.

ABSTRACT

The problem of representing and utilizing a large body of knowledge is fundamental to artificial intelligence. This thesis focuses on two important issues related to this problem,

1. An agent cannot maintain complete knowledge about any but the most trivial environment, and therefore, he must be capable of reasoning with incomplete and uncertain information.

2. An agent must act in real-time. Human agents take a few hundred milliseconds to perform a broad range of intelligent tasks, and agents endowed with artificial intelligence should perform similar tasks in comparable time.

It is argued that the best way to cope with partial and incomplete information is to adopt an evidential form of reasoning, wherein, inference does not involve establishing the truth of a proposition but instead, it involves finding the most likely hypothesis from among a set of alternatives.

It is also argued that in order to satsify the real-time constraint, we must identify the kinds of inference that need to be performed very fast, and provide a *computational account* of how this limited class of inference may be performed in an acceptable time frame. This latter requirement prompts us to consider massively parallel models of computation, in particular models that do not require an interpreter.

Inheritance and categorization within a conceptual hierarchy are identified as two operations that humans perform very fast. It is suggested that these operations are important because they seem to lie at the core of intelligent behavior and are precursors to more complex reasoning.

The above concerns and proposed solutions lead to an evidential

framework for representing conceptual knowledge, wherein the principle of maximum entropy is applied to deal with uncertainty and incompleteness. It is demonstrated that the proposed framework offers a uniform treatment of inheritance and categorization, and solves an interesting class of inheritance and categorization problems, including those that involve exceptions, multiple hierarchies, and conflicting information. The proposed framework can be encoded as an interpreter-free, massively parallel (connectionist) network, that can solve the inheritance and categorization problems in time proportional to the depth of the conceptual hierarchy.

TABLE OF CONTENTS

"Evidential reasoning in semantic networks: A formal theory and its parallel implementation"

Errata sheet

"entropuy" should be "entropy"

1

"Dempster-Dempster" should be "Dempster-Shafer"

2

"imheritance" should be "inheritance"

"aproaches" should be "approaches"

4 from bottom

"inference" should be "inferences"

12,

"quintessense" should be "quintessence"

tom of page:

"This shortcoming is avoided by Allen and Frisch" should read "This aspect of semantic network's is taken into account by Allen and Frisch".

: 1,

"Quakers are" should be "Quakers tend to be"

: 1 last paragraph,

"Mimic" should be "mimic".

line,

"1.4.2" should be "1.5.2"

line,

"appears section" should be "appears in section".

e 12,

"comparison the" should be "comparison of the"

e 1,

APL(C) should be "$\lambda(C)$"

e 19,

"(sat "netA")" should be "(say "netA")"

e 5,

"multiple hierarchies" should be "multiple views"

e 1 and 2, should read:

Potential: potential = the product of inputs at sites **QUERY RELAY** and **PV** divided by the inputs at site **INV**.

e 1,

"COND-2" should be "COND-2a"

ve line 6 (i.e. after COND-2a) insert the following text:

COND-2b: For any $P_1, P_2 \in \Phi$ if $V_1 \in \Lambda(P_1)$ and $V_2 \in \Lambda(P_2)$, then $V_1 \neq V_2$. In other words, distinct properties have distinct values.

: 14 and 15

same correction as one suggested for p. 155 line 1 and 2.

e 20

"AN" should be "An"

LIST OF FIGURES

Chapter 1

Introduction

1.1 The thesis

The problem of representing and utilizing a large body of knowledge is fundamental to artificial intelligence. It is well recognized that intelligent activity of any significance, whether it be natural language understanding, vision, problem solving, or planning, requires access to a large storehouse of knowledge at all levels of processing. Artificial intelligence has a rich tradition of work in knowledge representation and the related problem of inference.

Given the vast scope and the central nature of the knowledge representation problem, there is no dearth of issues that need to be addressed or problems that remain to be solved. This thesis focuses on two of the issues that I consider to be crucial.

These issues are:

1. The necessity of identifying and formalizing inference structures that are appropriate for dealing with incompleteness and uncertainty. An agent cannot maintain complete knowledge about any but the most trivial environments, and therefore, he must be capable of reasoning with incomplete and uncertain information.

2. The importance of computational tractability. An agent must act in real-time. Human agents take a few hundred milliseconds to perform a broad range of intelligent tasks, and we should expect agents endowed with artificial intelligence to perform similar tasks in comparable time.

It is argued that the best way to cope with partial and incomplete information is to adopt an evidential form of reasoning, wherein, inference does not involve establishing the truth of a proposition but instead, it involves finding the most likely hypothesis from among a finite set of alternatives.

It is also argued that in order to satsify the real-time constraint, we must identify the kinds of inference that need to be performed very fast, and provide a *computational account* of how this limited class of inference may be performed in an acceptable time frame. This latter requirement prompts us to consider massively parallel models of computation, in particular models that do not require an interpreter.

Inheritance and categorization within a conceptual hierarchy are identified as two operations that humans perform very fast. It is suggested that these operations are important because they seem to lie at the core of intelligent behavior and are precursors to more complex reasoning.

The above concerns and proposed solutions lead to an evidential framework for representing conceptual knowledge, wherein the principle of maximum entropy is applied to deal with uncertainty and incompleteness. The proposed framework offers a uniform treatment of inheritance and categorization, and solves an interesting class of inheritance and categorization problems, including those that involve exceptions, multiple hierarchies, and conflicting information. In particular, the proposed treatment of inheritance offers distinct advantages over existing solutions based on default logic and inferential distance ordering. The evidence combination rule employed in this thesis is demonstrably better than the Dempster-Shafer evidence combination rule in the context of the problems addressed in this thesis. Furthermore, it is established that the framework can be encoded as an interpreter-free, massively parallel (connectionist) network, that can solve the inheritance and categorization problems in time proportional to the depth of the conceptual hierarchy.

1.2 Reasoning with incomplete and uncertain information:

The computational cost of gathering, processing and storing information about a complex and constantly changing environment makes it impossible for an agent to maintain complete knowledge. However, an agent must act and make choices on the basis of available information, and this compels him to make inferences based on incomplete knowledge. With the exception of some formal and artificially constructed domains, an agent seldom has sufficient information to pin down the exact state of the world. Consequently, a straight forward application of deductive reasoning is not very useful in making decisions or choices. Consider a situation in which an agent *must choose* between actions A and B, where action A is appropriate if hypothesis P is true while action B is appropriate if hypothesis Q is true. Assume that the agent's knowledge is incomplete and the best that he can do is deduce "P V Q". What should an intelligent agent do in such a situation? Should he give up in despair and make an arbitrary choice, or is there any other option open to him? An excellent strategy would be to make the best use of all available information and try and identify which of the two hypothesis is *more likely* to be true. If P is more likely than Q then he should perform action A else he should perform action B. This form of reasoning is best viewed as evidential reasoning, wherein inference does not involve establishing the truth of a proposition, but instead it involves finding the most likely hypothesis from among a finite set of alternatives. Informally, evidential reasoning justifies inference, not on the grounds that the selected hypothesis is true in *all* possible worlds that are consistent with the agent's knowledge, but rather on the grounds that there are more possible worlds consistent with the agent's knowledge in which the selected hypothesis is true, than there are those in which one of the alternative hypothesis is true[1].

1. This interpretation does not apply to all theories of evidential reasoning. However, this is the interpretation adopted in this work.

Although the need for formalizing inference structures that deal with incomplete and uncertain knowledge is well recognized in knowledge representation circles [Doyle 83][Fox 81][Halpern & McAUester 84][[Joshi 78][Levesque 82][McCarthy 84][McDermott & Doyle 80][Moore 83][Nilsson 84][Reiter 80], only a few of these attempt to adopt an evidential approach [Haplern & McAllester][Joshi 78][Nilsson 84].

1.3 Computational tractability: limited inference and parallelism

The issue of computational tractability is of utmost importance to artificial intelligence and it should not be dismissed as a mere concern for the efficiency of implementation. Computational tractability is not about efficiency, programming tricks, or faster machines; the issue is more profound. The crucial question that it raises is: "does there exist a *computational* account *(any* account) of how an agent may draw the relevant inferences in the time scales he is permitted by the environment"? For if we desire to build an intelligent agent (robots or what have you), then we must realize that such an agent will have to act in real-time, and often this may mean that it will have to perform non-trivial inferences in a very short time. Human agents take but a few hundred milliseconds to perform a broad range of tasks such as visual recognition, categorization, and property inheritance, and it should not be unreasonable to expect agents endowed with artificial intelligence to perform similar tasks in comparable time.

Given that unrestricted inference is undecidable, a concern for tractability should prompt us to consider limited inference [Frisch & Allen 82] [Levesque 84]. One must identify the kinds of inference that an agent needs to perform very fast, and provide a *computational account* of how these may be performed within an acceptable time frame. It is possible that one may succeed in identifying such a class of restricted inference and also ensure that at least these inferences are performed with great efficiency.

In view of the above, the knowledge base and reasoning subsystem of an agent may consist of two components. One of the components, the knowledge base (KB), may contain a highly structured collection of the agent's knowledge together with a reasoner that is capable of performing a *limited* class of inference with *extreme efficiency*. The other component may be a collection of several domain dependent modules (domain experts), each of which may be capable of elaborate reasoning but contain a lot of domain specific meta-information (or control information) to help it perform its inferences efficiently. The operation of the overall subsystem may proceed as follows: The actual task dependent reasoning is performed by the various domain experts. Each expert may query the KB for information and use its own reasoner to draw requisite inferences based on the information provided by the KB. While performing these inferences, the domain expert's reasoner would make full use of the control information available locally, and hence perform the inferences efficiently. In the limiting case, a domain expert will have enough control information to make its reasoner functionally equivalent to a set of compiled routines, each of which is dedicated to a specific domain task. This approach to a knowledge base and reasoning subsystem is similar in spirit to the *functional view* proposed in [Brachman et al. 83].

The extremely tight constraint on the time available to an agent to perform non-trivial inferences entails that we need to look beyond limited inference in our quest for computational tractability. Many of the perceptual and cognitive tasks that need to be performed in a few hundred milliseconds would require millions of instructions on a serial computer, and it is fairly obvious that at some point we must resort to even *massive* parallelism. For this reason one must consider massively parallel models of computation, in particular, models that do not require an interpreter. It is my contention that one can exploit the full power of parallelism only if it is taken as an important premise, and used to direct one's search for interesting solutions in the space of possible knowledge representation frameworks; it will not suffice to find serial solutions and then look for possible uses of parallelism in implementing these solutions. Section 2 discusses this issue in greater detail.

1.4 From issues to problems

The specific problems this thesis addresses are those of inheritance and categorization in semantic networks. This choice is well motivated because a broad range of reasoning tasks that human agents perform effortlessly and extremely fast may be viewed as examples of inheritance and categorization, and furthermore - as is argued in section 1.6, inheritance as well as categorization involve reasoning with incomplete knowledge.

Inheritance involves inferring properties about an individual or class, based on our knowledge of the properties of a more general class. Consider the kind of reasoning that goes on when we act as though we know that "Tweety flies" when what we actually know is: "Tweety is a bird" and "birds fly". This kind of reasoning, often referred to as default reasoning, is commonplace - one may even argue that it is the quintessense of common sense reasoning - and may be characterized as inheritance. If artificial intelligence models of language processing are to be believed, then inheritance plays a crucial role in natural language processing [Findler 79][Allen & Frisch 82] (also see [Cottrell 85]).

Each one of us is constantly classifying and categorizing our experiences. Our survival in a complex environment depends on our ability to organize our knowledge in terms of a manageable number of categories. A perfect example of categorization is visual recognition; it would not be inappropriate to claim that at higher levels of processing, visual recognition is based on matching a collection of features detected in an image to some internal representation of the visual characteristics of objects (or classes of objects).

Inheritance and categorization may be defined with respect to complex entities besides physical objects and their properties. One could have hierarchically organized information about events and relations and apply the same form of reasoning. For example, inheritance may be generalized to include reasoning about parts and temporal intervals [Allen 83][Fahlman 79][Schubert et al 83], (also see section 3.3).

It is my belief that the operations of categorization and inheritance are among the fundamental ingredients of intelligent behavior. They lie at the core of and act as precursors to more elaborate forms of reasoning, and provide the raw material for more complex and specialized reasoning processes. With reference to the architecture of a knowledge base subsystem alluded to in section 1.2, the structure of the central knowledge base is that of a semantic network. Domain experts constantly query the semantic network and the questions they pose amount to inheritance and categorization requests.

In the following sections I demonstrate that the reasoning underlying inheritance and categorization is based on incomplete and uncertain information, and is best described as evidential reasoning.

1.5 Semantic networks

Since their introduction by Quillian [Quillian 68], semantic networks have played a significant role in knowledge representation research. Semantic networks are representation formalisms that express knowledge in terms of concepts and their properties. A salient feature of semantic networks is that they highlight the hierarchical (subsumption) relationships between concepts. Each concept is represented by a node and its hierarchical relationship is depicted by connecting appropriate concept nodes via IS-A links. Nodes at the lowest level of the hierarchy denote individuals[2] (Tokens) while nodes at higher levels denote classes of individuals (Types). As one moves up the IS-A links one encounters more and more abstract concepts. Properties are attached to concept nodes via suitably labelled links. A property is attached at the highest possible level of abstraction in the conceptual hierarchy. Thus, a property that is true of mammals is attached to the node MAMMAL instead of being attached to nodes such as DOG, WHALE, ... Consequently, if a property is attached to a node C then it is assumed to be applicable to all nodes that are descendants of C.

2. Alternately, the lowest level may denote "phases" of individuals, thus one may have nodes such as "Fido when he was a puppy" and "Fido when he was two years old".

Inheritance is the form of reasoning that permits an agent to infer ("by inheritance") properties of a concept based on the properties of its ancestors. For example, if the node BIRD has the property "FLIES" attached to it, and the node TWEETY is connected to BIRD via an IS-A link, and we do not know whether Tweety flies or not, then using inheritance one may infer that Tweety flies.

Categorization is the dual of the inheritance problem. Unlike inheritance which seeks a property value of a given concept, categorization seeks a concept that has some specified property values. The categorization problem may be described as follows: "Given a description consisting of a set of properties, find a concept that best matches this description. This problem is seldom analysed in artificial intelligence work related to semantic networks although it is the central problem in perception (a notable exception is NETL [Fahlman 79]).

In order to understand and formalize inheritance and categorization one must begin by questioning the nature of knowledge represented in semantic networks. In other words, one must pin down the meaning of Types and the properties associated with Types. These questions have been raised in the past [Woods 75], [Allen & Frish 82], and several answers have been provided. In the following section I discuss some attempts that have been made at answering the above question and point out their shortcomings in modelling natural concepts.

1.5.1 Semantic networks are notational variants of first-order logic.

Attempts to explain semantic networks as notational variants of first-order predicate calculus (FOPC), date back to Cercone & Schubert [Cercone & Schubert 75]. They claimed that semantic networks were simply a graphical notation for predicate calculus; - they even developed an elaborate graphical notation to represent ***all*** of FOPC. A formalization of the supposed content of a semantic network appears in [Hayes 79] and [Charniak 81]. The translation involves mapping Tokens to constants, Types to unary predicates, and properties to either binary or unary predicates. Thus, an IS-A link between TWEETY (Token) and BIRD (Type) is expressed as:

BIRD(TWEETY)

the IS-A link between BIRD and LIVING-THING (Type) is expressed as

∀x BIRD(x) => LIVING-THING(x)

If a property is mapped to a unary predicate then the property of a concept may be expressed as:

∀(x) CANARY(x) => YELLOW(x)

Alternately, properties may be treated as two place predicates in which case the above information may be expressed as:

∀(x) CANARY(x) => HAS-COLOR(x, YELLOW)

In the translation described above, inheritance amounts to one or more applications of modus ponens. For example, starting with CANARY(TWEETY) one may deduce YELLOW(TWEETY) or alternately, HAS-COLOR(TWEETY, YELLOW), by a simple application of modus ponens. However, if a property were attached several levels above in the conceptual hierarchy then inheritance would require repeated applications of modus ponens.

Despite its elegance and simplicity, the above translation has three shortcomings.

1. The above translation does not explain how the information encoded in a semantic network could be used to solve categorization problems.

2. The control aspect of IS-A links is lost in the proposed translation to FOPC for the translation treats IS-A links as if they were any other implication in the knowledge base.

3. It overlooks two crucial aspects of world knowledge namely, the presence of exceptions and multiple hierarchies.

The translation to FOPC overlooks a fundamental dimension along which knowledge representation formalisms ought to be compared. A knowledge representation framework should not merely prescribe how small units of information ought to be represented, but it should also detail how the totality of information ought to be structured and organized so that appropriate information may be located with ease and relevant inferences may be performed efficiently. In other words, two knowledge representation formalisms are equivalent only if they contain the same information about the world, as well as the same control information about how to make inferences using this information.

The above translation to FOPC only encodes the component of information in a semantic network, which is about the world being modelled. However, semantic networks not only encode information about the world, but they also encode knowledge that may be used by an interpreter to perform certain inferences efficiently. Semantic networks make certain commitments about which inferences are important, and provide mechanisms for facilitating these inferences. For example, if we view semantic networks as graphs then inheritance reduces to a simple graph traversal problem. This shortcoming is avoided by Allen and Frisch [Allen & Frisch 82]. They introduce special predicates: TYPE and SUBTYPE together with special axioms that describe the transitive nature of IS-A links, in order to capture the special role played by these links during inheritance.

However, the most fundamental problem with the proposed translation into FOPC is the manner in which information about properties is translated. Either of the two translations involve the use of the universal quantifier. Thus, if one wanted to say "Birds fly", a translation into FOPC may result in:

VXBIRD(X) =>FLIES(x)

or else one may say:

Vx BIRD(X) => has-mode-of-Uansportation(x, FLYING)

Such a characterization of properties is confounded by two aspects of world knowledge:

Exceptions: There is a preponderance of situations where one may want to associate a property with a class, although - strictly speaking, the property may not hold for *all* members of the class. This leads to the notion of exceptions. To use a classic example, one might consider it natural to associate the property of being able to fly with birds, knowing fully well that *all* birds do not fly.

Multiple hierarchies: Often there exist several alternate but equally useful hierarchical organizations of concepts in a domain.[3] Consequently, it becomes more natural to organize concepts in the form of multiple hierarchies, wherein a concept may have more than one ancestors.

3. It is possible to map such "multiple views" into a taxonomy. However, this can only be done by introducing artificial concepts that correspond to "cross products" of concepts in individual hierarchies. For instance, we may want to classify objects in two ways: one by their shape, and the other by their color. Collapsing these two classifications to yield a single taxonomy will force us to introduce concepts corresponding to all possible pairs of shape and color values.

The presence of exceptions and multiple hierarchies in semantic networks introduces non-monotonicity and ambiguity, neither of which can be handled within the theories of first order logic that were suggested to correspond to semantic networks.

To illustrate that exceptions lead to non-monotonicity, consider the following example:

Assume that the agent knows that "Birds fly", "Penguins do not fly", "Penguins are birds". If he is told that "Tweety is a bird", he will infer that "Tweety flies". However, if in addition to the above, he were told that "Tweety is a penguin", then we would expect him to retract his earlier inference "Tweety flies". This would give rise to non-monotonicity.

The problem of ambiguity arises when the agent may hold seemingly conflicting beliefs, as is illustrated with the help of a familiar example (henceforth the quaker example):

The agent may simultaneously believe that:

Quakers tend to be pacifists. - SI -
Republicans tend to be non-pacifists. - S2 -
Dick is a quaker. - S3 -
Dick is a republican. - S4 -

A translation into FOPC leads to inconsistency and renders the resulting theory unusable. In this example, Dick has two parents in the conceptual hierarchy, one (Quaker) arose from classifying persons in terms of their religious beliefs, and the other (Republican) arose from classifying persons on the basis of their political beliefs.

1.5.2 Semantic networks are equivalent to suitable theories of default logic.

Recognizing the inability of simple theories of first order logic to formalize semantic networks with multiple hierarchies and exceptions, Etherington and Reiter have proposed an alternate formalization based on default logic [Reiter 80]. Etherington and Reiter [Etherington & Reiter 83] have shown that default theories that admit semi-normal defaults can handle inheritance in semantic networks with exceptions.

The Etherington and Reiter proposal does handle the problem of exceptions[4]. However, its treatment of multiple inheritance (inheritance from multiple unrelated ancestors), seems to overlook an important characteristic of real world knowledge. We substantiate this with the help of the quaker example.

4. I have some basic reservations about this approach which are discussed in section 5.4.

Let us examine how the proposed system would draw conclusions about Dick's pacifism - or the lack of it, on the basis of the agent's world knowledge given in SI through S4 (cf. the quaker example). In default logic, SI through S4 would be represented as shown below:

QUAKER(X) : PACIFIST(X) — dr-1

PACIFIST(x)

REPUBLICAN(X) : NON-PACIFIST(X) — dr-2

NON-PACIFIST(x)

QUAKER(DICK) —- A-l

REPUBLICAN(DICK) —- A-2

SI and S2 map to default rules dr-1 and dr-2 respectively, while S3 and S4 translate into simple first-order assertions given by Al and A2. In each default rule, the formula to the left of the colon is called the *prerequisite* of the default, the formula to the right of the colon is called the *justification*, while the formula in the "denominator" is called the *consequent*. A default rule has the following interpretation: if the prerequisite is known to be true, and if the justification is consistent with what is assumed, then one may infer the consequent.

The knowledge encoded in dr-1, dr-2, A1 and A2, leads to two *extensions*. One of the extensions includes PACIFIST(DICK), while the other includes NON-PACIFIST(DICK). Default logic prescribes that *"any one* of these extensions may be interpreted as an acceptable set of beliefs about the world", and hence, a system like the one outlined in [Etherington & Reiter 83] would arbitrarily choose between one of these extensions and respond with an answer that lies in the chosen extension. The choice of extension would depend on which of the two default rules - "Quakers tend to be pacifists" and "Republicans tend to be non-pacifists", is selected first by the inference algorithm. If the default rule that encodes "Republicans tend to be non-pacifists" happens to be selected first, the system would infer that "Dick is a non-pacifist". Once this inference is made, the rule encoding "Quakers tend to be pacifists" would no longer be justifiable with reference to Dick and hence, would not play *any role* in drawing conclusions about Dick. If instead, the default rule "Quakers tend to be pacifists" happens to be selected first, the system would infer that "Dick is a pacifist" and the rule "Republicans tend to be non-pacifists" would be rendered inapplicable and play no further role in the inference process.

One may make two observations on the basis of the above discussion.

1. In deciding whether Dick is a pacifist or a non-pacifist, *only one* of the two default rules is used. In general, once a rule is selected and applied, it precludes the use of other default rules that conflict with the inferences made by the former. Thus, if the system happens to make use of the knowledge that "Quakers tend to be pacifists", it has to *essentially ignore* the world knowledge that "Republicans tend to be non-pacifists".

2. The conclusions drawn by the system depend on a non-deterministic choice of the order of rule application. Hence, the conclusions drawn by the system are not only dictated by the *inference structure* but also depend on the *process structure* that implements some ad-hoc *strategy* for choosing rules. (The italicized terms are used in the sense of [Hayes 77]).

Our intuitions about the knowledge in the quaker example suggest that in drawing conclusions about Dick, both the statements - "Quakers tend to be pacifists", and "Republicans tend to be non-pacifists", are *relevant* and hence, both must affect the final conclusion. In general, the *final conclusion should reflect the combined effect of all the relevant information*. However, because default logic makes the implicit assumption that all default rules have the same "significance" or "import", it follows that if two or more rules have conflicting consequences then either the use of one rule should preclude the use of the other rules (as was the case in the quaker example), or no conclusions should be drawn based on these rules (see [Reiter & Crisculo 81]).

The nature of knowledge encoded by default rules suggests that all rules need not have the same import (even those having the same inferential distance). For instance, an agent may believe that the tendency of Quakers to be pacifists outweighs the tendency of Republicans to be non-pacifists and an *epistemologically adequate formalism should be capable of expressing such differences.*

A way to formalize these distinctions is to treat statements such as S1 and S2 as *evidential assertions* and to associate a numeric quantity with each assertion to indicate its evidential import. If one could assign meaning to these numbers and explain how these may be extracted from world knowledge, and also specify a formal calculus for computing the combined effect of multiple (conflicting as well as reinforcing) evidential assertions then one would be able to handle situations such as the quaker example more satisfactorily. For instance, statements such as "Quakers are pacifists" may be interpreted to mean - "the fact that 'x is a Quaker' lends evidence α to the fact that 'x is a pacifist'". Similarly, "Republicans tend to be non-pacifists" may be interpreted to mean "the fact that 'x is a Republican' lends evidence β to the fact that 'x is a non-pacifist'". In the evidential version of the quaker example, there will be evidential support α for "Dick is a pacifist" (because Dick is a Quaker and Quakers tend to be pacifists), while there will be evidential support β for "Dick is a non-pacifist" (because Dick is a Republican and Republicans tend to be non-pacifists). *Deciding whether Dick is a pacifist or a non-pacifist need not be based on arbitrary choices but instead, be resolved by a formally specified theory of evidential reasoning.*

It may be argued that in order to handle interactions between default rules it may not be necessary to introduce numbers. For example, given n default rules one could enumerate all the possible cases of interactions (about n! combinations), and specify the correct inferences that need be drawn in each case. For instance, in the quaker example, if "Quakers tend to be pacifists" is a stronger default rule than "Republicans tend to be non-pacifists", then all one needs to do is posit the following default rules:

"If someone is a Quaker and a Republican then he is probably a pacifist"

"If someone is a Quaker and not a Republican then he is probably a pacifist"

"If someone is a Republican but not a Quaker then he is probably a non-pacifist".

These rules could mimick the effect of associating numbers with the original pair of simple default rules. However, the argument against this suggestion is that having a formal calculus for *computing* the effects of interactions between default rules *in a justifiable manner* is far more desirable than having to explicitly list the outcome of every possible interaction.

Furthermore, knowledge acquisition would be problematic in a scheme that handles interactions between defaults by explicit enumeration of cases because adding new knowledge may require extensive modifications of existing default rules.

In the quaker example, I have deliberately used the construct "tend to be" instead of "are". Thus, S1 reads "Quakers tend to be pacifists" and not "Quakers are pacifists" or "All Quakers are pacifists". However, none of the arguments offered above would have been less relevant had I used "are" instead of "tend to be". In everyday situations when we say "Quakers are pacifists" or even "All quakers are pacifists" we seldom mean that - "ALL quakers are pacifists- period". What we actually mean is more akin to "most Quakers are pacifists, of course there are exceptions, but generally it is the case ....". The normal usage of language often belies the complexity of the information being communicated. Details and qualifications are often left unarticulated, because the speaker relies upon the hearer to fill in the necessary qualification by utilizing his knowledge of linguistic (and cultural) conventions.

1.5.3 Touretzky's inferential distance ordering

Touretzky [Touretzky 84] has suggested a formal account of inheritance in semantic networks. His thesis may be summarized by quoting what he calls the principle of *inferential distance ordering*. The principle states that: if A inherits P from B, and ~P from C, then "if A has an inheritance path via B to C and not vice versa, then conclude P; if A has an inheritance path via C to B and not vice versa, then conclude ~P; otherwise report an ambiguity". (From [Touretzky 84] p. 204). In the quaker example, Touretzky's system would report an ambiguity, for the principle of inferential distance ordering does not solve the problem of combining information from disparate sources. The principle induces only a very specific kind of distinction between default rules. This distinction primarily addresses the issue of exceptions and indeed, using this ordering Touretzky's system is able to handle exceptions[5]. However, his formalism treats all rules at the same inferential distance as having the same import and this forces him to report ambiguity in multiple inheritance situations such as the quaker example.

5. I have some basic reservations about the manner in which Touretzky handles exceptions, these are discussed in section 5.4

1.6 Evidential nature of knowledge in semantic networks

It was pointed out in section 1.4.1 that a simple translation of the relationship between concepts and their properties into a universally quanitified implication does not suffice because very often we associate a property with a Type, even though the property may not hold for all members of the Type. Similarly, many of the properties that are used to categorize instances of a Type are not unequivocally indicative of the Type. In other words, it is extremely difficult - if not impossible, to determine a set of necessary and sufficient properties that may be associated with a Type.

It is worth pointing out that the above observations are being made with respect to a "working definition" of a Type, i.e. descriptions of Types we use in our day to day interactions; information we use to make inferences of the sort: "Tweety flies because Tweety is a bird" even though we know that all birds do not fly, or information we use to recognize a person walking down the hallway as being a friend, although there is a distinct possibility that the person is just a look alike.

It may be possible to pin down the ideal definition of a Type. For example, one could state that the definition of "gold" is "the element whose atomic number is 79". Although it seems possible to find definitions for extremely elementary concepts such as gold, the situation becomes tricky when one tries to write down precise definitions of concepts such as "furniture" - or to mention a classic example, "game" [Wittgenstein 53]. I do not intend to get involved in a philosophical debate, but the following observation is pertinent. The identification of precise definitions of concepts is not critical to our ability to cope with our environment, and hence is not of much interest to me in the context of this thesis. For when we interact with our environment, *the kind of information available to us may not be the kind that is required to evaluate the precise definitions of a category*; nature has not endowed us with an "atomic number finder", but with a visual system that can "see yellow and detect lustre". For a discussion of categories (Types), the related psychological findings, and references to relevant philosophical literature, see [Smith & Medin 81].

It is argued in section 3.4, that Types evolve in order to enable an agent to cope with a complex environment. Given his limited resources, the best strategy open to an agent is to make generalizations that help him in breaking up the world into a manageable number of categories. By doing this he is able to categorize novel experiences and make predictions about these, based on past generalizations. It is certain that some of the categorizations and the consequent predictions that the agent will make will turn out to be wrong, but this is a small price to pay for being most often right, when the alternative is inaction.

What has been said above about Types should not be new to researchers in knowledge representation, but it has been repeated because in spite of it being obvious, when the time comes to formalize the notion of Types, this basic characteristic of Types is overlooked, simply because it is inconvenient to deal with. This state of affairs is a very good example of a theory taking precedence over the phenomena it is purported to model.

The stand taken in this thesis is that the notion of Types - specially as it corresponds to natural kind terms, is best modelled within an evidential framework, the basic idea of which has already been introduced in section 1.4.2. Within an evidential framework, beliefs such as "Apples are red" are interpreted to mean that if ^{M}x is an apple" then there is some evidence a that ^{M}x is red in color", and also that "if x is red in color" then there is some evidence *p* that "x is an apple". In the general case, if we are told that an object is red in color then we would have varying degrees of evidence that it is an apple or a tomato or a rose. . . . Similarly, if we are told that an object is an apple, then we would have varying degrees of evidence that its color could be red, green or yellow.

If all the apples we had ever seen, read, or heard about, were red in color, then "x is a apple" would have provided evidence *only* to "x is red in color". (This situation would be equivalent to Vx APPLE(X) => RED(x), but notice that there is no requirement that the evidence provided to RED would equal 1.0 or some such *maximum* value). Finally, if all the red things that we were aware of were apples, then "x is red" would have provided evidence only to "x is an apple". (This would correspond to Vx RED(x) => APPLE(X)).

In the quaker example we considered a situation where two conflicting pieces of evidence needed to be resolved; evidential reasoning can deal with such situations. The details of an evidential treatment of inheritance and categorization in semantic networks are laid out in section 5, however, the following section gives a broad outline of the evidential framework.

1.7 Evidential reasoning

Evidential reasoning involves finding the most likely hypothesis from among a finite set of alternative hypotheses. This form of reasoning is essential when an agent must act and make choices on the basis of available information, and he does not possess sufficient information to pin down the exact state of the world. It was argued in section 1.2 that in such situations, a straight forward application of deductive reasoning is not very useful for making decisions because the agent would not just want to know the various possibilities, but also their relative likelihoods.[6]

Evidential reasoning may be summarized as follows:

Given a set of mutually exclusive hypotheses $\{\alpha_1, \alpha_2, \ldots \alpha_n\}$

For each hypothesis α_i

* Gather available evidence

* Combine evidence

Choose the hypothesis that receives the *highest* combined support

It is required that the evidence combination rule be such that the hypothesis which is *most likely* - given the information at hand, receives the highest combined support,

5. This is not to suggest that decision making is based on likelihoods alone. For instance, a complete account of decision making would at least have to incorporate the notion of "utility". The actual decision criteria may be arbitrary complex, but determining the likelihoods of the possible outcomes would still remain a basic step in the decision process.

The form of evidential reasoning employed in this thesis (cf. section 5.1) justifies inference, not on the grounds that the selected hypothesis is true in *all* possible worlds that are consistent with the agent's knowledge, but rather on the grounds that there are more possible worlds consistent with the agent's knowledge in which the selected hypothesis is true, than there are those in which one of the alternative hypothesis is true. If K denotes the agent's knowledge, then instead of K entailing α, if and only if all interpretations that satisfy K also satisfy α, we have the following situations:

from among $\alpha_1, \alpha_2, \ldots \alpha_n$, the most likely alternative α^* is found, where a hypothesis α_i is more likely than α_j if and only if from among all interpretations that satisfy K, more interpretations satisfy α_i than α_j. A measure of the likelihood of α_i is given by the number of interpretations that satisfy K and also satisfy α_i.

If only one of the alternatives is provable (and hence the rest are provably false), then only the provably correct hypothesis will receive any evidence; all other hypotheses will receive zero evidence. This is a situation in which one could have used deductive reasoning to make a choice, and in such a situation the evidential reasoning would also produce the same result.

A common objection to evidential reasoning has been that by abandoning standard deduction, one will be abandoning truth, and time and again it has been said that such reasoning is inexact and "unjustified". This results in a misconception that evidential reasoning is ad-hoc and informal. However, adopting evidential reasoning does not mean that one has given up the notion of truth. On the contrary, it means realizing that (absolute) Truth is often elusive, that one's knowledge is partial, and the desire to ascertain truth does not mean that if we ***cannot*** ascertain truth we should not try and seek what is most likely (or most probably) to be true.

Attempts to simulate evidential reasoning as a deductive system without facing up to the issue of likelihood, usually end up being mired in consequences such as non-monotonicity or probabilistic truth values. Other alternative approaches such as default logic also do not deal with likelihood and leave the issue of selecting an extension as an implementation detail. Likelihood becomes an issue of inference strategy, a quirk of implementation, and at best a fuzzy thing called heuristics.

Needless to say, there will occur situations in which two or more conflicting conclusions receive exactly equal evidential support and a system based on an evidential formulation will have to report an ambiguity or make arbitrary choices. However, this will happen only when a choice cannot be made because of the inherent nature of the world knowledge. For instance, if one is asked to choose between heads and tails as the possible outcomes of the toss of a *fair* coin, one is entirely justified in making an arbitrary choice. However, such cases will be extremely improbable, at least much less probable than those that will occur if all default rules were to have the same evidential import; which indeed is the case in default logic.

My apologies to the reader with a background in statistical inference and probabilistic reasoning, who may find this section stating the obvious. However, this thesis is primarily aimed at readers who are engaged in knowledge representation research, and given the tradition in this field, most of what has been said above is pertinent. In spite of continued interest in evidential reasoning [WUPAI 85], specially in the area of expert systems [Pople 77][Shortliffe 76][Duda et al. 76], researchers in knowledge representation have largely ignored evidential reasoning as a framework for modelling commonsense reasoning. This thesis establishes the relevance of evidential reasoning to mainstream knowledge representation work by demonstrating that an evidential approach suggests principled solutions to the problems of exceptions and multiple inheritance in semantic networks, and at the same time offers a uniform solution to the categorization problem.

1.8 System Overview

This section presents an overview of the proposed knowledge representation system with the help of two examples. The first example illustrates how knowledge may be encoded and accessed via networks of simple processing elements, and the second example describes how such a system has been simulated on a conventional computer.

1.8.1 Basic organization of the knowledge base system

The system's conceptual knowledge is encoded in a network referred to as the *memory network*. This network is composed of a large number of extremely simple processing elements (nodes) that interact with one another by propagating activation along weighted links. The memory network employs spreading activation to perform limited forms of inference such as inheritance and categorization in accordance with the theory developed in this thesis.

In order to perform specific inferences - as against exhibiting general associative behavior or modelling diffuse priming effects, the memory network must be capable of controlling the propagation of activation. An important aspect of this research was the development of control mechanisms for regulating the spread of activation. The control mechanisms suggested in this thesis are embodied in the pattern of interconnections between nodes as well as in the computational characterstics of the nodes. Furthermore, these mechanisms are entirely domain independent.

The translation from a body of knowledge to a network that embodies this knowledge and performs inferences based on it, is purely mechanical - in particular, it does not involve any ad hoc manipulation of weights to achieve the appropriate behavior.

In keeping with the commitment to massive parallelism, it is important that the network operate without the intervention of a central controller or a central interpreter. Consequently, the memory network is such that once the relevant nodes are activated, the appropriate inferences get performed automatically and the result becomes available in the form of activation levels of appropriate nodes. A mechanism for accessing the information encoded in the memory network is outlined below; a detailed description is provided in [Shastri & Feldman 84].

The information encoded in the memory network is accessed via other network fragments called *routines*. Thus, not only the representation of knowledge but also its access is actualized via networks. Routines encode canned procedures for performing specific tasks and are represented as a sequence of nodes (units) connected so that activation can serve to sequence through the routine. In the course of their execution, routines pose queries to the memory network by activating relevant nodes of the memory network. Once activated, the memory network performs the required inferences via controlled spreading of activation, and returns the answer by activating appropriate *response* nodes in the routine.

All queries are posed with respect to an explicit set of answers and there is a response node for each possible answer. The activation returned by the memory network is a measure of the evidential support for an answer. Response nodes compete with one other and the node receiving the maximum activation from the memory network dominates and triggers the appropriate action.

Figure 1.1 depicts the interaction between a fragment of an agent's restaurant routine and a part of his memory network. In this routine fragment, the task of deciding on a wine results in a query to the memory network about the taste of food and the decision is made on the basis of the answer returned by the memory network. Action steps are depicted as oval nodes, queries as hexagonal nodes and response nodes as circular nodes. The memory network in the example encodes the following information: Concepts in the example domain are characterized by two properties, has-taste and has-color. HAM and PEA are two concepts in the domain. HAM is SALTY in taste and is PINK in color, PEA is SWEET in taste and is GREEN in color.

The arcs in the memory network represent weighted links. The triangular nodes (called binder nodes) associate objects, properties and property values. Each node is an active element and when in an "active" state, sends out activation to all the nodes connected to it. The weight on a link modulates the activation as it propagates along the link. A node may become active on receiving activation from another node in the memory network or the routine. Binder nodes behave slightly differently in that they become active only on receiving simultaneous activation from a pair of nodes.

To find the taste of HAM a routine would activate the nodes has-taste and HAM. The binder node b1 linking has-taste and HAM to SALTY will receive coincident activation along two of its links and become active. As a result, it will transmit activation to SALTY which will ultimately become active.

If some routine needs to find an object that has a salty taste it would activate the nodes has-taste and SALTY. This will cause the appropriate binder node to become active and transmit activation to HAM. Eventually, HAM will become active completing the retrieval.

These two cases roughly correspond to how inheritance (finding property-values of a specified object) and categorization (identifying an object given some of its attributes) may be processed by the network. The above examples do not involve any evidential reasoning or actual inheritance, and are solely meant to give the reader an idea of the parallel implementation that is described in section 6.

1.8.2 An overview of simulation

This section describes how the behavior of the memory network is simulated on a conventional computer. The simulation proceeds in three stages, as depicted in figure 1.2. During the first stage, a high-level description of the information to be encoded in the memory network is processed by a compiler (SNAIL) and translated into a set of commands to a general purpose network builder (SPIDER). The high-level description input to SNAIL does not depend on any aspect of the parallel implementation.

During the second stage, SPIDER constructs a network in accordance with the commands generated by SNAIL. Finally, the activity of the network constructed by SPIDER is simulated using CISCON - a connectionist network simulator.

As an example consider the "quaker example" discussed in section 1.4.1. Assume that an agent has the following world knowledge:

"Most persons are non-pacifists"	----	S1
"Most quakers are pacifists"	----	S2
"Most republicans are non-pacifists"	----	S3
"Dick is a quaker"	----	S4
"Dick is a republican"	----	S5

Within an evidential framework, sentences such as S1 through S3 are viewed as evidential assertions and a numeric quantity is associated with them (cf. section 1.4.2). The precise interpretation of these numeric quantities and their subsequent use in drawing inferences is described in chapters 4 and 5, for now an intuitive interpretation of these numbers as being indicators of "evidential strenghts" will suffice.

An agent whose knowledge includes S1 through S3 may have the following view of the world:

70% of all persons are non-pacifists, while 30% of them are pacifists.

70% of all quakers are pacifists but 30% are non-pacifists.

80% of all republicans are non-pacifists while 20% are pacifists.

In addition to the above, let us assume that the agent also believes that 5% of all persons are quakers while 40% are republicans.

Let us also assume that persons have a property (attribute) "has-belief" (i.e. persons have the property that they hold beliefs), and that some of the values of the property "has-belief" are "pacifist" and "non-pacifist".

The above information may be input to SNAIL in the form of four lists:

i) a list of concepts,

ii) a list of properties and their associated values.

iii) a list indicating the "subtype" and "instance-of" relationship between concepts.

iv) a list specifying how members of a concept are distributed with respect to various property values. The numbers that appear in this list specify the distribution; these numbers have a simple interpretation that is explained in chapter 4.

The actual input to SNAIL is as follows:

```
(NB-concept '(   PERSON REPUBLICAN QUAKER
                 BELIEFS PACIFIST NON-PACIFIST))

(NB-property '((has-bel PACIFIST NON-PACIFIST))

(NB-is-a '(      (QUAKER PERSON 0.05)      (REPUBLICAN PERSON 0.40)
                 (DICK QUAKER 0.1)         (DICK REPUBLICAN 0.0125)
                 (PACIFIST BELIEFS 0.3)    (NON-PACIFIST BELIEFS 0.7))

(NB-delta '(     (PERSON has-bel PACIFIST 1.0 0.3)
                 (PERSON has-bel NON-PACIFIST 1.0 0.7)

                 (QUAKER has-bel PACIFIST 0.12 0.7)
                 (QUAKER has-bel NON-PACIFIST 0.02 0.3)

                 (REPUBLICAN has-bel PACIFIST 0.27 0.2)
                 (REPUBLICAN has-bel NON-PACIFIST 0.46 0.8) )
```

When the above input is provided to SNAIL, it generates commands to SPIDER which in turn constructs a network similar to the one shown in figure 1.3.

When the question: "Is Dick a pacifist or a non-pacifist" is posed to this network[7], the activation levels of the nodes PACIFIST and NON-PACIFIST stabilize to values that are in the following proportion:

1.00 : 0.66

This implies that based on the knowledge encoded in the network, it is more likely that Dick is a pacifist Furthermore, the ratio gives a quantitative measure for comparing the likelihood of Dick being a pacifist versus he being a non-pascifist

7. The exact steps involved in posing the question and obtaining the answer are described in chapter 6. In broad terms, the question is posed by activating the nodes DICK, has-belief, and BELIEFS, and enabling the IS-A links emanating from DICK.

1.9 Organization of the thesis

The rest of the thesis is organized into 6 chapters. Chapter 2 briefly describes the model of parallel computation adopted in this thesis and the reasons for its choice. Chapter 3 discusses some knowledge representation issues relating to Types (categories) and proposes an organization for the conceptual structure. Chapter 4 specifies the representation language for capturing the evidential nature of information associated with Types.

The evidential formulation and its applications to inheritance and categorization are developed in chapter 5. Section 5.1 derives an evidence combination rule based on the principle of maximum entropy, and section 5.2 compares this approach to the Dempster-Shafer evidence combination rule and the Bayes' rule for computing conditional probabilities. Section 5.3 derives a theory of inheritance based on the result derived in section 5.1, and section 5.4 compares this formulation to solutions proposed by Fahlman, Touretzky, and Etherington and Reiter. Section 5.5 develops the solution to the categorization problem based on the result derived in section 5.1. Both in the case of inheritance and categorization, the conditions are specified that the conceptual structure must satisfy for the solutions to apply. The solution developed in this chapter provides a natural treatment of defaults, exceptions and conflicting information.

The solutions developed in chapter 5 may be realized as a highly parallel network of active elements connected via weighted links. This parallel encoding is described in chapter 6, and it is proved that the resulting networks solve the inheritance and categorization problem in time proportional to the depth of the conceptual hierarchy. For pedagogical purposes, the network description and the proof of correctness are provided separately for inheritance (section 6.1) and categorization (section 6.2). Section 6.3 specifies how the two networks described in sections 6.1 and 6.2 can be synthesized into a single network that can solve the inheritance as well as the categorization problem. Section 6.4 elaborates on some implementation details. Finally, section 6.5 describes how the proposed system has been simulated on a conventional computer and presents the results of several simulations to illustrate the behavior of networks during inheritance and categorization tasks.

Chapter 7 discusses some limitations of the current approach and lists some unresolved issues. It also indicates possible directions that may be taken in pursuing the line of research described in this thesis.

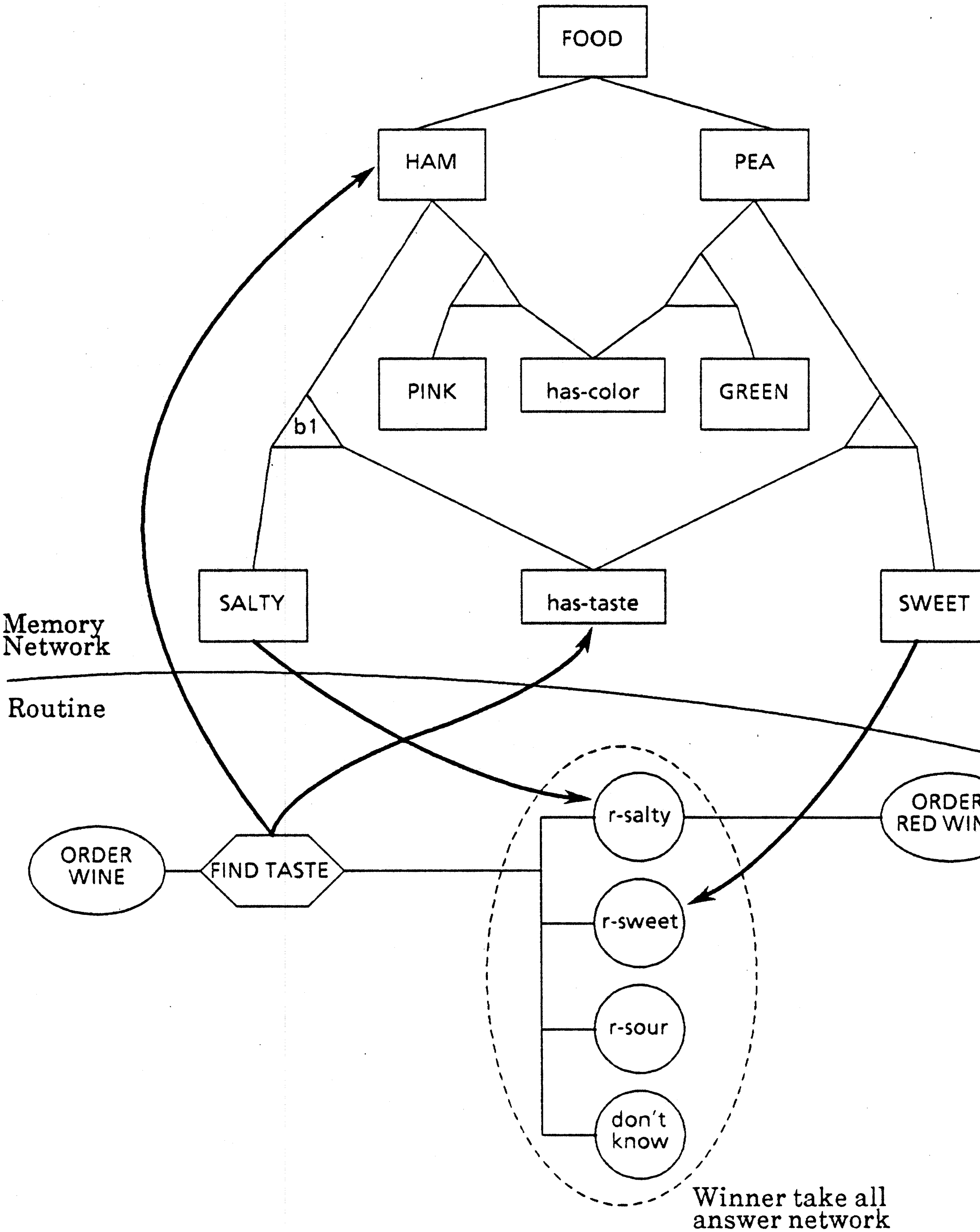


FIGURE 1.1 Connectionist Retrieval System

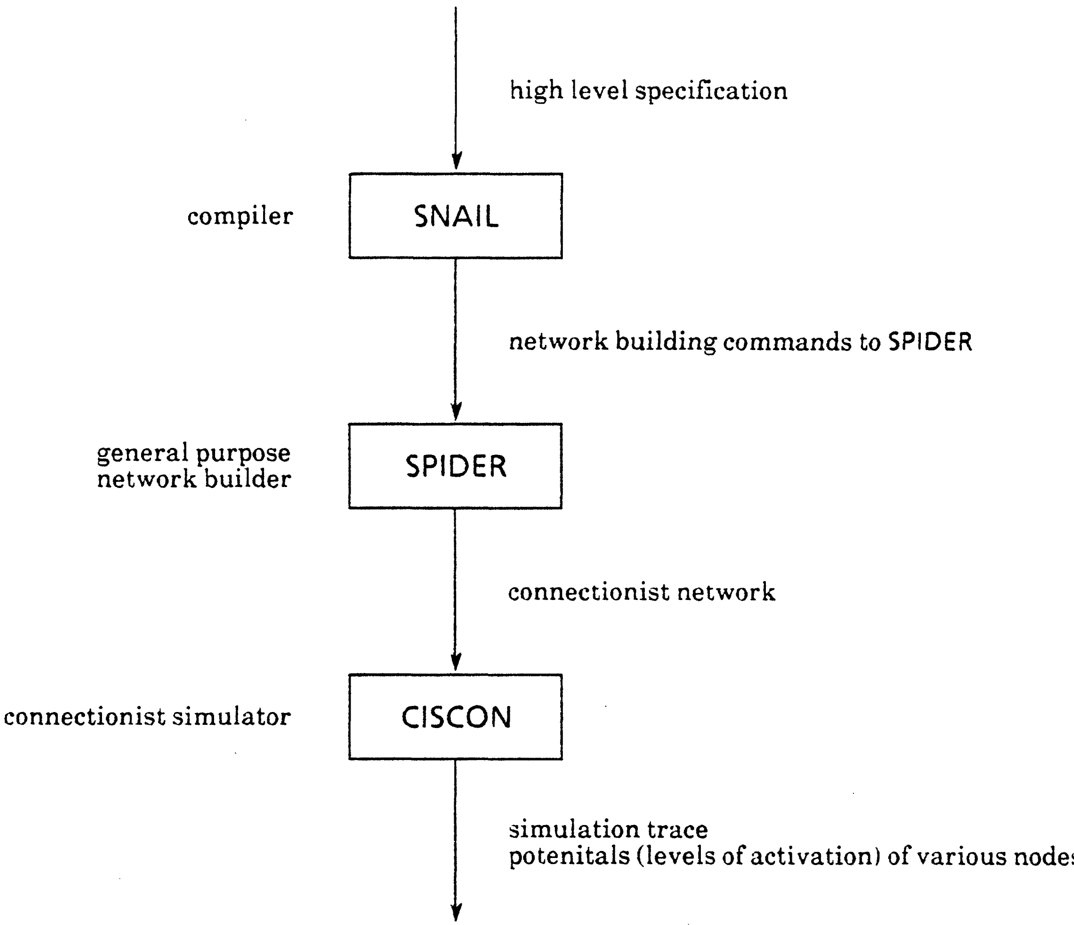


FIGURE 1.2 Overview of simulations

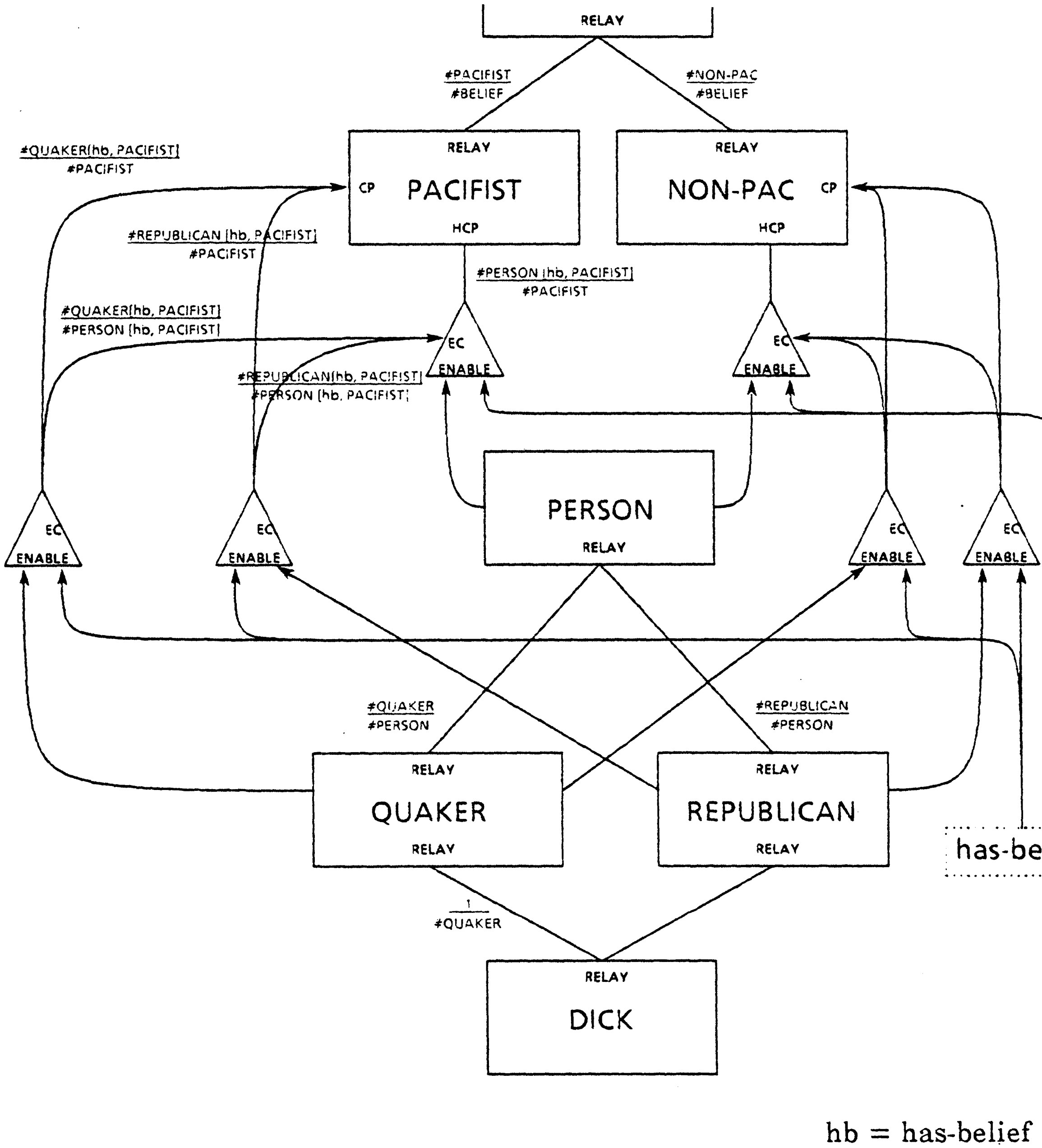


FIGURE 1.3 An example network

Chapter 2

Massive Parallelism

Human agents perform non-trivial cognitive tasks in but a few hundred milliseconds. If performed on a serial computer, these tasks would require millions of instructions, and it is fairly obvious that if one wants agents endowed with artificial intelligence to match or improve upon human performance, then at some point one will have to exploit parallelism. One may approach the issue of parallelism in two ways:

1. Treat parallelism purely as an implementation issue; work on problems independent of any consideration for parallelism, and having solved the problem, investigate the possibility of exploiting parallelism.

2. Take parallelism an an important premise and use it to guide the search for interesting solutions in the space of possible solutions.

This research was guided by the belief that in order to exploit the full potential of parallelism one must adopt the second strategy, and furthermore, one must focus on massively parallel models of computation. Consequently, I adopted connectionism as the underlying model of computation for this work.

The connectionist framework has emerged as a serious candidate for modeling cognitive processes. It provides a plausible model of computations carried out in neuronal networks, and independent of any such considerations, is a powerful model of massively parallel computation.

2.1 The connectionist model

A connectionist network consists of a large number of simple computing elements called units. Each unit has a very large number of incoming and outgoing connections and it communicates with the rest of the network by transmitting a single value. A unit transmits the same value to all elements it is connected to. The output value is closely related to the unit's *potential* and is best described as a level of activation. A unit's activation reflects the amount of activation the unit has been receiving from other units. All inputs are weighed and combined in a manner specified by the *site functions* and *potential function* in order to update a unit's potential. A more technical description follows:

A network consists of a large number of units connected via links. The units are computational entities defined by:

{q} : a small set of states, (fewer than 10)

p : a continuous value called potential

v : an output value, approximately 10 discrete values

i: a vector of inputs $i_1, i_2 \ldots i_n$ (this is elaborated below)

together with functions that define the values of potential, state and output at time t+1, based on the values at time t:

$$p_{t+1} \longleftarrow P(i_t, p_t, q_t)$$

$$q_{t+1} \longleftarrow Q(i_t, p_t, q_t)$$

$$v_{t+1} \longleftarrow V(i_t, p_t, q_t)$$

A unit need not treat all inputs uniformly. Units receive inputs via links (or connections) and each incoming link has an associated *weight.* A unit weighs each input using the weight on the appropriate link. Furthermore, a unit may have more than one *input site* and incoming links are connected to specific sites. Each site has an associated site-function. These functions carry out local computations based on the input values at the site, and it is the result of this computation that is processed by the functions **P**, **Q** and **V**. The notion of sites is useful in defining interesting behavior. For example, some inputs may be treated as "enabling" inputs whose presence or absence determines whether the unit attends to the remaining inputs or ignores them. The functions P, Q and V are arbitrary but keeping with the underlying philosophy of these models, they are assumed to be simple. For a detailed account of the connectionist model refer to [Feldman & Ballard 82].

In course of my work, the only major departure I have made from the standard model is to assume that the output of each unit may have the same precision as its potential (cf. Chapter 6).

Connectionist models have been used by researchers in the domain of vision. [Ballard 84] describes the application of the paradigm to problems in low-level vision. [Sabbah 85] presents a system to recognize objects in the origami world, and [Feldman 82] proposes a connectionist model of the visual system. [McClelland & Rumelhart 81], [Rumelhart & McClelland 82] describe a network that is similar to a connectionist network, for word or letter recognition from visual input. Other applications are in the areas of speech errors [Dell 80]; early language processing [Cottrell 85]; low-level motor control like that of the Vestibulo-ocuiar Reflex [Addanki 84]. Recently, Hopfield and Tank have shown that a good (not necessarily optimal but close to optimal) solution to the Travelling Salesman problem may be obtained within a few time steps by using a massively parallel network. Their networks are similar to the kind of networks employed in this work in that they also employ units that have a continuous (analog) potential. For a collection of papers describing state-of-the-art research based on connectionist models refer to [Cognitive Science 85]. This research demonstrates that connectionist network can be "programmed" to carry out relatively sophisticated reasoning tasks such as inheritance and categorization that were regarded to be "hard" problems for such networks.

2.2 Relevance of connectionist models to this work

There were three reasons for choosing connectionism as the model of parallel computation:

1. It is a *massively* parallel model of computation.

2. It provides a natural encoding of evidential reasoning.

3. It permits the encoding of non-trivial computations.

2.2.1 Importance of massive parallelism

The massively parallel nature of connectionism permits one to exploit parallelism to the fullest. The fine grain of parallelism permits one to assign *a single processing element to each unit of information.* This has the following interesting consequence:

Assume that besides enumerating facts about the world, we also identify the important *inferential connections* between these facts. Now if we encode each piece of information as a connectionist unit (henceforth unit), and an inferential interconnection between pieces of information as an explicit link between the appropriate units, then we can view inference as spreading of activation in a connectionist network[1].

1. Needless to say, this is a gross oversimplification, but the basic idea is sound - at least in the context of limited inference. That the idea works is demonstrated by this thesis.

The above metaphor has tremendous appeal because it suggests extremely efficient parallel implementations. This is not a new metaphor, and dates back at least to Quillian's work on semantic nets [Quillian 68] who employed the propagation of discrete "activation tags" in a network to find common properties of concepts. Quillian's semantic nets consisted of nodes interconnected by labeled links. Nodes in the network represented "word concepts" that were defined in terms of their associations with other word concepts. He proposed the notion of "breadth first tracing", with the trace leaving activation tags on nodes. This formed the basis of the "intersection[1]' operation for finding common properties of concepts.

Fahlman [Fahlman 79] was the first to seriously investigate the possibility of exploiting the power of parallelism using a Quillian like propagation of activation tags (markers). Fahlman's NETL system design consists of a central computer connected to a large number of node and link elements, each of which is a hardware element. The central computer is a serial machine and issues commands to links and nodes. Each basic concept is represented as a node, and each assertion by a link. Nodes and links communicate over a shared bus with a central computer. Nodes have a small number of memory bits, called *marker bits.* The nodes propagate markers along links, in response to the instructions issued by the controller. This operation, called *marker passing,* is the basis of computations performed by the network.

NETL uses *marker passing* to perform simple inferences based on operations like set *intersection,* and *transitive closure.* The intersection operation locates items that share a set of properties (i.e. categorization), whereas the transitive closure operation handles *inheritance* as well as closures of relations like *part-of.* These operations are performed in parallel and allow the system to conduct a very fast search, essentially independent of the size of the knowledge base. The marker passing mechanism is very similar to Quillian's idea of breadth first tracing using activation tags. However, Fahlman suggests a complete design of a hardware machine to implement marker passing as a parallel operation.

A limitation of marker passing systems is that the communication between network elements is in terms of a small number of discrete markers, each of which may be on or off. Furthermore, each element is only capable of detecting the presence or absence of a marker in the input. This makes the system incapable of supporting evidential reasoning. For example, categorization amounts to finding all concepts that have all the specified set of properties. As each propety has the same import, there "best match" or a "partial match". Furthermore, Fahlmar inheritance problem suffers from serious drawbacks, the an the network are sensitive to race conditions, specially ù exceptions and multiple hierarchies. The limitations c systems are discussed in [Fahlman 82], [Fahlman et ai. 81] and alsolii secuon 5.4. The connectionist networks described in chapter 6 encode the evidential theory of inheritance and categorization developed in chapter 5.

2.2.2 Ease of encoding evidential computations

The connectionist framework provides the neccessary primitives for capturing notions like "weighted evidence" and "evidence combination". This makes it ideal for performing evidential reasoning, and differentiates connectionist models from schemes like NETL. A unit may be interpreted as representing a hypothesis and the inputs to the unit may be viewed as evidence provided to it by the rest of the network. A unit's potential may be viewed as the result of combining all the evidence impinging on the unit using the evidence combination rule encoded by the site functions and the potential functions. Connectionism does not prescribe any single theory of combining evidence and it is possible to encode a wide range of these in the connectionist framework.

2.23 Sophisticated processing elements

The presence of multiple sites, and multiple states make it possible to encode non-trivial behavior in connectionist network. It becomes possible to encode rules for performing built-in inference that are far more sophisticated than those of marker-passing systems. Furthermore, unlike NETL, where a serial computer controlled the activities of the parallel system, connectionist networks can support interpreter-free systems that may operate autonomously. The latter is significant from the point of view of computational tractability because it frees us from the interpreter bottleneck.

When knowledge is encoded in connectionist networks the traditional distinction between representation and an inference engine (interpreter) gets blurred. Since the networks consist of active elements and the computations in a network take place via spread of activation, the "static" structure of the network also determines the "dynamic" relationships between bits and pieces of knowledge. Thus, during any computation or information processing task, the structure itself determines what information is relevant for the computation. The links, weights on links and the computational characterstics of units encode not just the knowledge but also the way in which various constituents of knowledge interact during computation. The strong coupling between the structure of knowledge and inference is a desirable state of affairs.

Chapter 3

Structure of Knowledge

It was argued in section 1.5 that in the light of the problems concomitant with inheritance and categorization, a simple treatment of concepts as one place predicates, and properties as binary relations is inadequate, and so are other frame based schemes that are mere notational variants of the above. We had argued that an evidential treatment of conceptual information would be more appropriate for formalizing categorization and inheritance. This section presents our intuitions about conceptual knowledge that form the basis for the formal language specified in section 4. We focus mainly on the notion of Types because it directly relates to inheritance and categorization.

At the outset we would like to state that in discussing conceptual knowledge we are concerned with an agent's *internal* representation of the external world; we are not interested in the external world per se, but rather in its conceptualization by the agent. We will refer to the latter as the agent's ***conceptual structure.***

We begin by discussing the notion of ***conceptual attributes***, which we consider to be fundamental to the notion of concepts.

3.1 Conceptual attributes

A cognitive agent interprets the external world in terms of *conceptual attributes* and their associated *values*, and his entire factual knowledge is represented using these attributes and values. In the restricted context of vision, a conceptual structure may be defined in terms of visual attributes like "has-color" (with values such as red, blue, purple), "has-shape"(with values such as round, oblong, square), "has-size", "has-texture" etc. Such a conceptual structure may be extended by including non visual attributes like "has-taste", "has-weight," "has-temperature," "has-odor," "has-location," "has-utility" and "has-function" (i.e. has-use). In addition to the conceptual attributes mentioned above, other knowledge structuring relations such as is-a-kind-of, is-a-part-of are also considered to be conceptual attributes. The distinction between different kinds of conceptual attributes is discussed in section 3.3.1. Notice that an "attribute-value" pair in our notation corresponds to the term "feature" in many psychological models [Smith & Medin 81], thus what we refer to as [has-color RED] in our notation would correspond to the feature RED.

The explicit identification of conceptual attributes (henceforth, simply attributes) and their values is a crucial first step in extracting the structure of knowledge because all other components of the conceptual structure are defined in terms of attributes and their values.

Attributes need not be unstructured entities. One may easily imagine complex attributes like "has-shape" that may have finer structure consisting of several "sub" attributes such as "has-length-to-breadth-ratio"; the attribute "has-color" may itself have other attributes such as "has-hue", "has-brightness", "has-saturation"; and "has-physical-property" may be regarded as an attribute in some domain but may be composed of more specific attributes like "has-size", "has-weight", "has-color" etc. However, in our present endeavor, we will assume that all attributes are primitive.

3.2 Concepts

Concepts are *labelled* collections of [attribute, value] pairs. For instance, a concept labelled FIDO may partially consist of the following [attribute, value] pairs:

{ is-an-instance-of DOG,

is-an-instance-of ANIMAL,

has-body-part LEGS,

has-body-part TAIL,

has-coat-type FURRY

... }

The values of attributes are also concepts and hence, concepts may be arbitrarily complex. For example, the concept of FIDO refers to other concepts such as DOG and FURRY. This definition does not imply circularity because some concepts (in particular values of properties such as color) are grounded in perception and some attributes are assumed to be innate.

Given that concepts are described as (labelled) collections of [attribute, value] pairs, there exists a very direct relationship between concepts that denote objects possesing a certain attribute value, and the attribute value itself. For example, one may think of RED as being a value of the attribute has-color, or one may create a concept RED-THING denoting "all red things" by attaching the label RED-THING to the [attribute value] pair [has-color RED]. There are no a priori criteria for determining whether some attribute value has been so labelled.

Concepts may denote different types of things in the domain. Some examples are: individuals, categories, events, locations and relations. For instance, concepts may denote "my dog Fido", "the color red", "Dog", "Color", "the Sox Phillies game", "the concert tonight" or "John's passing of the ball to Leo".

All attributes need not be applicable to all concepts. In general, different types of concepts will have different types of attributes applicable to them. For example, attributes such as "is-an-instance-of", "has-color", "has-shape", and "has-size" may apply to physical objects, whereas attributes such as "has-location", "has-agent", and "has-time-of-occurrence" may be associated with events.

3.3 Attributes: properties and structural links

Attributes may be classified into two broad categories: ***properties*** and ***structural links***. This distinction is crucial and forms the basis of controlled interactions that may occur between concepts.

Structural links: Structural links provide the coupling between structure and inference. They reflect the epistemological belief that world knowledge is highly organized and that much of this structure can be factored out to provide general *domain independent* inference rules. Structural links are attributes that have this quality and are used to provide built-in inference paths. These paths correspond to special axioms that define the capabilities of specialized inference engines employed in limited inference formalisms such as [Allen & Frisch 82]. The most representative structural link is the is-an-instance-of link that is used for inheritance in semantic networks.

One can extend the notion of property inheritance to include other structural links such as the is-a-part-of, and occurs-during links [Schubert et al. 83][Allen 83]. For example, is-a-part-of links may be used to infer values of attributes such as has-location, while occurs-during links may permit inferences pertaining to time. Each structural link has an associated set of properties that may be inherited along the link and this information may be used to perform inferences.

Properties: Properties correspond to the intrinsic features of concepts and may vary from one domain to another. When describing physical objects the relevant properties may be has-weight, has-shape, and has-color, while events may have properties such as has-location, has agent, and has-time-of-occurence. So far, properties roughly correspond to the notion of "roles" of KLONE [Brachman 77], "role nodes" of NETL [Fahlman 79] and "slots" of

FRL [Robert & Goldstein 77]. However, in spite of superficial similarities, our interpretation of a property and its associated values is different from any of the above approaches. The crucial difference is that we give an *evidential interpretation* to the relation between property values and the concepts they describe. [Cf. section 3.6].

3.4 Types and Tokens

3.4.1 Tokens

An agent interpretes the world as consisting of instances. These are represented as Tokens in his conceptual structure. For example, Tokens may represent: "Fido the Dog", "the table in my office" and "the location that is the top of my table".

3.4.2 Types

An agent has to deal with a complex and constantly changing environment, consisting of a large number of objects and an even larger number of interactions between them. In comparison, the memory resources and processing capabilities of an agent are extremely limited and in order to survive it becomes essential that the agent impose some structure on the external world in order to reduce its complexity. A way of structuring the knowledge about the environment is to detect and record similarities between objects that occur in the environment. Once recorded, these similarities may be exploited to categorize novel objects and to make predictions about their properties. Types serve exactly this purpose.

Types are *abstractions* defined over Tokens, and they arise when there are similarities in a large number of Tokens. Types are summary descriptions that may be viewed as encoding the agent's belief that there are objects in the physical world that *conform* to these description and that these descriptions may be used to categorize objects, and thence to make inferences about them.

The most important inference supported by Types is inheritance. For example, having identified an object (Tweety) as a bird, and knowing that birds fly, an agent may infer that Tweety flies. Without a notion of Type, an agent will not be able to perform such inferences.

Types ensure that the "quantum" of knowledge remains within manageable bounds. Once a Type exists, the shared properties need only be encoded in the representation of the Type, and need not be copied for each Token separately.

The mere presence of similarities in a large number of Tokens is not a sufficient condition for the existence of a Type. The above condition is too weak and may lead to a proliferation of Types. The primary justification for creating a new Type would be its usefulness to the agent. It is assumed that Types arise out of recognizing similarities that help the agent make useful distinctions. Although, one may create a Type that corresponds to "all dogs that have more than 13 fleas on them" - and there may be numerous such objects (dogs), such a Type would hardly serve any useful purpose.

The is-an-instance-of relation expresses the relationship between Tokens and Types while the inverse relationship between Types and Tokens is expressed by the "is-instantiated-by". Thus, FIDO is-an-instance-of DOG and DOG is-instantiated by FIDO.

Property values and Types

The [attribute, value] pairs associated with a Type are summary descriptions of the property values of its Tokens. Thus, the Type ELEPHANT may own the *value* GRAY for the property has-color to represent the fact "most elephants are gray", and the Type APPLE may own the values RED, GREEN, and YELLOW to represent that "apples may be red, green or yellow".

However, simply associating property values with Types is not sufficient.

For a given Type, a property value may occur more often among its instances than some other property value. For example, there may be more instances of apples that are red in color, than those of apples that are green. This sort of information may be useful to an agent in making predictions about the world, and hence, he may want to remember such information. Furthermore, there may be many concepts that share a property value. For example, both apples and grapes share the color value green. However, a shared value may be more indicative of one concept than of another. For example, green may be more indicative of grapes than apples. Such information may also be relevant to an agent in making predictions about the world. An epistemologically adequate representation should allow additional information to be stored with property values in order to capture the distinctions alluded to above. As is discussed in section 3.6, and subsequently made precise in section 4, our formalism does allow such distinctions to be made.

3.4.3. Hierarchies

The process of abstraction need not stop at one level. Abstractions over Types may yield more abstract Types (or a Type may be differentiated to produce more refined Types). This leads to a hierarchical structure. In general, multiple hierarchies may be defined over the same set of Tokens. For example, one may define a hierarchy over physical objects based on the functions they perform (i.e. their use), or one may classify physical objects according to their physical properties based on their form (i.e. their appearance). The result of having multiple hierarchies is that each Token may be related to more than one Type. A toy example of multiple hierarchies is shown in figure 3.1, where persons have been classified according to their religious beliefs in one hierarchy, and according to their political beliefs in another. Notice that DICK is related to two Types: QUAKER and REPUBLICAN.

3.5 A proposal for structuring concepts: "multiple views"

In this section we suggest a particular organization for concepts. Since our primary interest is in suggesting a way of *structuring* or organizing concepts, and not in the compilation of human knowledge, we would not argue for or against a *specific* set of Types that fill the proposed structure.

In the proposed scheme, concepts are organized in a three tier structure as shown in figure 3.2

The topmost tier consists of a pure taxonomy that is called the *ontological tree.* This tree classifies the universe of concepts into several distinct *ontological types,* where any two ontological types represent fundamentally different sorts of things. If these were to correspond to *categories* of Aristotle, then at the first level of branching, the concepts in the ontological tree would be: Substance, Quantity, Relation, Time, Position, State, Activity, and Passivity. Alternately, using *ontological categories* suggested by Jackendoff [Jackendoff 83], one would have: Things, Place, Direction, Action, Event, Manner, Amount, Sound, Smell, Time etc., as the ontological types at the first level of branching.

Keil [Keil 79][Sommers 65], has argued extensively in support of a hierarchical structure composed of *ontological categories.* Some ontological categories suggested by Keil are: Thing, Physical object, Solid, Aggregates, Event, Intentional event, Nonintentional event, Organism, Functional artifact, Animal, Plant, Human, Nonsentient etc. Keil derives ontological categories using the principle of *predicability,* which says that different sorts of things have different sorts of predicates applicable to them (predicates correspond to properties in our terminology), and one may classify things according to the predicates that apply, or do not apply, to them.

Keil describes what he calls the *M-constraint* and the *W-constraint,* (based on earlier work by Sommers) that rule out the possibility of an ontological category having multiple parents. As a consequence, ontological categories form a strict taxonomy. Figure 3.3 gives an example of an ontological tree based on [Keil 79]. According to Keil, predicates may be attached to categories in the ontological tree, and once a predicate is attached to a category, it applies to all categories that are dominated by this concept. This agrees with our notion of property applicability described in section 3.6.

In the proposed structure, the leaves of the ontological tree are all Types, and we envisage these to be concepts such as Animal, Human, Furniture, Instrument, Liquid, Color, Shape, Taste etc. These concepts roughly correspond to the superordinate categories of Rosch [Rosch 75]. To our mind, superordinate categories have the right level of complexity to be the leaves of the ontological tree.

The third, or the lowest tier of the conceptual structure consists of Tokens.

The second tier consists of a large number of taxonomies called *views*. The root of each view is a leaf of the ontological tree, and the leaves of each view are Tokens. There may be multiple views that have the same leaf of the ontological tree as their root. Furthermore, there may be multiple views that have the same Token as one of their leaves. The latter implies that Tokens may have multiple parents, however each parent of a Token must lie in a distinct view.

Each view corresponds to a classification of the Tokens represented by a leaf of the ontological tree. Thus, multiple views permit the agent to classify the same set of Tokens in different ways.

The organization suggested above offers advantages permitted by tangled hierarchies by allowing Tokens to have multiple parents, but retains the tree like characterstic that helps in analyzing the properties of the conceptual structure.

3,6 A representational notation

We will employ a graphical notation in order to present the role of evidence in the representational framework. Figure 3.4 displays a sample network encoding the following information: "Fruits are a kind of Things, Apple is a kind of Fruit, Things have the property color, Apples are generally Red or Green, and Red and Green are instances of Color". (This graphical notation is not a description of a connectionist networks. Those are discussed in section 6).

The representation uses three kinds of nodes: the concept nodes, property nodes and binder nodes. Concept nodes label collections of [attribute, value] pairs. While properties and values are associated to concepts via binder nodes, structural links are encoded directly as links. The framework permits associating properties as well as property-values with concepts. For example, the binder node b1 in the above network associates the property of having color (has-color) with Things (and hence, Fruits and Apples) without specifying any particular color values. On the other hand, the Binder node b2 represents: "the value of the property color for Apples may be Red" and also: "something that has color Red may be an Apple". The interpretation of b3 is analogous to that of b2 with Green replacing Red. The interpretation of node b4 is slightly different. It represents the agent's belief that there is a non zero possibility that Apples may have other colors besides Red and Green. Thus, b4 provides a compact representation for: "it is possible that Apples could be Blue, Yellow, Brown ..." This would otherwise require a binder node, one for each possible color value, to associate these color nodes with Apple.

A weight - in the range of 0 and 1 - is associated with each link and provides the basis for an evidential semantics of knowledge. The precise interpretation of these weights is given in section 4, but a general explanation follows. With reference to Figure 3.4 the weight W1 on the link from b2 to RED is a quantitative measure of the evidence provided by the fact "an object

X is an Apple" to the fact "the color of X is Red" and similarly, the weight W4 on the link from b2 to APPLE is a measure of the evidence provided by the fact "the color of an object X is Red" to the fact "X is an Apple". The weights W2 and W5 have a similar interpretation for the relationship between Apple and Green. The weights W3 and w6 measure the likelihood of Apples having colors other than Red and Green.

The graphical notation points out some salient features of the representation:

Applicability of properties: The framework permits associating properties as well as property values with concepts. Thus, it is possible to specify the properties that apply to a concept. Once a property is associated with a Type it gets associated with all Types or Tokens that occur below the Type in the is-an-instance-of hierarchy. Furthermore, a Type or Token may own a value for a property only if the property is owned by itself or by a Type higher up in the is-an-instance-of hierarchy. For instance, APPLE may own a value for has-color because this property is owned by FRUIT which is a supertype of APPLE.

Multiple and weighted property values: Apples could be Red or Green in color, and the representation of the value of the property has-color for the Type APPLE accounts for both these colors. The weights wl and w2 would depend on the agent's beliefs about what fraction of Apples are Red and what percentage are Green. This is explained in section **4.**

Distinction between Property and Value: The representation includes a node has-color and a node COLOR. These two represent two distinct aspects of the knowledge encoded in the network. The node has-color represents a property whereas the node COLOR represents a Type whose instances may include WHITE, BLUE, RED etc, each of which could be a value of has-color.

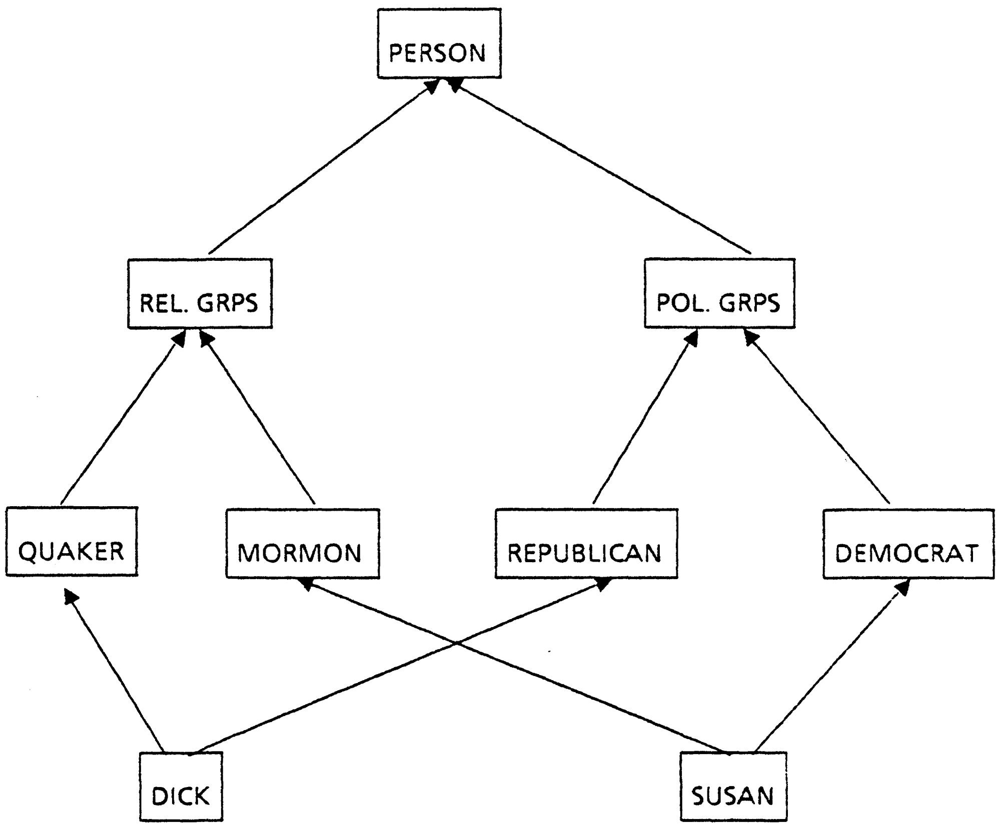
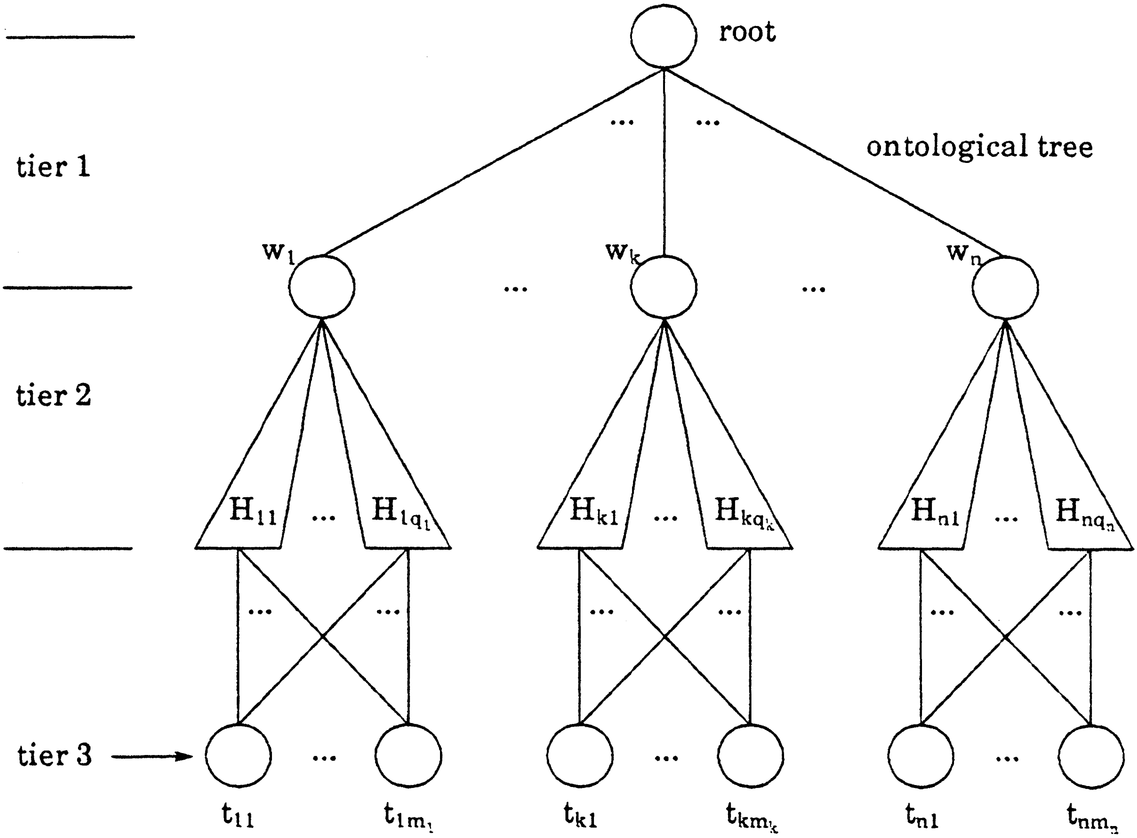


FIGURE 3.1 Multiple hierarchies



w_1, ..., w_n are leaves of the ontological tree.

H_{i1}, ..., H_{iq_i} are q_i views defined over tokens of ontological type w

t_{11} ..., t_{nm_n} are tokens.

A token may have multiple parents but at most one parent per v

FIGURE 3.2 Multiple view organization

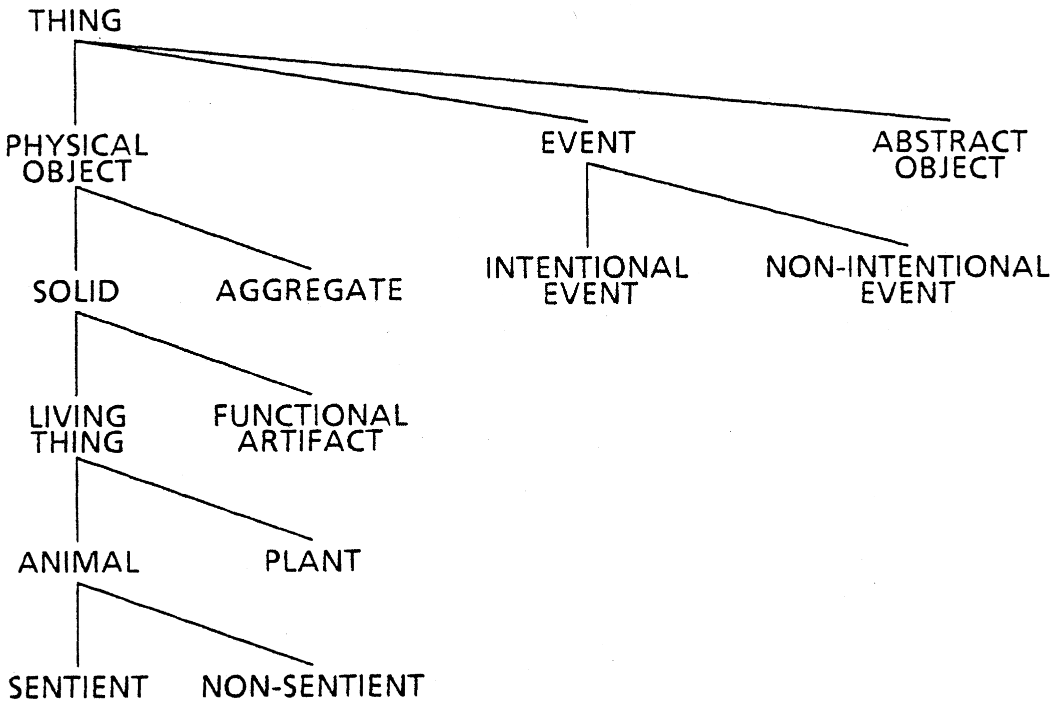


FIGURE 3.3 Ontological categories

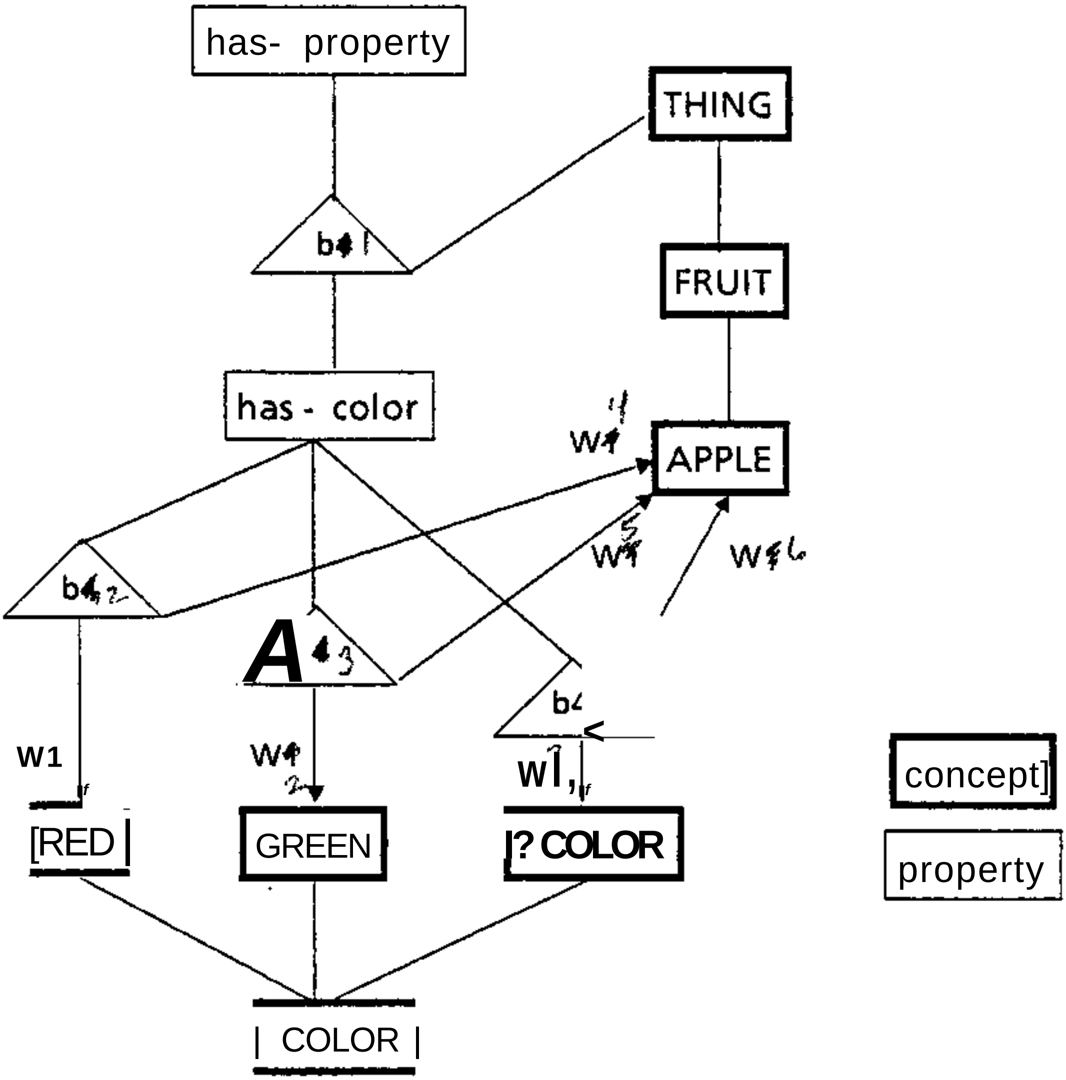


FIGURE 3.4 A graphical notation

Chapter 4

Representation language

4.1 A formal description of the representation language

This section specifies the representation language that is used for formalizing the problems of inheritance and categorization.

An agent's apriori knowledge consists of the septuple:

$\Theta = \langle \mathbb{C}, \Phi, \lambda, \Lambda, \#, \delta, \lessdot \rangle$, where

$\mathbb{C}$ is the set of *concepts*, Φ is the set of *properties*, λ is a mapping from $\mathbb{C}$ to the power set of Φ, Λ is a mapping from Φ to the power set of $\mathbb{C}$, $\#$ is a mapping from $\mathbb{C}$ to the integers I, δ the *distibution* function is a mapping from $\mathbb{C} \times \Phi$ to the power set of $\mathbb{C} \times I$, and $\lessdot$ is a partial ordering defined on $\mathbb{C}$. These terms are explained below in a more meaningful manner.

$\mathbb{C}$ is the set of concepts and Φ is the set of properties. These terms were described in sections 3.3 and 3.4.

For each $C \in \mathbb{C}$, $\lambda(C)$ is the subset of Φ that consists of those properties that are applicable to C. For example, λ(APPLE) may be {has-color has-taste has-shape}. As discussed in section 3, different properties apply to different sorts of concepts.

For each $P \in \Phi$, $\Lambda(P)$ is the subset of $\mathbb{C}$ that consists of all possible values of P. For example, Λ(has-color) may be {RED, GREEN, BLUE, YELLOW, BROWN}

For each $C \in \mathbb{C}$,

If C is a Token then $\#C = 1$, and if C is a Type then $\#C =$ the number of instances of C *observed by the agent.*

Recall that values are also concepts. If V is a value of some property P then #V is defined to be equal to the number of instances that possess this value. Thus, #RED equals the number of red colored entities in the domain.

Given a concept C and a property $P \in \lambda(C)$,

#C[P,V] = the number of instances of C that are observed by the agent to have the value V for property P

Thus, #APPLE[has-color, RED] equals the number of red apples observed by the agent.

The above notation may be extended to include multiple property values. Thus,

$\#C[P_1,V_1]\ [P_2,V_2]\ \ldots\ [P_n,V_n]$ = the number of instances of C, observed to have the value V_1 for property P_1, value V_2 for property P_2, ... and value V_n for property P_n.

The distribution function $\delta(C,P)$, where $C \in \mathbb{C}$ and $P \in \lambda(C)$, specifies how instances of C are distributed with respect to the values of property P. Recall that a concept may have several values for the same property and hence, if C is a Type, then $\delta(C,P)$, corresponds to the summary information abstracted in C based on the instances of C.

Example: if $\Lambda(P)$ = {RED, GREEN} then, δ(APPLE, has-color) may be expressed as:

{(RED, 60), (GREEN 40)}

indicating that 60 apples are red and 40 are green.

Alternately, $\delta(C,P)$ may be expressed in terms of #C[P,V]'s:

Thus, S(APPLE, has-color) may also be expressed as:

{#APPLE[has-color, RED] = 60, #APPLE[has-color, GREEN] = 40}

It was pointed out in section 3.3 that properties correspond to roles or slots. Consequently, 5 may be taken as a specification of role values or slot fillers. However, there is a crucial difference. For a given concept C, and a property P, S(C,P) may specify *multiple* values for P. Furthermore, the specification of 5(C,P) involves stating the quantities #C[P,V] for each V € A(P). The ratios of these quantities are used during inheritance and categorization. How this is done is discussed in chapter 5.

It may be observed that if the domain is extremely well behaved and the concepts have unique property values such that generalizations such as:

vx TYPE(x, T) => P(x,V)

{i.e. for all instances of Type T, the value of property P is V.}

may be made, then *S* reduces to a "non-evidential" mapping. In such a situation 5 may be expressed simply as a binary predicate:

P(C,V), or equivalents 8(C,P) = V.

The following observations about 8 are significant.

L *An agent's knowledge of the distribution mapping 8 is* ***partial.*** Thus, the agent *may not know* 5(C,P) *even though* P *may belong to* \(C). It is assumed that an agent acquires and stores only those distributions that are useful to him.

2. Although we have used absolute numbers to specify the distributions and the size of concepts, it is shown in section 5.3 that in order to solve the inheritance and categorization problems, an agent need only deal with *relative* information and rational numbers that lie in the interval [0,1],

The relation <^ structures the concepts in C into a partially ordered set and corresponds to the is-an-instance-of relation mentioned in section 3. In this formulation, < relates Types to other Types as well as to Tokens.

The applicability of properties is such that: if P € X(A), then for all B such that B < A, P € \(B).

The ordering induced by < on C may be compactly represented in the form of an ***ordering graph.*** Figure 41 depicts an ordering graph for a specified < Each node in the graph denotes a concept. A directed link connects aj to every node aj, (a[* ap such that aj < a; and there exists no a^ (other than a} and a;) such that aj < a^ <^ aj. If there is a direct link from a^ to a; then aj is referred to as a ***parent*** of **a^.**

Given a set of concepts S, where S = {s^ S2 ... s_n}, if a; is such that for all S} € S, *S[* < aj, and there is no a^ (other than aj) such that for all s^ € S, *S[* < a^ < aj, then aj is a ***reference concept*** for S. For instance, in figure 4.1, b is the reference concept for {c **e** f}.

4.2 Some properties of the "multiple views" organization

In terms of the language described in section 4.2, the multiple views organization described in section 3.5 has the following restrictions:

1. The partial ordering < is restricted. Refer to section 3.5 for a detailed specification. For example, concepts in tier I (the ontological tree) are organized in a strict hierarchy, but the concepts in tier III may have multiple parents from among concepts in tier II.

2. The applicability of properties is such that: if *co* is a leaf of the ontological tree then for all C such that C < <o, if P € X(C), then P € \(<o). In other words, the applicability of properties is defined within the ontological tree, and no new properties become applicable at tiers II and III.

Caution: This restriction applies to properties and NOT property values. Thus, information about property values may be attached to concepts in tiers II and III.

In addition to the above, we assume that the conceptual structure decribed in section 3.5 has the following *well-formedness* property:

WFR-mv-1: If ω is a leaf of the ontological tree then for each $P \in \lambda(\omega)$, $\delta(\omega,P)$ is known.

4.3 An Example

An agent's knowledge about a hypothetical domain may comprise:

$\mathbb{C}$ = {THING, FRUIT, APPLE, GRAPE, A-5, A-9, G-8, COLOR, RED, GREEN, TASTE, SWEET, SOUR}

Φ = {has-taste, has-color}

λ(FRUIT) , λ(APPLE), λ(GRAPE), λ(A-5), λ(A-9), λ(G-8)
= {has-taste, has-color}

λ(COLOR), λ(TASTE), λ(RED), λ(GREEN), λ(SWEET), λ(SOUR) = $\emptyset$

Λ(has-color) = {RED, GREEN},

Λ(has-taste) = {SWEET, SOUR}

#FRUIT = 150
δ(FRUIT, has-color) = {(RED, 65), (GREEN 85)}
δ(FRUIT, has-taste) = {(SWEET, 100), (SOUR 50)}

#APPLE = 100
δ(APPLE, has-color) = {(RED, 60), (GREEN, 40)}

#GRAPE = 50

δ(GRAPE, has-taste) = {(SWEET, 30), (SOUR 20)}

#COLOR = 150; #RED = 65, #GREEN = 85;

#TASTE = 150; #SWEET = 100; #SOUR = 50;

$\leqslant$ is given by:

(APPLE $\leqslant$ FRUIT), (GRAPE $\leqslant$ FRUIT), (A-5 $\leqslant$ APPLE),
(A-9 $\leqslant$ APPLE), (G-8 $\leqslant$ GRAPE), (RED $\leqslant$ COLOR),
(GREEN $\leqslant$ COLOR), (SWEET $\leqslant$ TASTE), (SOUR $\leqslant$ TASTE)

Notice that the agent's knowledge is partial and he *does not store all the information observed by him.* Although the agent has observed 100 apples and 50 grapes, only 2 apples and 1 grape are stored explicitly. Furthermore, for apples and grapes, the agent does not remember δ(APPLE, has-taste) and δ(GRAPE, has-color). Finally, 3 instances of apples and grapes have been represented (A-5, A-9, and G-8), the agent only remembers δ(G-8, has-taste).

Figure 4.2 represents a part of the information given above in the graphic notation introduced in section 3.6. One may now assign a precise meaning to the weights on links. If a binder associates a property P, a concept C and a value V, then the weight on the link from the binder to the the value V is given by:

#C[P,V] / #V

and the weight from the binder to the concept C is given by:

#C[P,V] / #C

If concept A is a parent of concept B in the ordering graph defined by $\mathbb{C}$ and $\leqslant$, then the weight on the link from A to B is given by:

#A / #B.

The weights in figure 4.2 correspond to these quantities.

We conclude this section by stating the problems of inheritance and categorization in terms of the representation language described above. In section 5 we present a solution to these problems based on evidential reasoning. As it turns out, an evidential approach leads to a natural interpretation of exceptions, defaults, and multiple inheritance in the presence of conflicting information.

4.4 The inheritance and categorization problems in terms of the representation language.

The inheritance problem may be restated as follows:

Given: $\Theta = \langle \mathbb{C}, \Phi, \lambda, \Lambda, \#, \delta, \ll \rangle$,

$C \in \mathbb{C}$, $P \in \lambda(C)$, and

V-SET $= \{V_1, V_2, \ldots V_n\}$, a subset of $\Lambda(P)$

Find: $V^* \in$ V-SET, such that among members of V-SET, V^* is the *most likely value* of property P for concept C. In other words, find $V^* \in$ V-SET such that, for any $V_i \in$ V-SET, the best estimate of $\#C[P,V^*] \geq$ the best estimate of $\#C[P,V_i]$'s.

The inheritance problem where C = APPLE, P = has-color, V-SET = {RED, BLUE, GREEN}, may be paraphrased as:

Is the color of an apple more likely to be red, green or blue?

The standard way of posing the above problem would be:

What is the color of an apple?

Our specification is similar to the standard specification of the inheritance problem but with two crucial differences. In the evidential framework, the question is always posed with respect to a set of possible answers, and the correctness of an answer lies in it being the *most likely* answer relative to the set of possible answers. The requirement that a set of possible answers be specified is not a restriction, because by default one may always assume V-SET to be $\Lambda(P)$. Furthermore, the measure of correctness employed by us is a generalization of the standard measure of correctness; if a particular member of V-SET is the *correct* answer (i.e. the *only* possible answer) and the rest are *false* (i.e. are *impossible*), then the correct answer will be found. However, in situations where no definite answer exists, the solution presented in section 5 will ascertain the *most likely* answer. We argued in section 1.2 that except in some trivial situations, one does not have sufficient knowledge to *deduce* the correct answer, and one must resort to judgements based on likelihoods. The next section specifies the precise sense in which an answer is judged as most likely.

The categorization problem is posed as follows:

Given: $\Theta = \langle \mathbb{C}, \Phi, \lambda, \Lambda, \#, \delta, \lessdot \rangle$, and

i) an explicit enumeration of possible answers, i.e. a set of concepts, C-SET $= \{C_1, C_2, \ldots C_n\}$, where either all members of C-SET are Types, or all members of C-SET are Tokens.

ii) a description consisting of a set of property value pairs, i.e., a set DISCR $= \{ [P_1,V_1], [P_2,V_2], \ldots [P_m,V_m] \}$, such that, for each $[P_j,V_j] \in$ DISCR,

$$P_j \in \cap_{C \,\in\, \text{C-SET}} \lambda(C), \text{ and } V_j \in \Lambda(P_j).$$

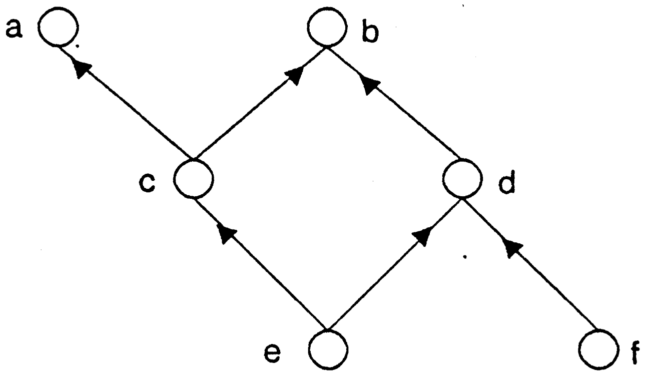
In other words, a property mentioned in the description should apply to every concept in C-SET, and the values specified for these properties should be appropriate.

Find: C € OSET such that *relative* to the concepts specified in C-SET, *C* is the *most likely* concept described by DISCR.

If C-SET = {APPLE, GRAPE}, DISCR = {[has-color, RED], [has-taste, SWEET]} then the categorization may be paraphrased as follows:

"It is red in color and sweet in taste", is it more likely to be an apple or a grape"?

Once again, the two distinguishing features of the problem statement are that a choice is to be made from among a set of possible answers, and the most likely answer is to be found. In the simple case where only one of the possible answer matches the given description, the evidential treatment finds the appropriate answer. But even when the description is partial, and many possibilities exist, the present formulation finds the answer that best matches the given description. The detailed solution appears section 5.5.



The partial ordering is:

{(c,a) (e,a) (c,b) (d,b)
(e,b) (f,b) (e,c) (e,d) (f,d)}

The directed graph on the left is the ordering graph for the partial ordering defined on the right.

FIGURE 4.1 Ordering graph

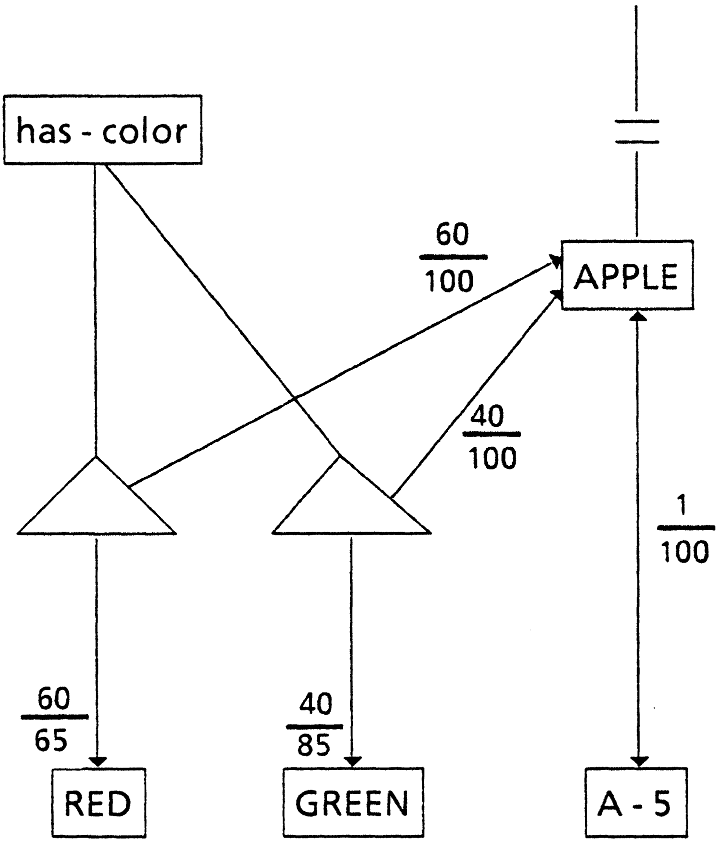


FIGURE 4.2 Partial representation of the example in section 4.2

Chapter 5

An evidential treatment of inheritance and categorization

5.1 The problem of combining evidence

It was argued in section 1.2 that an agent is often compelled to make decisions and perform actions based on partial knowledge. Imagine a situation in which an agent *has to* choose and perform an action from amongst a set of mutually exclusive actions: a_1, a_2, ...a_n. It is assumed that the conditions under which each action is appropriate are known to the agent. For example, he may be aware that that if condition α_j is true then action a_i is appropriate. If the agent had complete knowledge about the world, he could have deduced which of the conditions: α_1, α_2, ... α_m, is true and thence, performed the appropriate action. However, the agent may only have partial knowledge about the world, and therefore, he may be unable to deduce which condition is true.

In the above situation, the best strategy that the agent can adopt is to take into account whatever he knows and determine as to which condition is *most likely* to be true, and thereafter perform the action indicated by this condition.

As an example, assume that an agent has the following knowledge:

$\mathbb{C}$ = {APPLE, GRAPE, RED, GREEN, SWEET, SOUR}

λ(APPLE), λ(GRAPE) = {has-color, has-taste}
λ(RED), λ(GREEN), λ(SWEET) and λ(SOUR) = $\varnothing$

Λ(has-color) = {RED, GREEN}; Λ(has-taste) = {SWEET, SOUR}

#(APPLE) = 100;
δ(has-color, APPLE):

$\#$APPLE(has-color, RED) = 60; $\#$APPLE(has-color, GREEN) = 40

δ(has-taste, APPLE):

$\#$APPLE(has-taste, SWEET) = 70; $\#$APPLE(has-taste, SOUR) = 30

$\#$(GRAPE) = 50

δ(has-color, GRAPE):

$\#$GRAPE(has-color, RED) = 5; $\#$GRAPE(has-color, GREEN) = 45

δ(has-taste, GRAPE):

$\#$GRAPE(has-taste, SWEET) = 30; $\#$GRAPE(has-taste, SOUR) = 20

$\lessdot$: $\emptyset$

Given the above knowledge, a rational agent would have no difficulty in guessing the most probable identity of an object given one of its property values. The following table lists the choices we expect him to make:

Description of object	most likely identity of object
has-color RED	APPLE (60 apples v/s 5 grapes)
has-color GREEN	GRAPE (45 grapes v/s 40 apples)
has-taste SWEET	APPLE (70 apples v/s 30 grapes)
has-taste SOUR	APPLE (30 apples v/s 20 grapes)

In each case the agent chooses the more likely concept on the basis of the available information. For example, there are 30 instances of sour apples as against 20 of sour grapes. Hence, in the absence of any other information, a rational agent will believe that a sour object is more likely to be an apple than a grape.

But how should a rational agent decide the most probable identity of an object given a description specifying multiple property values such as "red and sweet". In other words, given that an agent possess sufficient knowledge to decide if something is an apple or a grape when given that it is red or when given that it is sweet, then how should he use this knowledge to decide whether something is an apple or a grape when given that it is both red and sweet.

The above question embodies a central problem of evidential reasoning namely, the problem of combining evidence. This section presents a solution to this problem based on the notion of maximum entropy, a notion that is fundamentally related to information theory and statistical mechanics. (See [Jaynes 57] and [Jaynes 78]) and also bears resemblance to statistical methods developed for error estimation and hypothesis testing.

5.1.1 Problem formulation

The information about apples and grapes given above, may be expressed in the form of matrices as shown in figure 5.1. The rows of the two matrices correspond to the different values of the property has-taste while the columns correspond to the different values of the property has-color. The numbers at the end of each row(column) represent the number of instances of the concept that have the appropriate value of taste(color).

In general, an agent's knowledge about a concept A may be represented as an n-dimensional matrix where $n = | \lambda(A) |$. Each dimension of the matrix corresponds to an applicable property and the extent of a dimension is given by the number of distinct values the property may have. The #A[P,V]'s appear as marginals or the sums of hyper-rows and hyper-columns.

The internal matrix elements may be used to specify the number of instances of the concept that have the appropriate combination of property values. For instance, the top left element of the APPLE (GRAPE) matrix in figure 5.1 indicates the number of instances of apples (grapes) that are *both red in color and sweet in taste.*

The problem of guessing the identity of an object given its color as well as its taste would be trivial if the agent knew the internal matrix elements. For example, to decide whether a red and sweet object is an apple or a grape, the could simply compare the top left elements of the two matrices in figure 5.1 and choose the concept that has the higher value.

However, if the agent does not know the internal matrix elements the best that he can do is find the most probable estimates of these on the basis of the available information and use these estimates to reason about the world. In section 5.1.2 we show how the most probable estimate may be found.

5.1.2 Computing the most probable configuration

The general 2-dimensional case, may be represented as shown in figure 5.2.

The matrix represents the concept A, and

R_i = #A[property 1, i^{th} value of property 1]

C_j = #A[property 2, j^{th} value of property 2]

$N = \#A = \Sigma_{i=1,n} R_i = \Sigma_{j=1,m} C_j$

a_{ij} = #A[property 1, i^{th} value of property 1][property 2, j^{th} value of property 2]

i.e. the number of instances of A having the i^{th} value for property 1 ***and*** the j^{th} value for property 2.

The a_{ij}'s are unknown and need to be determined on the basis of N, R_i's and the C_j's.

Let a ***macro-configuration*** be a specification of all the a_{ij}'s. Our goal is to find the most probable macro-configuration indicated by the following information:

Vi (i = l,n) $Z_{j=:Um}$ (ay) = $R_{\{}$

vj 0 = ljn) 2 $_{i = l,n}$ (ay) = Cj

^{2}i=l,n;j = l,m (^{a}ij) = ^{N}

The problem of finding the most probable macro-configurations may be recast as follows:

Consider distributing N *distinct* objects into a 2-dimensional array of cells.

Then a macro-configuration specifies the *number of objects* placed in each cell Let a *micro-configuration* be the complete specification of the result of such a distribution In other words, for each cell a micro-configuration specifies the *objects* that are placed in the cell.

Let the *number* of objects placed in the ijt*[1] cell be given by ay. It follows that there is a many to one mapping from the space of micro-configuration to the space of macro-configurations.

Let a micro-configuration be termed *feasible* if it satisfies the constraints imposed by row sums and column sums. Then:

Given his knowledge, an agent has *no basis* for assuming that a particular feasible micro-configuration is more probable than some other feasible micro-configuration and the *only rational* assumption he can make is that all feasible micro-configuration are equally probable.[1]

In view of the above assumption the *most probable macro-configuration* will be that which results from the *greatest number of feasible micro-configurations.*

1. This is in essence the principle of indifference or the principle of insufficient reason first stated by Bernoulli in 1713.

If w denotes the number of placements resulting in a configuration then w is given by:

$$\mathbf{w = N!/n_{i=lfll} j_{=lfm}\ ay!}$$

{number of ways of dividing N distinct objects into n*m groups of a^, a_{12} ... a_{nm} each.}

One may now maximize w *subject to the constraints:*

Vi (i = l,n) $Z \wedge_{m}$ (ay) = Rj

Vj (j = l,m) $2_{i=l,n}$ (ay) = Cj

^{2}i=l,n;j = l,m (^{a}ij) = N

in order to find the most probable macro-configuration.

We show that for the above maximization problem:

vij (i = l,n; j = l,m):

$$ay = R_i\text{-}Cj/N$$

satisfies the condition of maximality.

The maximization is performed by using a technique that involves introducing the constraints as Lagrange multipliers.

Keeping in view the presence of product terms in w we work with log w instead of w and use the Stirling's approximation for factorials; if n » 0 then log n! $\approx$ n*log n - n where log is the natural logarithm.

tog w = log N! - $2_{i=1,n}$. Hm log ay!

log w = N *log N - N - $2_{i=1,n}$·*ylm* (ay*log ay - ay)   Stirling's approximation

Differentiating the above and introducing Lagrange multipliers obtained by differentiating the constraint expressions:

$\lambda_o * (\Sigma_{i=1,n;j=1,m} (a_{ij}) - N) +$

$(\Sigma_{i=1,n} (\lambda_i * (\Sigma_{j=1,m} (a_{ij}) - R_i)) +$

[illegible]

we have:

Vij (i = l,n; j = l,m)

d log w/d ay = - [ay/ay + log ay -1] + Xo+ Xi + 5j

Setting the derivative to 0, we have

Vij (i = l,n; j = l,m)   log ay = X o+ Xi + 5j

i.e.   vij (i = l,n; j = l,m)   ay = exp(X o+ Xi + 8j)   -I-

Substituting exp(Xo) = ao, exp(Xi) = ai, vi (i = l,n); and

exp(*8])* =)8j, vj (j = l,m) in I we have,

Vij (i = l,n; j = l,m)   ay = ao*ai*8j   - II-

Substituting -II- into the constraint equations to solve for ao, afs and /?j's, we have,

$\forall i \ (i=1,n) \ \alpha_o * \alpha_i * \Sigma_{j=1,m} \beta_j = R_i$ -III-

Vj(j = l,m)ao*j8j*$2_{i=1,n}$ai = Cj   -IV-

$\alpha o^{*}\Sigma_{i=1,n;j=1,m}{}^{ai}\wedge \; = \; N$

i.e. $ao\text{-}(2_{i=1,n}\ ai) * (2_{j=1,m}\ ft) = N$ -V-

We now show that:

$ao = l/(a*b*N)$, «i = a*R[vi (i = l,n); and /?j = b*Cj vj (j = l,m)

are a set of constant solutions, where a and b are arbitrary constants

This may be shown by substituting the above values of ao. ai's and ft's into equations III, IV and V.

Notice that $2_{j=1m}ft = b\text{-}2_{j=s1m}\ Cj = b*N$

and $2_{i=1n}ai = a*2_{i=1n}R_i = a*N$

Hence, Vi (i = l,n) $ao'oi*\ 2j_{=Un}fl = (l/(a*b*N))*a*R_i*b*N = R;$

as required by equation III,

Similarly, Vj (j = l,m) $ao«/3j«2_{i=ln}\ ai = (l/(a*b*N))*b*Cj*a*N = Cj$

as required by equation IV,

Finally, $ao*(2_{i=ln}\ ai) * (2_{j=Un}\#) = (l/(a*b*N))*a*N*b*N = N$.

as required by equation V.

This shows that the suggested solutions of ao. ai and *fa* are indeed correct.

Hence, vij (i = l,n; j = l,m), the values of ay are given by:

$ay = ao*ai«Sj = (l/(a*b*N))*a*R_i*b*Cj = R\wedge Cj/N.$

In other words the best estimate of:

$$\#A[P_1,V_1][P_2,V_2]$$

that an agent may make on the basis of $\#A[P_1,V_1]$ and $\#A[P_2,V_2]$ is given by:

$$\#A[P_1,V_1] * \#A[P_2,V_2] / \#A$$

The above result can be extended to higher dimensions. The result is analogous to the result for 2-dimensions and is given by:

$$a_{ijk...} = A_i * B_j * C_{k...} / (N^{n-1})$$

where N equals the total number of objects, n equals the number of dimensions in the array, A_i, B_j, C_k ... denote the sums of hyper-rows or hyper-columns and $a_{ijk...}$ is the most probable number of objects in the $ijk...^{th}$ cell of the array.

The above result will be referred to as the ***best estimate rule*** and may be restated as follows. Based on the knowledge of:

$$\#A[P_1,V_1],\ \#A[P_2,V_2],\ ...\ \text{and}\ \#A[P_n,V_n],$$

the ***best*** (i.e. the most probable) estimate of,

$$\#A[P_1,V_1][P_2,V_2]\ ...\ [P_n,V_n]$$

is given by:

$$(\Pi_{i=1,n} \#A[P_1,V_i]) / \#A^{n-1}$$

Referring back to the example about apples and grapes - the result derived above implies that a rational agent would believe that the most probable way in which the instances of apples and grapes could be distributed is given by the matrices shown in figure 5.3. Thus, he will identify a "red and sweet" object to be an apple as there are probably 42 apples meeting this description as against only 3 grapes.

In conclusion, it must be pointed that the best estimate rule is based on the agent's state of knowledge and provides a relative measure of likelihood instead of an absolute probability value. The following section discusses how the approach adopted in this section compares with the Dempster Shafer evidence theory and Bayes' rule for conditional probabilities.

5.2 Relation of the maximum entropy approach to some other treatments of uncertainty.

5.2.1 Relation to the Dempster-Shafer theory

The Dempster-Shafer (DS) evidence theory [Shafer 76][Garvey et al. 81][Barnett 81] suggests an evidence combination rule that is currently in vogue in aritificial intelligence. One can show that a straight forward application of the DS rule for evidence combination does not produce the correct results - for the kinds of problems we wish to solve. It is shown that the DS result agrees with the best estimate rule if one assumes that the frequency (i.e. the prior probability) of all concepts is the same.

A simple example illustrates the point. Consider the information about apples and grapes as given in section 5.1.

If one wishes to use the DS rule to decide whether a green and sour object is an apple or a grape one would essentially proceed as follows:

One would treat each property value as a source of evidence. The evidence provided by green and sour will be:

$E(\text{Apple} \mid \text{green}) = 40/85, \qquad E(\text{Grape} \mid \text{green}) = 45/85$

E(Apple | sour) =30/50, E(Grape | sour) = 20/50

Applying the DS rule for evidence combination we get:

E(Apple | green & sour) = (40/85)*(30/50) and

E(Grape | green& sour) = (45/85)*(20/50)

{The above is a simplified account of the actual steps using DS theory. We have focused on the essentials. In particular, we have not normalized the quantities because we are only interested in a relative measure.}.

Comparing the evidence for Apples and Grapes we have

E(Apple | green & sour) : E(Grape | green & sour) equals,

$$(40/85)*(30/50) : (45/85)*(20/50) = \mathbf{4:3}$$

and the decision is in favor of Apple.

However, on the basis of the given information, the best (most probable) estimate of the number of green and sour Apples is 12 and that of green and sour Grapes is 18. (See figure 5.3). Hence the appropriate ratio is:

$$12 : 18 = \mathbf{2:3}$$

and the decision is in favor of Grapes!

It is not difficult to locate the reason for this discrepency. Given that one is only interested in making comparisons, the ratio of the relative likelihood of two concepts A and B using the DS rule is given by:

$$\Pi_{i=1,n}(\#A[P_i,V_i] \;/\; \#B[P_i,V_i]) \qquad \mathrm{DS}_{ratio}$$

However, the best estimate rule gives the ratio as:

$$[\Pi_{i=1,n}(\#A[P_i,V_i] / \#B[P_i,V_i])] * (\#B / \#A)^{n-1}$$

which may be restated as:

$$DS_{ratio} * (\#B / \#A)^{n-1} \qquad \text{-Eq-I-}$$

If one were to assume $\#A = \#B$, or in effect that *all concepts have the same prior probability*, then the DS rule and the best estimate rule become equivalent.

One might suggest that by including an additional source that provides evidence about the prior probabilities of apples and grapes, one might be able to correct the DS result. However, an examination of Eq-I will indicate that the problem is more complex. In order to make the DS_{ratio} the same as that obtained by the best estimate result one will have to multiply it by the factor:

$$(\#B / \#A)^{n-1}$$

But introducing an evidential source to account for the prior probability only introduces the factor $\#A/\#B$, which acts in the wrong direction.

5.2.2 Relation to Bayes' rule

It can be shown that in case none of the internal entries in the concept matrix are known, then conventional Bayesian inference used in conjunction with certain independence assumptions produces the same results as those obtained by the maximum entropy formulation. However, as discussed below, the maximum entropy formulation is more attractive as it offers a precise means for formalizing and conceptualizing the problem of estimating unknown probabilities; independence assumptions do not appear as ad hoc assumptions but can be justified as the only meaningful assumptions one may make *given the state of one's knowledge.* If additional information suggesting dependence is available (i.e. if a few internal entries are known), the maximum entropy formulation exploits this information in a consistent manner.

One can pose the problem raised in section 5.1 about deciding whether an object is an apple or a grape, in terms of conditional probabilities. The decision task amounts to determining which of the two probabilities is greater:

Pr(Apple | red & sweet) or Pr(Grape | red & sweet)

In the above context Pr(A | B) denotes the conditional probability of A given B. "Apple", and "Grape" are to be read as "x is an apple", and "x is a grape" respectively, while "red" and "green" are to be read as "x has the value RED for property has-color", and "x has value SWEET for property has-taste" respectively.

The following are some of the probabilities known to the agent by virtue of the information given in section 5.1:

Pr(Apple) / Pr(Grape) = #APPLE / #GRAPE;

Pr(red | Apple) = #APPLE[has-color, RED] / #APPLE;

Pr(red | Grape) = #GRAPE[has-color, RED] / #GRAPE

Pr(sweet | Apple) = #APPLE[has-taste, SWEET] / #APPLE;

Pr(sweet | Grape) = #GRAPE[has-taste, SWEET] / #GRAPE

Using Bayes' rule of conditional probabilities:

Pr(Apple | red & sweet) equals:

Pr(red & sweet | Apple) * Pr(Apple) / Pr(red & sweet)

and

Pr(Grape | red & sweet) equals:

Pr(red & sweet | Grape * Pr(Grape) / Pr(red & sweet)

Hence, Pr(Apple | red & sweet) / Pr(Grape | red & sweet) equals:

(Pr(red & sweet | Apple) * Pr(Apple)) /

(Pr(red & sweet | Grape) * Pr(Grape) - EQ-I

If we make the following *independence assumption*:

$Pr([P_i,V_i] \& [P_j,V_j] \mid C_k) = Pr([P_i,V_i] \mid C_k) * Pr([P_j,V_j] \mid C_k)$ - IA-I

we have: Pr(red & sweet | Apple) = Pr(red |Apple) * Pr(sweet | Apple), and

Pr(red & sweet | Grape) = Pr(red | Grape) * Pr(sweet | Grape)

Substituing these probabilities in EQ-I we have,

Pr(Apple | red & sweet) / Pr(Grape | red & sweet) equals:

(Pr(red | Apple) * Pr(sweet | Apple)) /

(Pr(red | Grape) * Pr(sweet | Grape)) * (Pr(Apple) / P(Grape))

Expressing the probabilities in terms of $\#C[P_i,V_i]$'s and #C's,

Pr(Apple | red & sweet) / Pr(Grape | [red & sweet) equals:

((#APPLE[has-color, RED] / #APPLE) *

(#APPLE[has-taste, SWEET]/#APPLE)) /

((#GRAPE[has-color, RED]/#GRAPE) *

(#GRAPE[has-taste, SWEET]/#GRAPE)) *

(#APPLE/#GRAPE)

which equals:

((#APPLE[has-color, RED] * #APPLE[has-taste, SWEET]) /
(#GRAPE[has-color, RED] * #GRAPE[has-taste, SWEET])) *
(#GRAPE/#APPLE) - EQ-II

as would be computed using the best estimate rule.

Thus, using Bayes' rule in conjunction with the independence assumption IA-I, leads to the same result that would be obtained by the best estimate rule derived in section 5.1.

However, in deriving the best estimate rule we did not make any assumption such as IA-I. Therefore, let us examine, why the two results are identical, and identify the relation between assumption IA-I and the derivation of the best estimate rule.

The independence assumption provides a way of dealing with unknown probabilities. For example, the computation of Pr(Apple | red and sweet) required the knowledge of Pr(red & sweet) | Apple), but since the latter was not known, one applied the independence assumption and assumed that Pr(red & sweet | Apple) was equal to Pr(red | Apple) * Pr(sweet | Apple).

The independence assumption IA-I is equivalent to the independence assumption of the form:

$$Pr(e_i \ \& \ e_j \mid H) = Pr(e_i \mid H) * Pr(e_j \mid H)$$

which in turn is equivalent to the assumption:

$$Pr(e_i \mid e_j \ \& \ H) = Pr(e_i \mid H)$$

In the above, H is a hypothesis, and e_i, e_j are two pieces of evidence.

This form of independence is often assumed in reasoning systems based on Baysean statistics [Hart & Duda 77], but its use is widely criticized as being unreasonable and even unwarranted [Charaiak 83].

The problem with the independence assumption is that it usually appears as an ad hoc assumption, unrelated to the rest of the agent's body of knowledge. Because it is not stated as to how the rest of the agent's knowledge affects the validity of this assumption, it remains unclear as to when the assumption is warranted and when it is not.

In the derivation of best estimate result, the agent's knowledge consisted of terms of the form #A's and #A[Pj,Vjj's for different concepts A and property value pairs [Pi,Vi]'s. *None of the quantities* #A[Pj,,VJ[Pj,Vj] *were known.* The best estimate result established that *in this specific situation,* the most probable estimate of the unknowns:

#A[P_i,V_i][P_j,Vj]'s

is given by,

tfApV^] * #A[Pj,Vj] / #A

It is only in this specific condition that the best estimate result matches the result obtained by using Bayes' rule in conjunction with the independence assumption. Indeed, *it is only under this condition* that the independence assumption is justified.

The maximum entropy approach offers a precise way of formulating and conceptualizing the problem of estimating missing probabilities. In essence, if Pr(X) is not known then the principle of maximum entropy prescribes that we use all the information at hand, and compute the probability of each possible value of Pr(X), and choose that value of Pr(X) which is most probable.

Under the maximum entropy approach, each piece of information is viewed as a constraint that is used to determine which micro-configurations of the domain are feasible and which are not. Next, the most probable macro-configuration of the domain is ascertained by finding a macro-configuration that is supported by the largest number of feasible micro-configurations. Once such a macro-configuration has been ascertained, the unknown probabilities are computed using this macro-configuration. If there is no dependence or correlation in the underlying information then the result obtained by the maximum entropy principle does agree with the result obtained by using Bayes' rule in conjunction with the independence assumption. However, if additional information indicating dependence is available, it is incorporated in the derivation of the most probable configuration as an additional constraint and the result reflects this dependence. For example, if the agent knows one of the $\#A[P_i,V_i][P_j,V_j]$'s, i.e. with respect to figure 5.2, if the agent knows one of the internal matrix elements, then the constraints used during the maximization of:

$$w = N!/\Pi_{i=1,n;j=1,m}\ a_{ij}!$$

are altered. Without loss of generality, let the agent know that $a_{11} = \alpha$. Then, the constraints are:

$$\Sigma_{j=2,m}\ (a_{1j}) = R_1 - \alpha$$

$$\Sigma_{i=2,n}\ (a_{i1}) = C_1 - \alpha$$

$$\forall i\ (i = 2,n)\ \Sigma_{j=1,m}\ (a_{ij}) = R_i$$

$$\forall j\ (j = 2,m)\ \Sigma_{i=1,n}\ (a_{ij}) = Cj$$

$$\Sigma_{i=1,n;j=1,m}\ (a_{ij}) = N$$

The most probable configuration is given by:

$$\forall j\ (j = 1,m)\ a_{1j} = C_j * (R_1 - \alpha)/(N - C_1)$$

$\forall i\ (i = 1,n)\ a_{i1} = R_i * (C_1 - \alpha)/(N - R_1)$ and

$\forall i,j\ (i \neq 1$ and $j \neq 1)\ a_{ij} =$

$$R_i * C_j * [N + \alpha - (R_i + C_j)] / [(N - R_i) * (N - C_j)]$$

As should be obvious, if many internal elements are known then the above computations get complex; the implications of this are discussed in section 7.

5.3. Evidential inheritance

This section develops an evidential theory of inheritance based on the result derived in section 5.1. Let us recall the definition of the inheritance problem.

Given: $\Theta = \langle \mathbb{C}, \Phi, \lambda, \Lambda, \#, \delta, \lessdot \rangle$,

$C \in \mathbb{C}$, $P \in \lambda(C)$, and

V-SET $= \{V_1, V_2, \ldots V_n\}$, a subset of $\Lambda(P)$

Find: $V^* \in$ V-SET, such that among members of V-SET, V^* is the *most likely value* of property P for concept C. In other words, find $V^* \in$ V-SET such that, for any $V_i \in$ V-SET, the best estimate of $\#C[P,V^*] \geq$ the best estimate of $\#C[P,V_i]$'s.

In order to solve the inheritance problem, an agent needs to know the $\#C[P,V_i]$'s. If the agent knows $\delta(C,P)$ then the solution to the problem becomes trivial. However, if the agent does not know $\delta(C,P)$, then he has to compute the most probable (best) estimates of the $\#C[P,V_i]$'s based on his knowledge about other concepts in the conceptual structure. It is this latter case - where $\delta(C,P)$ is unknown, that is of interest to us. In the rest of this section, we develop a theory of inheritance which prescribes how $\#C[P,V_i]$'s may be computed based on knowledge available at concepts that lie above C in the ordering induced by $\ll$ on $\mathbb{C}$. The resulting theory can deal with exceptions as well as multiple inheritance situations. In particular, it is shown that the results apply to the "multiple views" organization proposed in section 3.5.

The section begins by considering the simplest case of inheritance (direct inheritance) and progressively considers more complex cases that require combining evidence from multiple sources (multiple inheritance).

5.3.1 Direct inheritance

Given two concepts C and B, and a property P such that:

$C \ll B$, $\delta(C,P)$ is not known, but $\delta(B,P)$ is known,

then *in the absence of any other information*, #C[P,V]'s are *best estimated* by:

$$\#B[P,V] * (\#C/\#B)$$

For example, if 40% of fruits are red, then in the absence of any other information, except that apples are a subtype of fruits, the best estimate of the percentage of red apples is 40%.

A proof follows directly from an application of the best estimate rule, and is a special case of the proof given in section 5.3.4.1

Direct inheritance is analogous to the notion of direct inference in statistical inference.

5.3.2 Generalization of direct inheritance

As stated above, the direct inheritance rule applies only when the agent's knowledge is limited to knowing that $C \ll B$, and $\delta(B,P)$ is known. A useful generalization of direct inheritance would be to include the case where it is known that B has other offsprings besides C. This may be done if we make the following assumption about the conceptual structure of the agent:

Well-formedness rule 1 (WFR-cs-1)

The agent stores (or remembers) all distributions that are important to him and that cannot be estimated accurately on the basis of information available at concepts higher up in the conceptual structure. In other words, if the knowledge of $\#C[P,V_i]$'s is important to the agent, and if $\#C[P,V_i]$'s are significantly different from those that would be obtained by inheritance (i.e. if $\#C[P,V_i]$'s are exceptional), then the agent must store $\delta(C,P)$.

In the absence of such an assumption, the generalization of direct inheritance could lead to erroneous results, as is illustrated by the following example.

Let apples and grapes be two subtypes of fruits. Assume that there are 150 fruits, 100 of which are apples and 50 of which are grapes. Also assume that 40% of the fruits are red (i.e. 60 fruits are red). In the absence of any other information, the best estimate that one can make is that there are 40 red apples, and 20 red grapes. (This follows from direct inheritance). However, if it is also known that 50% (i.e. 50) apples are red then it follows that only 10% (i.e. 5) grapes are red, and this differs significantly - the error is 300%, from the estimate obtained by direct inheritance (40% or 20). WFR-1 would ensure that *if it is important* for the agent to predict the number of red grapes, then in the above situation he would store δ(has-color, GRAPE).

It should be noted that there are two preconditions associated withWFR-1. The rule suggests that $\delta(C,P)$ be stored if the information encoded in $\delta(C,P)$ is useful to the agent, ***and*** the estimate obtained by direct inheritance are significantly different. In the above example, if we reverse the situation so that instead of knowing that 50% apples are red, it is known that 50% (i.e. 25) grapes are red, then the correct estimate of the number of red apples is 35%. Now 35% may not be considered significantly different from the estimate obtainable by direct inheritance, (direct inheritance leads to 40%, hence the error is only 12.5%), and the agent may not store δ(has-color, APPLE), even though it may be important for the agent to predict the number of red apples.

5.3.3 Principle of relevance

Given a concept C and a property P ∈ λ(C), a concept B is ***relevant*** to C with respect to P, if and only if:

i) C ≪ B,

ii) δ(B,P) is known and

iii) there exists no concept A (distinct from C and B) such that δ(A,P) is known and C ≪ A ≪ B.

We will often need to refer to the set of concepts that are relevant to C with respect to P, and this set will be referred to as Γ(C,P).

Figure 5.4 illustrates the above definition. Notice that a concept could be relevant to itself with respect to some property P. Furthermore, it follows that for such a concept, there exists no other concept relevant to it with respect to P.

The ***principle of relevance*** states that:

Given a concept A and a property P such that P ∈ λ(A) and δ(A,P) is not known, and if there is *only one* concept (say B) that is relevant to A with respect to property P, then the *best estimate* of δ(A,P) may be directly inherited from B, and all other information may be ignored.

Example: Suppose an agent knows that apples are a subtype of fruits, delicious is a subtype of apples, 40% fruits are red and 60% apples are red. Given this information, the best estimate of the percentage of red delicious is 60%, based on the more specific information about apples.

The principle of relevance appears as the reference class problem in statistical inference [Kyburg 83] and also corresponds to the inferential distance ordering in [Touretzky 84].

5.3.4 Multiple inheritance

This section presents a solution to a restricted class of the multiple inheritance problem. However, the scope of the solution is general enough to apply to a conceptual structure that has the "multiple views" organization described in section 3.5.

The existence of multiple relevant concepts requires that evidence from many sources be combined. The goal is to apply the result derived in section 5.1 and arrive at a evidence combination rule. We begin by observing that for the purpose of inheriting the property P of concept C, we need to consider only those concepts D_i that are above C in the conceptual hierarchy, and for which $\delta(D_i,P)$ is known. This is the motivation behind introducing the following definition:

Given a concept C and a property $P \in \lambda(C)$:

$\mathbb{C}/C,P$, the ***projection*** of $\mathbb{C}$ with respect to C and P, is defined as:

$$\mathbb{C}/C,P = \{ x \mid x \in \mathbb{C} \text{ and } \delta(x,P) \text{ is known and } C \lessdot x \}$$

Figure 5.5 illustrates the above definition.

In the ensuing discussion we will always work with the projected structure.

We now consider two cases of multiple inheritance. In the first case, all the relevant concepts have a common parent and a direct application of the best estimate rule suffices. In the second and the more general case, a common parent does not exist In this case, evidence from relevant concepts is combined by progressively moving up the conceptual structure and repeatedly applying the result derived in the common parent situation. This is done until a common ancestor of all the relevant concepts is reached. It is shown that by doing so evidence from all relevant concepts gets assimilated.

5.3.4.1 Multiple inheritance: the simple case

Let 6 = <C, <S, A, A, #, 5, <> be the conceptual structure of an agent If the ordering induced on C/C,P by < is such that there exists a unique reference concept (say Q) for r(C,P), that is *also the parent of all members of* r(C,P), then:

the best estimate of #C[P, Vj] / #C[P,V_q] is given by:

<n_{kaU}(#B_k[P,Vj] / #B_k[P,V_q])) * (#O[$P_f V_q$] / tfQPWj])^{11}-^{1} - EQ-I

In the above expressions, Vj and V_q are members of A(P), B^'s are members of r(C,P), and n = |r(C,P)|.

(Notice that if there is only one relevant concept B, then Q may be taken to be B itself, and the above result reduces to the direct inheritance result)

In other words, if an estimate of #C[P,Vj] is required solely for the purpose of comparing it with estimates of other #C[P,V_q]'s, then it suffices to compute:

<^{n}k=u$^{\#B}$k(^{p}< ^{v}jD / tfalP.VjF^1 - EQ-II

The result is summarized in figure 5.6, and illustrated with the help of an example in figure 5.7.

Proof: We prove the above result by first establishing the following result:

Given a set V, and n sets $S_1, S_2, \ldots S_n$, each $S_k \subseteq V$;

let W denote $\cap_{k=1,n} S_k$

Then, the best estimate of #W based on all available information is given by:

$$\#W = \#V * \Pi_{k=1,n}(\#S_k/\#V)$$

$$= (\Pi_{k=1,n}\#S_k) / \#V^{n-1} \qquad \text{- EQ-III}$$

Proof: The problem of estimating #W is a special case of the problem of determining the most probable configuration. The correspondence is as follows: Each S_k may be treated as a 2-valued property applicable to V. For each $v \in V$, if $v \in S_k$ then the value of the property S_k for v equals 1 otherwise it equals 0. Hence, finding the best estimate of #W is identical to finding the best estimate of the number of elements in V that have the value 1 for each of the properties S_1 through S_n. The result of section 5.1 states that that the best estimate is given by EQ-III above.

After having established EQ-III we may now establish EQ-I by recognizing that:

If D be the set $\cap_{k=1,n} B_k$, then #D[P, Vj] is best estimated by:

$$\#D[P, Vj] = (\Pi_{k=1,n}\#B_k[P, Vj]) / \#Q[P,Vj]^{n-1} \qquad \text{- EQ-IV}$$

; using EQ-III above and identifying V in EQ-III with Q[P,Vj] in EQ-IV and each S_k in EQ-III with B_k[P, Vj] in EQ-IV.

Although, the set D may not be explicitly defined as one of the concepts in the domain, it may be introduced for the purpose of this analysis[2].

With reference to D, one may estimate $\#C[P, V_j]$ by:

$$\#C[P, V_j] = \#D[P, V_j] * (\#C / \#D)$$

Similarly, one may estimate $\#C[P, V_q]$ by

$$\#C[P, V_q] = \#D[P, V_q] * (\#C / \#D)$$

Thus, the best estimate of $\#C[P, V_j] / \#C[P, V_q]$ is given by:

$$\#C[P, V_j] / \#C[P, V_q] = (\#D[P, V_j] * (\#C / \#D)) / (\#D[P, V_q] * (\#C / \#D))$$

$$= \#D[P, V_j] / \#D[P, V_q] \qquad \text{- EQ-V}$$

It is easy to see that EQ-V is equal to EQ-I by using the result derived in EQ-IV.

This ends the proof.

2. Any arbitrary set may not be introduced in this manner. However, D is a well defined set in that it is the intersection of existing sets.

With reference to derivation of EQ-III above, suppose it is known that there are a number of sets $V_1, V_2, \ldots V_m$ such that, for each V_i, each $S_k \subseteq V_i$.

In this case there would be m different estimates of $\#W$, one for each V_i:

$$(\Pi_{k=1,n} \#S_k) / \#V_i^{n-1}$$

Which of these is the best estimate of $\#W$? In other words, which of the V_i's should be used as the reference set for finding the most likely intersection of S_k's. This is essentially the reference class problem mentioned in section 5.3.3.

If one of the V_i's - say V^*, is such that $V^* \subseteq V_i$, for each V_i $1 \leq i \leq m$, then the best estimate of $\#W$ is the one that is computed using V^* as the reference set. That is, the best estimate of $\#W$ is given by:

$$(\Pi_{k=1,n} \#S_k) / (\#V^*)^{n-1}$$

Notice that in the derivation of EQ-IV, V was identified with $\Omega[P,V_j]$ and each S_k with $B_k[P,V_j]$, where B_k's are the members of $\Gamma(C,P)$. This explains the motivation behind the requirement on Ω that it be a parent of all the members of $\Gamma(C,P)$.

5.3.4.2 Multiple inheritance: the more complex case

This section deals with a more complex multiple inheritance situation, one in which a single common parent of the members of $\Gamma(C,P)$ does not exist. The result of this section states that:

If the ordering induced on $\mathbb{C}/C,P$ by $\lll$ is such that there exists a unique reference concept Ω for $\Gamma(C,P)$ and there is a unique path from each $B_i \in \Gamma(C,P)$ to Ω, then the best estimate of

$$\#C[P,V_j] / \#C[P,V_q]$$

is computed by the following algorithm:

$$\#C[P,V_j] / \#C[P,V_q] := \text{BEST-ESTIMATE}(\Omega,P,V_j) \; / \; \text{BEST-ESTIMATE}(\Omega,P,V_q);$$

The function BEST-ESTIMATE(ξ, φ, ν) operates on the tree induced by $\lll$ on $\mathbb{C}/C,P$ and is as follows:

Function BEST-ESTIMATE (ξ, φ, ν) ; returns real

{ξ is a concept, φ is a Property and ν is a value}

If $\xi \in \Gamma(C,P)$ then BEST-ESTIMATE := $\#\xi[\varphi,\nu]$

else BEST-ESTIMATE :=

$$\#\xi[\varphi,\nu] \;*\; \Pi\; (\text{BEST-ESTIMATE}(\xi_i,\varphi,\nu)\; / \#\xi[\varphi,\nu])$$

{The product is taken over all the sons (i.e. the ξ_i's) of ξ}

Explanation: The requirement placed on the conceptual structure is such that the ordering induced by $\lessdot$ on $\mathbb{C}$/C,P results in a graph that includes a tree with B_i's as its leaves and Ω as its root. The situation is depicted in figure 5.8 The above algorithm combines evidence provided by each $B_i \in \Gamma(C,P)$ by repeated application of the result derived in section 5.3.4.1. Specifically, the result computed at each concept ξ that lies on a path from Ω to a B_i, combines the evidence provided by all the B_i's that lie below ξ. This ensures that the evidence provided by all the B_i's has been combined when the result is computed at Ω.

The second and the third arguments of the function BEST-ESTIMATE (B.E. in brief), remain fixed during recursive invocations of the function. Hence, in analyzing the computation performed by B.E. (Ω,P,V_j), we will refer to all recursive call by specifying only the first argument. The first argument indicates the concept at which the function is currently being evaluated. The invocation trace of the function corresponds to a tree traversal, where the tree consists of the concept Ω, and all concepts that lie on paths from Ω to a B_i.

It can be proved that the B.E.$(\Omega$,P,V1) / B.E.$(\Omega$,P,V2) equals:

the best estimate of #C[P,V1] / the best estimate of #C[P,V2].

As the first step of the proof we derive the following lemma:

Lemma-MI: B.E. $(\Omega$,P,Vj) computes the best estimate of:

#D[P,Vj], where D is the set $\cap_{i=1,n} B_i$, such that, for all i, $1 \leq i \leq r$, B_i's are the members of $\Gamma(C,P)$.

Inductive proof for Lemma-MI:

Base case:

At the lowest level (level 0) of the ordering defined by $\ll$ and $\mathbb{C}/C,P$, each concept ξ, is a member of $\Gamma(C,P)$. Hence, the result computed by B.E.(ξ), where ξ is some $B_i \in \Gamma(C,P)$, is $\#\xi[P,V_j]$.

If ξ is a concept at level 1 then each of its offsprings will be a relevant concept. Let these be: B_1, B_2 ... B_r. Then it follows that B.E.(ξ) computes:

$$(\Pi_{i=1,r} \# B_i[P, V_j]) \ / \ \#\xi[P,V_j]^{r-1}$$

Hence, the result computed by B.E.(ξ), for all concepts ξ at level 1, corresponds to the best estimate of $\#D[P, V_j]$, where D is the set $\cap_{i=1,r} B_i$, and B_1, B_2, ... B_r are the offsprings of ξ. (cf. section 5.3.4.1 EQ-IV).

This establishes the base case (for level 1 nodes).

Induction step

Inductive hypothesis: Assume that the result computed by B.E.(ξ) for each ξ at some intermediate level k, corresponds to the best estimate of $\#D[P,Vj]$, where D is the set $\cap_{i=1,r} B_i$, such that, for all i, $1 \leq i \leq r$, B_i's are *descendants* of ξ.

We now show that the result computed by B.E.(ξ) for each ξ at level k+1, also corresponds to the best estimate of $\#D[P,Vj]$, where D is the set $\cap_{i=1,r} B_i$, such that, for all i, $1 \leq i \leq r$, B_i's are *descendants* of ξ.

Let ξ be a concept at level k+1, and let ξ_t, t =1, q, be its offsprings. As each offspring will be a concept at level k, it follows from the inductive hypothesis that:

$$\text{B.E.}(\xi_t) = \#D_t[P,Vj],$$

where D_t is the set $\cap_{i=a_t,b_t}$ Bj, such that, for all i, at $\leq$ i $\leq$ bt, Bj's are *descendants* of $£_t$.

Then the result computed by B.E.(£), for some £ at level k + 1, is:

$(\cap_{t=1,q} \#D_t[P,Vj])$ / ^^P.VjF[1]

But by the derivation of EQ-IV (cf. section 5.3.4.2), the above expression is the best estimate of #D[P,Vj] where D is given by:

$\cap_{t=1,q} (\cap_{i=a_t,b_t} B_i)$

which may be rewritten as:

$B_{a1} \cap B_{a1+1} \sim\cap B_{b1} \cap B_{a2} \cap B_{a2+1} - \cap B_{b2} \ldots \cap B_{aq} \cap B_{aq+1} - \cap B_{bq}$

which is the intersection of all B^s that are descendants of £.

This concludes the inductive step.

But by definition, Q is the ancestor of all members of r(C,P), hence, B.E. (Q,P,Vj) computes the best estimate of:

#D[P,Vj], where D is the set $\cap_{i=1,n}$ Bj, such that, for all i, 1 $\leq$ i $\leq$ r, Bj's are the members of r(C,P).

This concludes the inductive proof.

Hence, B.E.(Q,P,V1) / B.E.(Q,P,V2) equals:

#D[P,V1] / #D[P,V2]

where D is the set $\cap_{i=1,n}$ $B_{i?}$ where, $\{B_{15}\ B_2, \ldots B_n\}$ = r(C,P).

This establishes Lemma-MI.

By virtue of the arguments leading to EQ-V (cf. 5.3.4.2):

#D[P,V1] / #D[P,V2] equals:

the best estimate of #C[P,V1] / the best estimate of #C[P,V2].

This establishes the correctness of the inheritance algorithm.

5.3.5. Evidential inheritance: a summary

Problem statement:

Given: $\Theta = \langle \mathbb{C}, \Phi, \lambda, \Lambda, \#, \delta, \ll \rangle$,

$C \in \mathbb{C}$, $P \in \lambda(C)$, and

V-SET = {V1, V2, ... Vn}, a set of values of P, i.e. V-SET $\in \Lambda(P)$

Find: V* $\in$ V-SET, such that among members of V-SET, V* is the *most likely value* of property P for concept C. In other words, the best estimate of #C[P,V*] $\geq$ the best estimate of #C[P,V_i]'s.

Solution:

i) Find $\Gamma(C,P)$

ii) If there exists a unique reference concept Ω for $\Gamma(C,P)$ and there is a unique path from each $\gamma \in \Gamma(C,P)$ to Ω in the ordering diagram defined by $\ll$ and $\mathbb{C}$/C,P then:

Find V* such that for all i

$$\text{BEST-ESTIMATE}(\Omega, C, V^*) / \text{BEST-ESTIMATE}(\Omega, C, V_i) \geq 1$$

(Direct inheritance and the case where $\delta(C,P)$ is known are special cases of the above result)

The condition specified in step ii) is not unduly restrictive. The conceptual structure described in section 3.5 is more restricted than the conceptual structure permitted by the above condition. Thus, the solution developed above applies to the "multiple views" organization in which the Type structure defined over Tokens consists of *several* distinct taxonomies. Recall that in such an organization, each Token may have several parents, and hence, multiple relevant concepts.

In particular, condition ii) does NOT require all concepts in $\mathbb{C}$ to be organized as a tree; if this were the case, multiple inheritance situations would not even arise. Notice that, the condition stated in step ii) does not require the ordering graph defined by $\mathbb{C}$ and $\leqslant$ to have tree like properties, it only requires that the ordering graph defined by $\mathbb{C}/C,P$ and $\leqslant$, have a tree like structure.

Consider the concepts in figure 5.9a. Assume that the property has-belief with values pacifist, and non-pacifist, applies to the concepts shown in the figure, and that the agent has stored δ(C, has-belief) for any concept C that is enclosed in a dark box. The concepts are embedded in a multilevel hierarchy ans some of the concepts have multiple relevant concepts. Yet the most likely property-values of these concepts may be obtained via inheritance. In particular, the question: "Is Dick a pacifist or a non-pacifist", may be posed as an inheritance problem. Figure 5.9b shows the projected structure for this question. The issue of Dick's pacifism would be resolved by combining evidence from the two relevant concepts Quaker and Republican at the reference concept Person, and the answer would depend on which of the two:

1) ((#QUAK[has-belief, PAC] * #REPUB[has-belief, PAC]) /
#PERSON[has-belief, PAC])

2) ((#QUAK[has-belief, NON-PAC] * #REPUB[has-belief, NON-PAC]) /
#PERSON[has-belief, NON-PAC])

is greater. If 1) is greater then it is more likely that Dick is a pacifist, and if 2) is greater then it is more likely that Dick is a non-pacifist If 1) and 2) are the same then there is nothing in the agent's body of knowledge to make the above decision, and he may make an arbitrary choice. The answers still depend on the information available at the concepts Quaker, Republican, and Person, just as was the case in the simpler example considered in section 5.3.4.1, figure 5.7. The distribution information available at Christian node plays no role in arriving at a decision.

5.3.6 The role of numbers in the theory

The representation language specified in section 4.1 required the specification of absolute numbers. It was assumed that the agent knew the values of #C for each member of C. It was also assumed that he knew the values of #C[P,V], for Cs and Fs for which 8(C,P) was defined.

However, an important characteristic of the theory of inheritance developed above is that none of the calculations require the knowledge of absolute numbers. All the necessary numeric information is embodied in the following ratios each of which lies in the interval [0,1}:

a) For all C and P such that 5(C,P) is known, ratios of the form

$$\#C[P,V_i] \;/\; \#D[P,V_i] \text{ and } \#C[Py_{\{}] \;/\; \#V\hat{}$$

where, D is a parent of C in the ordering induced by <^ on C/C,P and Vys are possible values of P.

b) For all pairs of concepts C and D such that D is a parent of C in the ordering induced by < on C, the ratios:

$$\#C / \#D$$

5.4 Comparison of the evidential approach to other approaches to inheritance

This section illustrates some of the advantages of adopting an evidential framework for formalizing inheritance in conceptual hierarchies. It was shown in section 1.5 that none of the existing formalizations of inheritance: those based on first-order logic, Reiter's default logic [Etherington and Reiter 83], or Touretzky's principle of inferential distance ordering [Touretzky 84], can deal with multiple inheritance adequately. This section considers a simpler situation that involves exceptions but does not involve multiple inheritance. It is shown that in this situation, a system based on default logic, or one based on Touretzky's principle of inferential distance ordering, would *yield* intuitively correct result, but these solutions are not backed by any model-theoretic account of the reasoning that underlies these systems. I am not considering systems based on first-order logic because the formalization of inheritance based on first-order logic is clearly *not intended* to handle exceptions, whereas Touretzky, as well as Etherington and Reiter claim to deal with the problem of inheritance with exception.

We consider a very simple example (henceforth, the penguin example) and solve it using the evidential approach. Next, we compare the representation of the problem and the interpretation of the answer suggested by my approach with that suggested by Fahlman [Fahlman 79], Tourtezky, and Etherington and Reiter (henceforth referred to as E &R.)

Let us assume that an agent's knowledge consists of the beliefs that 80% of the birds fly, penguins and robins are subtypes of birds, and while none (0%) of the penguins fly, 95% of the robins do. Let us posit that there is a property has-mode-of-transport (has-md-trnspt in brief) with one of its values being FLY, and let us gather all other values into a single value -FLY. The other concepts are BIRD, ROBIN and PENGUIN.

If the agent is told that "Tweety is a bird[1]', and asked "does Tweety **fly",** how would he respond? The problem is easily analyzed in terms of evidential reasoning. As the agent does not know whether Tweety can fly or not, he considers all possible configurations of the world that are consistent with his knowledge. On the basis of the information available about Tweety, it could be ***any*** bird; it could be a penguin, a robin, (and if there were any other subtypes such as ravens and ostrich, it could also be any one of these). Based on the information that is available to him, an agent may compute the ratio of the number of feasible configurations of the world in which Tweety can fly, to the number of feasible configurations in which Tweety cannot fly. This ratio equals 80 : 20, and is given by:

#BIRD[has-md-trnspt FLY]

Σ_X #BIRD[has-md-trnspt X]

where, X ranges over all other modes of transport, in our case there are only two values FLY and ~FLY.

The ratio 80 : 20 (i.e. 4 : 1), suggests to the agents that it is four times more likely that Tweety is a bird that flies than that Tweety is a bird that does not fly. Therefore, if the agent *has to* make a choice he would choose "Tweety flies".

There are three points that need to be made:

1. The agent is not certain that Tweety flies. He only believes that it is more likely that Tweety flies.

2. The above does not imply any fuzziness about birds that fly or about Tweety. It is not as if Tweety is a 0.8 flier and 0.2 non-flier. Tweety either flies or it does not; it is only the likelihood of Tweety being either of these that is 4 : 1.

3. It is *not assumed* that Tweety is *not a penguin*. (The agent would have

arrived at the same answer even if he had reasoned by cases: considering Tweety to be a penguin or not a penguin ... etc.)

Now consider another situation. The agent has the same information as before about birds and penguins, but this time he is told that Tweety is a penguin. (It does not matter whether he is also told that Tweety is a bird, for this follows from the fact that Tweety is a penguin, and all penguins are birds). In this situation, Tweety could be ***any*** penguin. Given that 0% penguins fly and 100% don't, the ratio of the number of feasible configurations of the world in which Tweety can fly, to the number of feasible configurations in which Tweety cannot fly is 0 : 100. This ratio is given by:

#PENGUIN[has-md-trnspt FLY]

Σ_X #PENGUIN[has-md-trnspt X]

The ratio 0 : 100 (i.e. 0 : 1), tells the agents that it is *impossible* that Tweety flies. Hence, the agent will infer that "Tweety does not fly". Notice that this time the agent is certain of his choice, he knows that that there is no feasible configuration of the world in which Tweety could fly.

Finally, if the agent were told that "Tweety is a Robin" he would choose "Tweety flies", in this case the ratio of the likelihood of "Tweety flies" to "Tweety does not fly", is 95 : 5, i.e. 19 : 1. The agent's response is the same as it was when he was told that "Tweety is a bird", but this time he will be much more confident about his choice.

The above example demonstrates that in each of the three situations, evidential reasoning gives a crisp meaning to what the agent believes about Tweety. The likelihood of Tweety being a flier or not being one, changed in each of the three cases because the agent had different information about Tweety. In each case, the agent found the most likely possibility on the basis of all the information that was available to him, and in all the three situations, the measures of likelihood were based on the same body of a priori knowledge and each of the conclusions were mutually consistent. More importantly, in each case the choices made by the agent can be *justified* in terms of his beliefs about the possible configurations of the world. Notice that there was no need to "cancel" any links, nor to arbitrarily select one of many extensions, neither was it necessary to introduce the notion of non-monotonicity or fuzzy truth values.

Let us now examine how this example would be treated by Fahlman, Touretzky and E & R. The agent's knowledge about the world would be rendered as follows:

Touretzky and Fahlman:

Birds fly.

Penguins are birds.

Robins are birds.

Penguins don't fly.

E & R:

Normally birds fly, but there may be exceptions.

Penguins are always birds.

Robins are always birds.

Penguins never fly.

Figure 5.10 gives the graphical notation associated with these representations. As robins are not exceptional, the information available about the ability of robins to fly would not be represented explicitly in either formalism. Fahlman and Touretzky employ a special CANCEL link to block the effect of the IS-A link between BIRD and FLYER in the context of

PENGUIN. Thus, BIRD IS-A FLYER, PENGUIN IS-A BIRD, but it is not the case that PENGUIN IS-A FLYER. E & R employ a total of 5 different kinds of links, four of these kinds are required in this example (see figure 5.10).

Before we examine the solutions that follow from the above representations, let us point out a basic problem with each of these approaches. The above approaches treat properties and Types as equivalent constructs[3]. Not making a distinction between Types and properties leads to some unusual results; the cancel links employed by Fahlman and Touretzky, and the fact that E & R entertain situations wherein there may be a default IS-A link and also a default ISN'T-A link from A to B, are symptomatic of these confusions.

3. In NETL, Fahlman does make a distinction between properties and Types but his subsequent remarks in the context of cancellation [Fahlman et al. 81] indicate that he thinks that this distinction is not crucial.

Touretzky's proposal is an improvement over Fahlman's NETL. Touretzky has given a precise specification of what he expects the inheritance hierarchies to do, and shown how to *condition* NETL networks so that they perform inheritance in accordance with the principle of inferential distance ordering. However, Touretzky also makes use of CANCEL links, and what follows applies to Fahlman and Touretzky as well. In Fahlman's and Touretzky's scheme one may say: A IS-A B and B IS-A C, and at the same time have a CANCEL link from A to C. This CANCEL link is supposed to imply that A's are not B's. But how can A's be B's and B's be C's and yet A's not be C's? It would be desirable to have a clear semantic account of what this means. As it stands, the CANCEL link appears to be a construct that is an artifact of implementation, required to make the network do the right thing, but without any clear representational import.

Now consider E & R's proposal. This proposal allows one to construct a situation as shown in figure 5.11 (see E & R 83] pp. 107). There is a default IS-A link as well as a default ISN'T-A link from A to B. According to the semantics (?) of these links as suggested in [E & R 83]: this situation is to be understood as follows:

"Normally A's are B's but there may be exceptions, and normally A's are not B's but exceptions are allowed".

The above seems to say nothing meaningful about what A's normally are.

Returning to the penguin example, one may see how each of the proposals would handle the three questions posed about Tweety. If the agent is told "Tweety is a bird", Fahlman's and Tourtezky's systems would infer that "Tweety flies" by moving up the IS-A link from TWEETY to BIRD and thence to FLYER. A similar explanation holds for the case where "Tweety is a robin". However, if the agent is told that "Tweety is a penguin", then the CANCEL link at PENGUIN would block the effect of the IS-A link between BIRD and FLYER.

The explanation of the E & R's system is along similar lines. The strict ISN'T-A link from PENGUIN to FLYER would block the inference that would have been made by the presence of the default IS-A link from BIRD to FLYER. The graphical notation of E & R may be mapped to a set of default rules, and the information related to flying in the penguin example may be translated as follows:

$$\frac{\text{BIRD}(x) : \sim\text{PENGUIN}(x)\ \&\ \text{FLYER}(x)}{\text{FLYER}(x)} \qquad \text{-- dr-1}$$

$$\text{PENGUIN}(x) \Rightarrow \sim\text{FLYER}(x) \qquad \text{- R1}$$

In the situation where it is known that "Tweety is a bird," dr-1 applies and it is inferred that "Tweety flies". However, if it is known that Tweety is a penguin, it may directly be inferred that "Tweety does not fly" by using R-l (notice that in any case, the default rule dr-1 remains blocked).

In each of the above cases the response of the E & R system and that of the Touretzky or Fahlman systems matches our intuitions. However, these formalisms do not provide a model-theoretic account of their inferential behavior. It is not sufficient to have precise rules of inference; the inferences drawn should be justifiable in terms of how they claims the world to be. Default rules and CANCEL links seem to characterize a methodology that amounts to identifying the inferences that are desirable in certain special situations, and explicitly encoding these as default rules or CANCEL links.

Turning to the evidential framework, an attractive feature of the representation is that either a concept is an instance of (subtype 00 another concept or it is not. The < relation specifies this unequivocally. The notion of exception only applies to property values and even here, exceptions do not entail "cancellation" or "blocking" of properties. I claim that the approach presented in this thesis is the only one - within the knowledge representation circles, that handles inheritance in hierarchies with exceptions, in a *justifiable* manner, and offers a model-theoretic account of the conclusions that are drawn.

The approach applies only under conditions stated in section 5.3, but the class of situations handled by this approach subsumes the class of situations handled by Touretzky and Etherington and Reiter. Furthermore, for this class of inheritance problem there exists an extremely efficient implementation on an appropriate parallel computer. Section 6.1 demonstrates that it is possible to encode the evidential knowledge of an agent as a highly parallel (connectionist) network of active elements connected via weighted links that can perform the computations required to solve the inheritance problem in only $O(d)$ time, where d is the length of the longest path in the conceptual hierarchy defined by <

5.5 Evidential categorization

This section develops a solution to the categorization problem based on the results derived in sections 5.1 and 5.3. In terms of the language presented in section 4.1, the problem of categorization may be defined as follows:

Given: $\Theta = \langle \mathbb{C}, \Phi, \lambda, \Lambda, \#, \delta, \ll \rangle$, and

i) an explicit enumeration of possible answers, i.e. a set of concepts, C-SET = $\{C_1, C_2, \ldots C_n\}$, where either all members of C-SET are Types, or all members of C-SET are Tokens.

ii) a description consisting of a set of property value pairs, i.e., a set DISCR = $\{ [P_1,V_1], [P_2,V_2], \ldots [P_m,V_m] \}$, such that, for each $[P_j,V_j] \in$ DISCR,

$$P_j \in \cap_{C \in \text{C-SET}} \lambda(C), \text{ and } V_j \in \Lambda(P_j).$$

In other words, a property mentioned in the description should apply to every concept in C-SET, and the values specified for these properties should be appropriate.

Find: C' $\in$ C-SET such that *relative* to the concepts specified in C-SET, C' is the *most likely* concept described by DISCR.

In order to solve the categorization problem, we need to compute the most likely estimates of:

$$\#C_i[P_1,V_1][P_2,V_2] \ldots [P_m,V_m]$$

for each $C_i \in$ C-SET, and choose C' such that the above estimate for C' is greater than or equal to the estimates of all other members of C-SET.

By virtue of the best estimate result derived in section 4, the best estimate of:

based on the knowledge of #qPi,V1], #qp2,V2], ••• #qPm»VnJ is:

$$(\Pi_{j=1,m}\ \#C_i[P_j,V_j]) \ / \ (\#C_i)^{m-1}$$

Hence, the primary step in solving the categorization problem is the estimation of #Cj[Pj,Vj], for each *Cx* € C-SET and each Pj mentioned in DISCR.

The categorization problem is more complex than the inheritance problem for, whereas inheritance involves only a single concept and a single property, categorization involves multiple concepts and multiple properties. In this section we identify conditions under which the solution to the categorization problem remains computationally simple. Section 5.1 considers the case where for each *C[* € C-SET, and each Pj mentioned in DISCR, there is only one concept relevant to *Cx* w.r.t. Pj. Section 5.5.2 considers the more complex case where multiple relevant concepts exist It is shown that under suitable restrictions, even these cases may be handled with ease. Finally, section 5.5.3 evaluates the results of sections 5.5.1 and 5.5.2 with reference to the conceptual structure described in section 3.5, and demonstrates that there is a natural fit between these results and the "multiple views[11] organization proposed in section 3.5.

5.5.1 Unique relevant concepts:

If there exists a unique concept By, that is relevant to *C[* w.r.t Pj, then by direct inheritance, (section 5.3.1), the best estimate of #CJ[PJ,VJ] is given by:

$$\#C_i[P_j,V_j] \ = \ \#B_{ij}[P_j,V_j] \ * \ (\#C_i \ / \ \#B_{ij})$$

Therefore, if DISCR is such that for each Pj mentioned in DISCR, there exists a unique concept By relevant to *Cx* w.r.t Pj, then the best estimate of:

$\#C_i[P_1,V_1][P_2,V_2] \ ... \ [P_m,V_m]$

which is given by: $(n_{j=1m} \ \#q[Pj,Vj]) \ / \ (\#C_i)^{m-1}$

equals: $(n_{j=} i_{iin}(\#B_ij[Pj,Vj] \ * \ \#q \ / \ \#By)) \ / \ (\#C_i)^{m"1}$

which in turn equals : $q \ * \ nj_{=1?m}(\#By[Pj,Vj] \ / \ \#By)$

If the above holds for every member of C-SET, i.e. if each q € C-SET is such that for each Pj mentioned in some [Pj,Vj] € DISCR, there exists a unique concept By relevant to q w.r.t to Pj, then the categorization problem may be solved as follows:

i) For each q € C-SET, and each Pj mentioned in some [Pj,Vj] € DISCR, find By, relevant to q w.r.t. Pj.

ii) For each q € C-SET, compute: $q \ * \ nj_{=1m}(\#By[Pj,Vj] \ / \ \#By)$

where Pj, $1 \leq j \leq m$, are properties mentioned in DISCR.

iii) Choose a C for which the quantity computed in ii) has the highest magnitude.

Note that, there is no requirement that the By's be distinct, i.e. it is possible that By = By for some $1 \leq i,k \leq n$, and $1 \leq j,l \leq m$.

5.5.2 **Multiple** relevant concepts

If there exist multiple concepts By\ By% ... By^, that are relevant to q w.r.t. Pj, i.e. r(q,Pj) = { By^1, By^2, ... By^ }, and if there exists Qy such that, Qy is the unique reference concept of r(q,Pj), and it is also the reference concept of all subsets of r(q,Pj), then the best estimate of #q[Pj,Vj] is given by:

$\#C_i[Pj,Vj] \ = \ \#nBy^k[Pj,Vj] \ * \ \#q \ / \ \#nBy^k,$

EQ-I

where $\cap B_{ij}^k$ refers to: $\cap_{k=1,q_{ij}} B_{ij}^k$; the class obtained by intersecting all the B_{ij}^k's.

Because Ω_{ij} is not only a reference concept for $\Gamma(C_i,P_j)$ but also for all subsets of $\Gamma(C_i,P_j)$, it may be treated as the "parent" of all the B_{ij}^k's - the members of $\Gamma(C_i,P_j)$, for the purpose of computing $\#\cap B_{ij}^k[P_j,V_j]$. Thus, EQ-III derived in section 5.3.3.1 applies and,

$$\#\cap B_{ij}^k[P_j,V_j] = (\Pi_{k=1,q} \#B_{ij}^k[P_j,V_j]) / (\#\Omega_{ij}[P_j,V_j])^{q_{ij}-1} \quad \text{- EQ-II}$$

The expression for $\#C_i[P_j,V_j]$ given in EQ-I above, includes the factor $\#\cap B_{ij}^k$; the size of the set obtained by intersecting all B_{ij}^k's. It is not possible to estimate $\#\cap B_{ij}^k$ without taking into account information about $\delta(B_{ij}^k,P_x)$ for all properties P_x that apply to B_{ij}^k's - *even those that are not mentioned in* DISCR.

However, if one makes the simplifying *approximation*, that:

$$\#\cap B_{ij}^k = (\Pi_{k=1,q_{ij}} \#B_{ij}^k) / (\#\Omega_{ij})^{q_{ij}-1} \qquad \text{- APPROX-1}$$

then, by substituting EQ-II in EQ-I and employing APPROX-1 we have that:

$$\#C_i[P_j,V_j] =$$

$$\#C_i * (\#\Omega_{ij} / \#\Omega_{ij}[P_j,V_j])^{q_{ij}-1} * (\Pi_{k=1,q} \#B_{ij}^k[P_j,V_j] / \#B_{ij}^k)$$

--EQ-III

The above expression depends only on the properties and property values mentioned in DISCR. The expression may be further simplified by imposing the following conditions on the conceptual structure:

COND-1: For each P_j there exists a unique Ω_j such that for all $1 \leq i \leq n$,

$\Omega_{ij} = \Omega_j$. In other words, Ω_j is the common reference concept for all $\Gamma(C_i,P_j)$, $C_i \in$ C-SET.

COND-2: The number of concepts relevant to C_i w.r.t P_j is the same for all $C_i \in$ C-SET, i.e. $|\Gamma(C_1,P_j)| = |\Gamma(C_2,P_j)| \ldots = |\Gamma(C_n,P_j)|$. Alternately, $q_{1j} = q_{2j} \ldots = q_{nj} = q_j$

If the conceptual structure satisfies the above two conditions, then in the expression for $\#C_i[P_j,V_j]$ given in EQ-III (this section), the term:

$$(\#\Omega_{ij} \ / \ \#\Omega_{ij}[P_j,V_j])^{q_{ij}-1}$$

is identical for all $C_i \in$ C-SET. This is because for all $1 \leq i \leq n$, COND-1 implies that $\Omega_{ij} = \Omega_j$, and COND-2 entails that $q_{ij} = q_j$.

For the purpose of categorization, it is only required to make a relative comparison the magnitudes of $\#C_i[P_1,V_1][P_2,V_2] \ldots [P_m,V_m]$ for different C_i's. Given that the computation of $\#C_i[P_1,V_1][P_2,V_2] \ldots [P_m,V_m]$ for each C_i equals:

$$(\Pi_{j=1,m} \ \#C_i[P_j,V_j]) \ / \ (\#C_i)^{m-1}$$

which only involves a product of $\#C_i[P_j,V_j]$'s, it follows that it is not necessary to compute $\#C_i[P_j,V_j]$'s using the full expression given by EQ-III; it suffices to compute:

$$\#C_i * (\Pi_{k=1,q_j} \ \#B_{ij}{}^k[P_j,V_j] \ / \ \#B_{ij}{}^k) \qquad \text{-EQ-IV}$$

which is derived from EQ-III by ignoring the term common to all $\#C_i[P_j,V_j]$'s.

Consequently, for the purpose of categorization, it is only necessary to compute the expression:

$$(\Pi_{j=1,m}(\ \#C_i * (\Pi_{k=1,q_j} (\#B_{ij}{}^k[P_j,V_j] \ / \ \#B_{ij}{}^k)))) \ / \ (\#C_i)^{m-1} \qquad \text{-EQ-V}$$

for each $C_i \in$ C-SET.

EQ-V may be simplified to yield:

$$\#C_i * \Pi_{j=1,m;k=1,q_j}(\#B_{ij}^k[P_j,V_j] / \#B_{ij}^k) \qquad \text{- EQ-VI}$$

5.5.3 Categorization in the "multiple views" organization

In this section we examine how the categorization problem may be solved if the conceptual structure has the "multiple views" organization described in section 3.5. In particular, we show how the results of sections 5.5.1 and 5.5.2 apply in this situation.

There are two kinds of categorization problems; one in which members of C-SET are all Types and the other in which they are all Tokens. We refer to these as Type categorization and Token categorization, respectively.

Type categorization

In the "multiple views" organization described in section 3.5, all Types are organized into a strict taxonomy, and there is a unique path from a Type to the root of the conceptual structure. Hence, each Type C_i may have at most one concept relevant to it w.r.t. a property $P_j \in \lambda(C_i)$.

However, the existence of at least one relevant concept is guaranteed by the fact that if $P_j \in \lambda(C_i)$ then, if ω is the leaf of the ontological tree for which $C_i \ll \omega$, then $\delta(\omega,P_j)$ is known. (C.f. section 4.2 WFR-mv-1).

Hence, each Type C_i has exactly one concept relevant to it w.r.t. any $P_j \in \lambda(C_i)$. Consequently, the results of section 5.5.1 directly apply to the case of Type categorization in the "multiple views" organization described in section 3.5.

Token categorization

In case of token categorization, all members of C-SET are Tokens, and hence they may have multiple parents. Consequently, each $C_i \in$ C-SET may have multiple relevant concepts w.r.t. a property P_j.

It was shown in section 5.5.2 that even if there exist multiple relevant concepts, the categorization problem may be solved by computing the relatively simple expression given in EQ-VI, provided conditions: COND-1 and COND-2 are satisfied.

COND-1 requires that for a given P_j mentioned in DISCR, the reference concept for $\Gamma(C_i,P_j)$, be the same for all C_i, members of C-SET. In the context of the "multiple views" organization, COND-1 may be satisfied if we place the following well-formedness constraint on the manner in which the categorization problem is posed:

WFR-cat-1: C-SET should be such that, all members of C-SET are descendants of a *single* leaf of the ontological tree. It follows that all members of C-SET belong to the same ontological type.

If WFR-cat-1 holds, then there exists a ω such that ω is a leaf of the ontological tree and for all $C_i \in$ C-SET, $C_i \ll \omega$. Let H_1, H_2, ...H_q be q views defined with ω as their root. Each C_i may have at most one parent in each view H_k, and hence, there can be at most one concept $B_{ij}{}^k$, that is relevant to C_i w.r.t. P_j, and that lies in the view H_k. In other words, there may be multiple concepts relevant to C_i w.r.t. P_j - but at most one per view defined over C_i. Consequently, each relevant concept lies in a different view, and hence, Ω_{ij} the reference concept for $\Gamma(C_i,P_j)$ in the ordering defined by $\ll$ on $\mathbb{C}/C_i,P_j$, cannot lie within any of these views, but must lie at or above ω. But $\delta(\omega,Pj)$ is known and hence ω is the reference concept for $\Gamma(C_i,P_j)$. It follows that ω will be the reference concept for $\Gamma(C_i,P_j)$, for all $C_i \in$ C-SET. This satisfies COND-1.

COND-2 requires that the number of concepts relevant to C_i w.r.t P_j, be the same for all $C_i \in$ C-SET. In the context of the "multiple view" organization, this condition would be satisfied if the conceptual structure were such that:

WFR-mv-2: If for some Token C_i, there exists a concept $B_{ij}{}^k$, relevant to C_i w.r.t. P_j, and $B_{ij}{}^k$ lies in the view H_k (refer figure 5.5.1), then for *every* Token C_i that lies below H_k, there exists a concept in H_k that is relevant to C_i w.r.t. P_j.

The above well-formedness rule requires that if the distributions for property P_j are stored at concepts in some view H_k, then such distributions should be stored at enough concepts in H_k so that for every Token C_i that is under H_k, there exists at least one concept $B_{ij}{}^k$ within H_k, for which $\delta(B_{ij}{}^k, P_j)$ is known (stored).

If the conceptual structure satisfies WFR-mv-2 then, for a given property P_j and view H_k, either each $C_i \in$ C-SET has exactly one concept $B_{ij}{}^k$ relevant to it w.r.t. P_j in view H_k, or each $C_i \in$ C-SET has no concept relevant to it w.r.t. P_j in view H_k. Thus, every $C_i \in$ C-SET will have the same number of relevant concepts w.r.t. any property P_j mentioned in DISCR.

Thus, the result of section 5.5.2 (EQ-VI), applies to the case of Token categorization provided:

a) the conceptual structure is as described in section 3.5 and conforms to the additional well-formedness properties specified by WFR-mv-1 (cf. section 4.2) and WFR-mv-2, and

b) the categorization problem is posed so that C-SET meets the criteria laid out in WFR-cat-1.

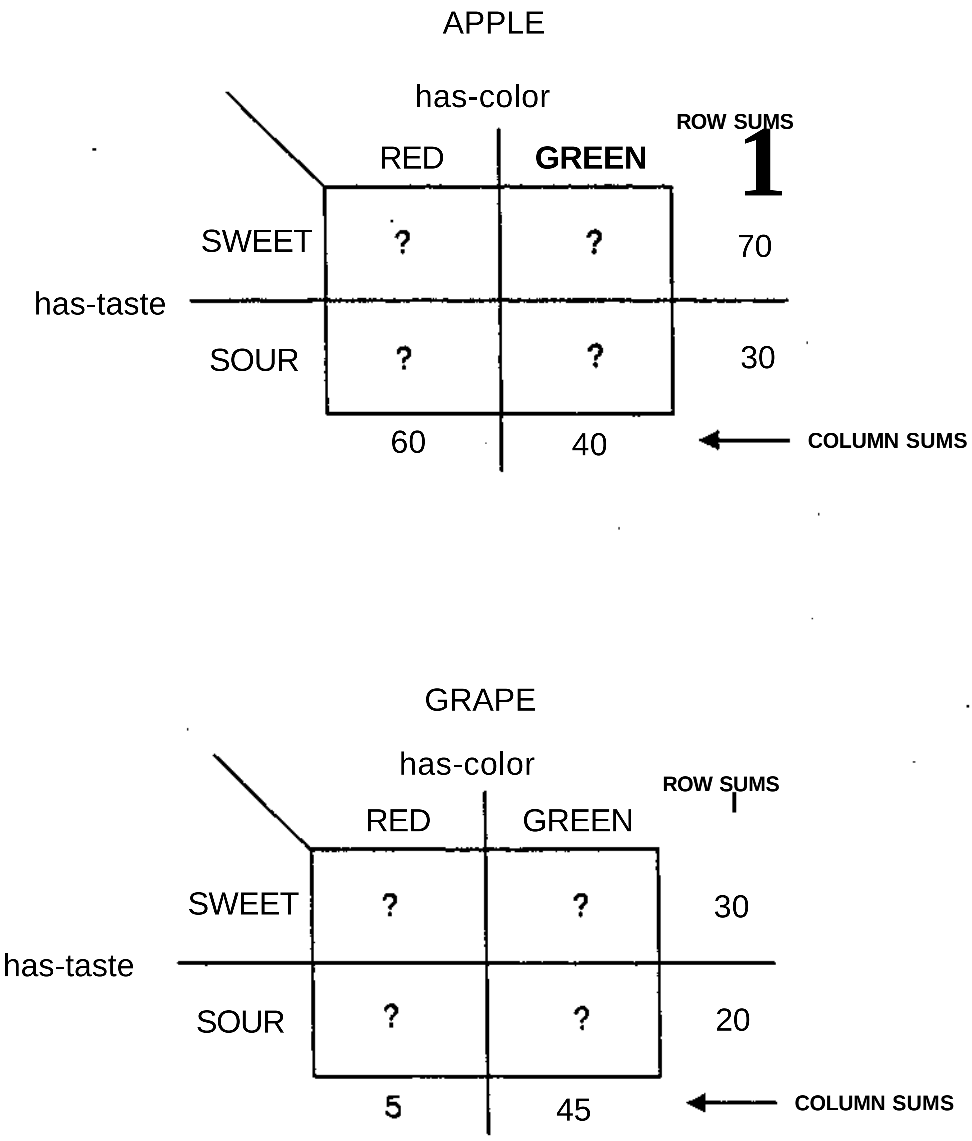


FIGURE 5.1 A matix representation of apples and grapes

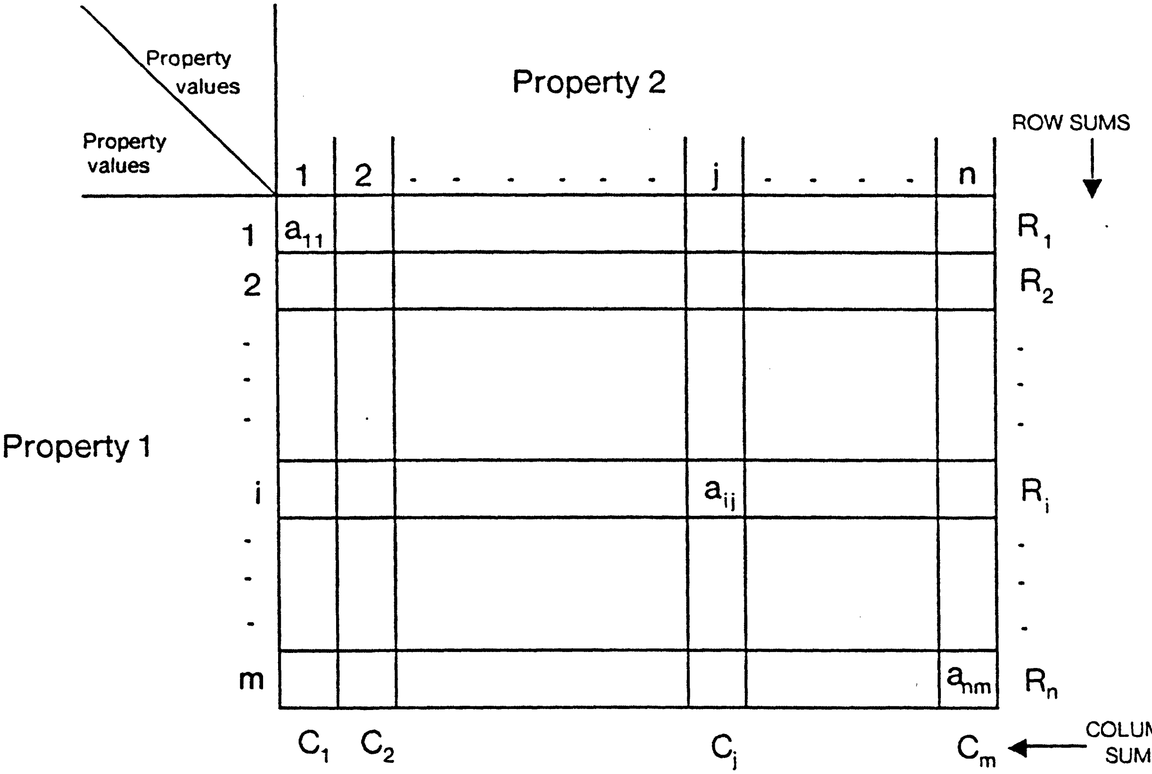


FIGURE 5.2 The general 2-dimensional case

APPLE

has-taste \ has-color	RED	GREEN	ROW SUMS
SWEET	42	28	70
SOUR	18	12	30
COLUMN SUMS	60	40	

N = 100

GRAPE

has-taste \ has-color	RED	GREEN	ROW SUMS
SWEET	3	27	30
SOUR	2	18	20
COLUMN SUMS	5	45	

N = 50

Each matrix element

$$= \frac{\text{ROW SUM} \cdot \text{COLUMN SUM}}{\text{N}}$$

FIGURE 5.3 The most likely distribution of apples and grapes

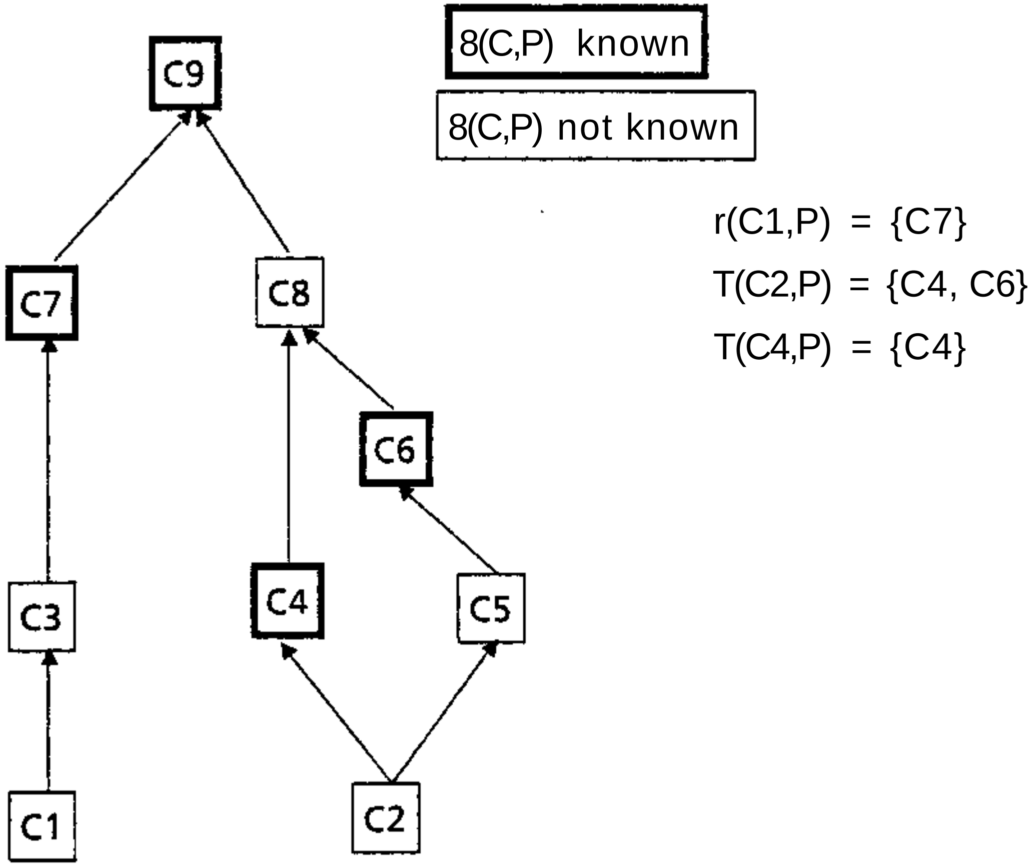


FIGURE 5.4 Relevance

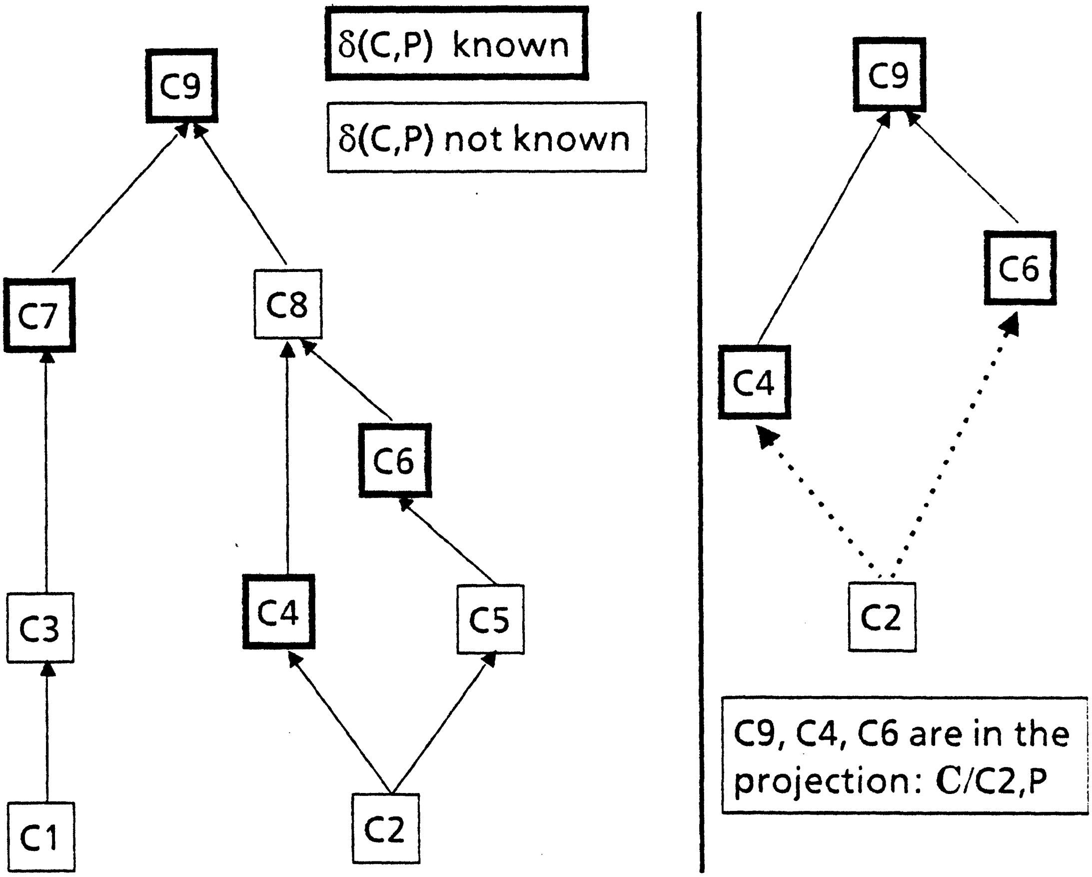
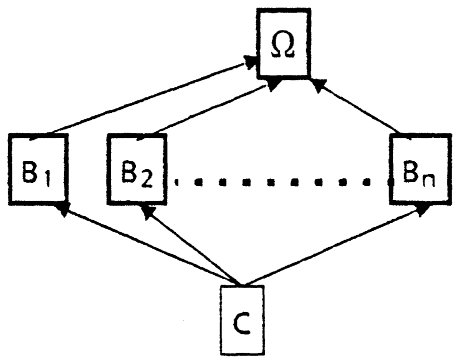


FIGURE 5.5 Projection

Ω is the parent of all B_k's and $\delta(\Omega, P)$ is known



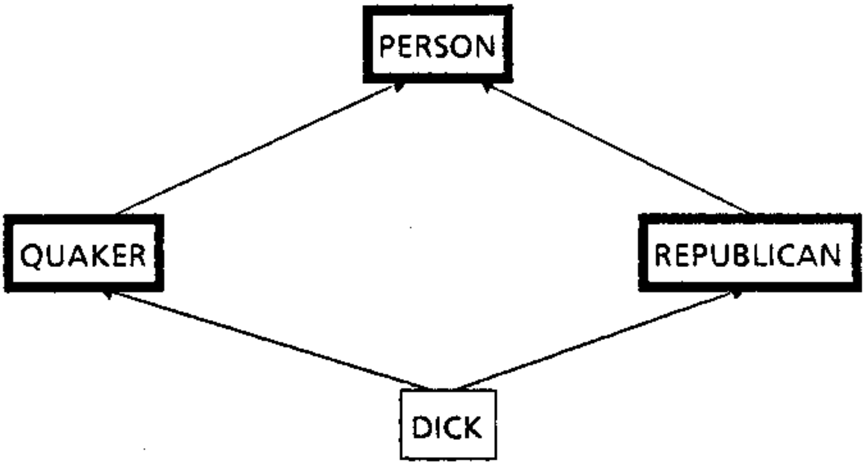
$B_1, B_2, \ldots B_n$ are relevant to C w.r.t. to P.

Ω is the parent of all B_k's and $\delta(\Omega, P)$ is known.

The best estimate of $\#C[P,V_j] \,/\, \#C[P,V_q]$ is given by:

$$\left[\frac{\#\Omega[P, Vq]}{\#\Omega[P, Vj]}\right]^{n-1} \times \prod_{k} \left[\frac{\#B_k[P, Vj]}{\#B_k[P, Vq]}\right]$$

FIGURE 5.6 Multiple inheritance with a common parent



IF

$$\frac{\#QUAK[has\text{-}bel, PAC] \times \#REPUB[has\text{-}bel, PAC]}{\#PERSON(has\text{-}bel, PAC]}$$

$>$

$$\frac{\#QUAK[has\text{-}bel, NON\text{-}PAC] \times \#REPUB(has\text{-}bel, NON\text{-}PAC]}{\#PERSON(has\text{-}bel,NON\text{-}PAC]}$$

then DICK is pacifist.

FIGURE 5.7 Simple case of multiple inheritance

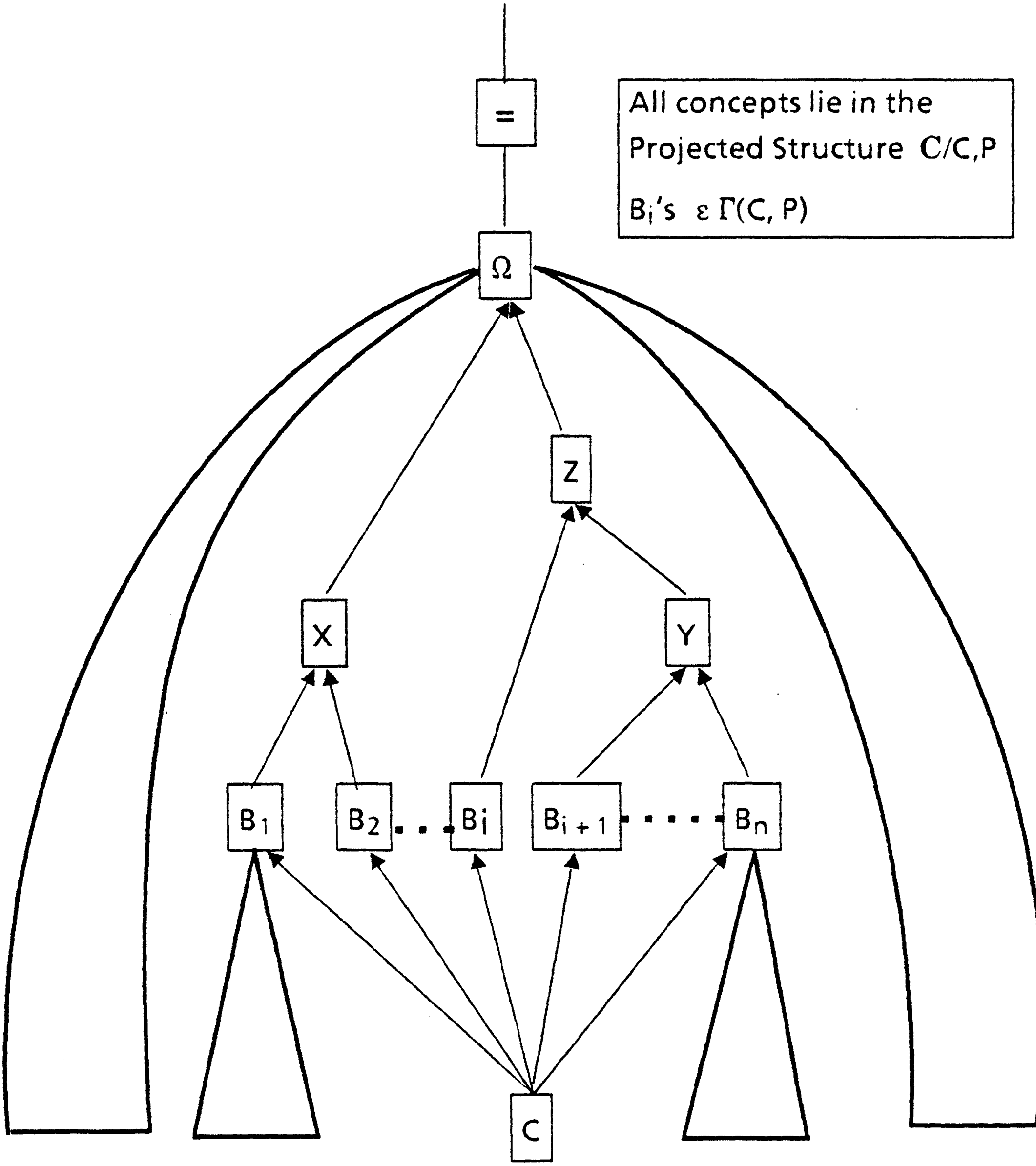
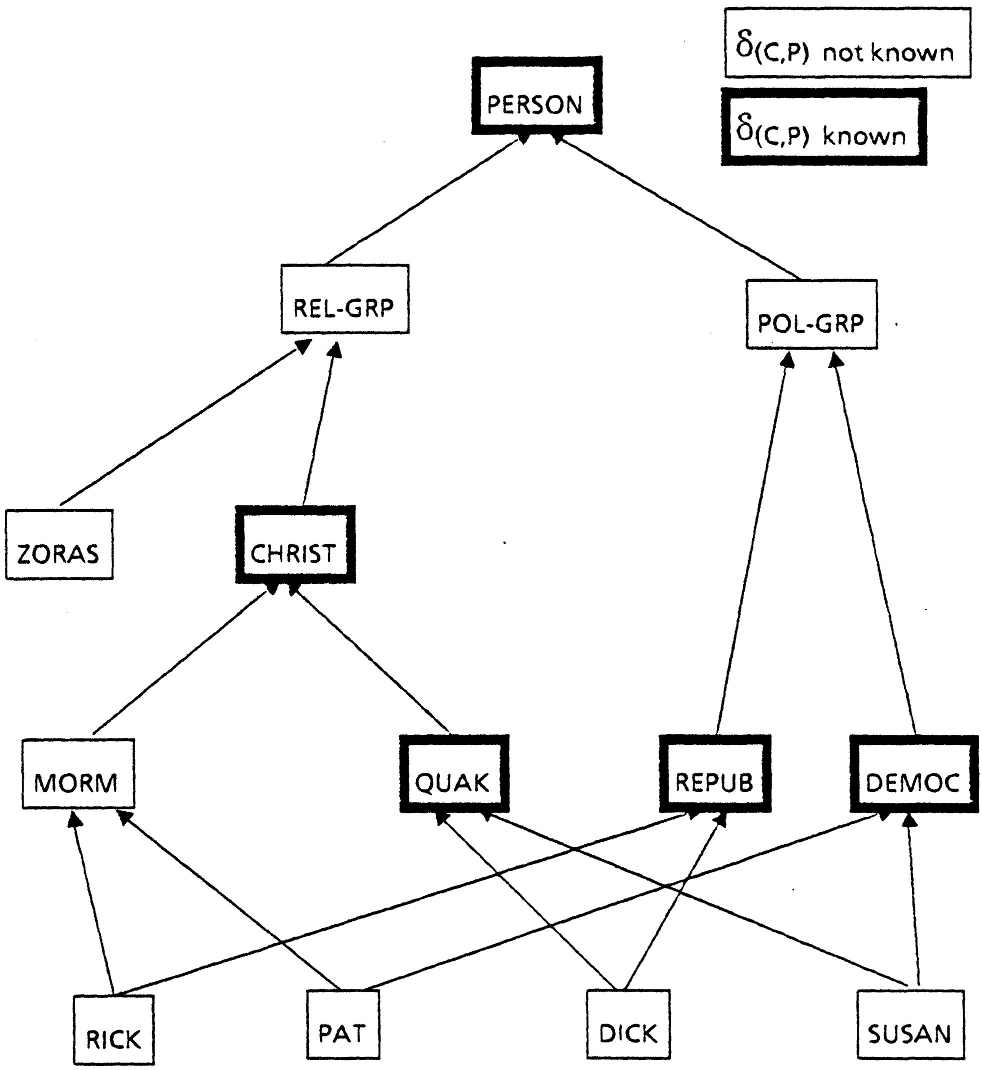


FIGURE 5.8 Generalization of multiple inheritance



P = has-belief
Values: PACIFIST, NON-PACIFIST

FIGURE 5.9a A complicated case of multiple inheritance

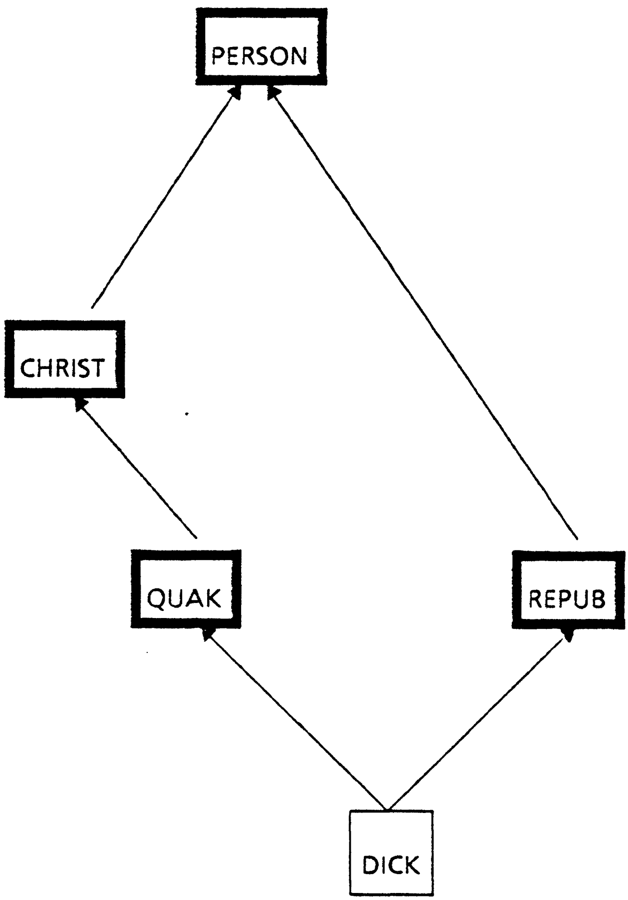
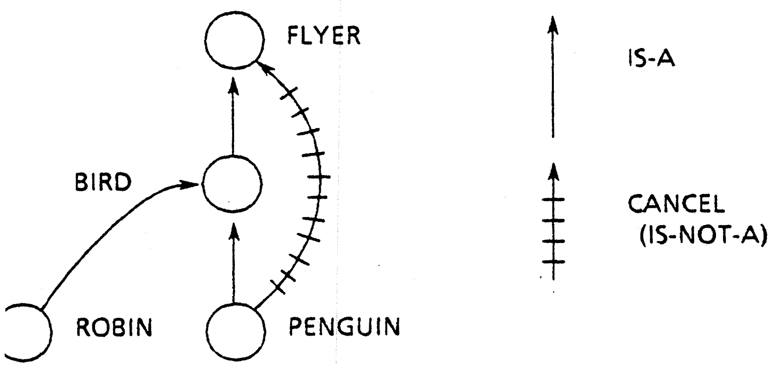


FIGURE 5.9b Projection w.r.t. DICK and has-belief

Fahlman and Touretsky



Etherington and Reiter

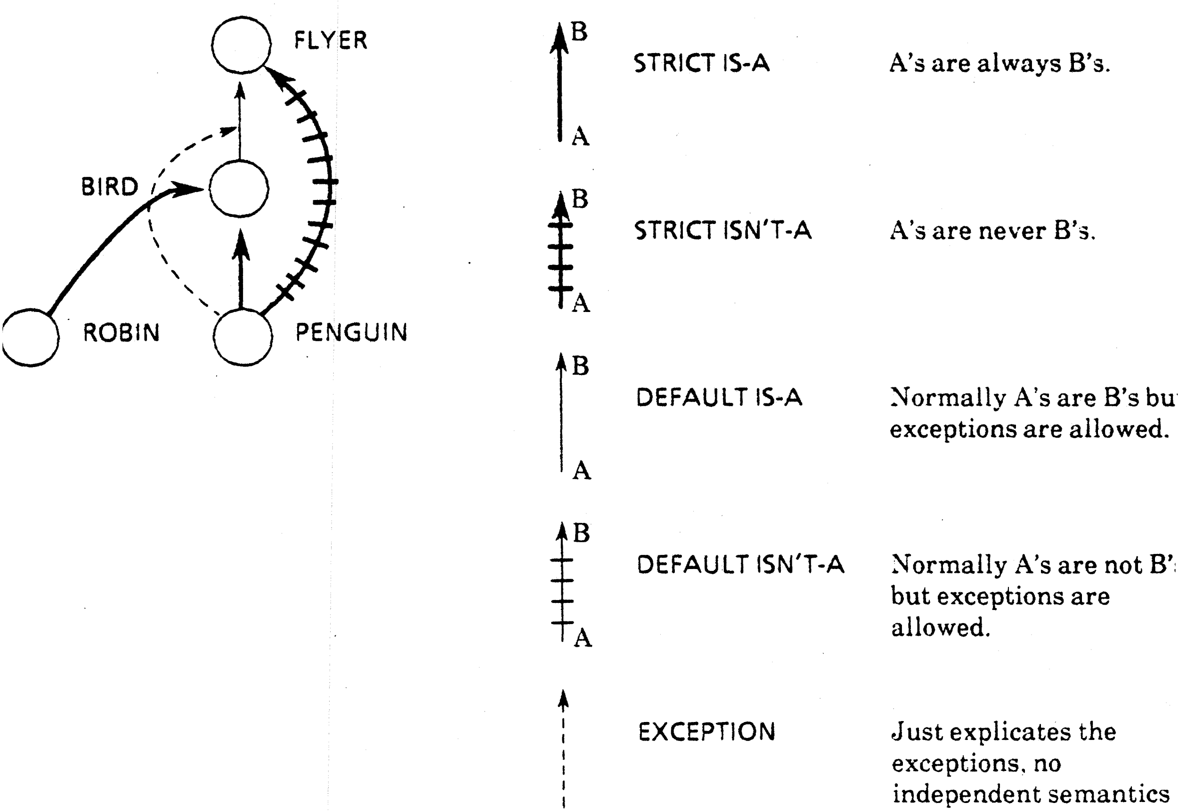


FIGURE 5.10 Representation of the penguin example

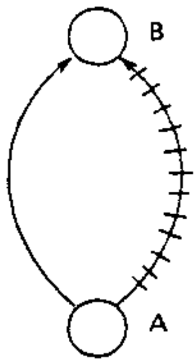


FIGURE 5.11 An ambiguous case in Etherington & Reiter's representation

Chapter 6

Parallel implementation

This chapter describes how an agent's a priori knowledge may be encoded as a network of active elements. It is shown that if the knowledge encoded in the network satisfies the assumptions listed in sections 6.1.4 and 6.2.4, then the network computes the solution to the inheritance and categorization problems in accordance with the results developed in section 5. The time required to perform these operation is only $O(d)$ where d is the maximum depth of the conceptual hierarchy (i.e. the longest path in the ordering graph defined by $\mathbb{C}$ and $\lessdot$). For most practical applications, the value of d would be small (perhaps 10).

In order to keep the exposition clear, the parallel implementation is described in four stages. Section 6.1 focuses on inheritance and suppresses all details pertaining to categorization, section 6.2 focuses on categorization and describes interconnections required to solve the categorization problem, and section 6.3 specifies how the networks described in sections 6.1 and 6.2 can be synthesized into a single network that can solve the inheritance as well as the categorization problems. Finally, section 6.4 explains the implementation of some control mechanisms alluded to in section 6.1. The chapter concludes with section 6.5 that describes how the proposed networks have been simulated on a conventional computer and presents several examples to illustrate the behavior of the networks during inheritance and categorization tasks.

The networks described below need to perform specific inferences - as against exhibiting general associative behavior or modelling diffuse priming effects. This requires that the networks be capable of controlling the propagation of activation.

Furthermore, the network design has to satisfy another crucial constraint in that *the networks should operate without the intervention of a central controller.* Once a query is posed to the network, it is expected to function autonomously, with all the nodes computing in parallel. This required that local control mechanisms be encoded in each node while satisfying the requirement that each node be a simple processing element. The design involves introducing explicit control nodes - namely binder nodes and relay nodes, that provide *foci* for controlling the spread of activation.

6.1 Inheritance

6.1.1 Encoding the conceptual structure

A concept is represented in the network by a node while its relationship with other concepts is encoded via links to appropriate nodes.

Nodes encoding concepts are called ξ-nodes. These nodes have four input sites: **QUERY**, **RELAY**, **CP**, and **HCP**. The significance of these sites will be explained in due course.

With reference to the ordering graph defined by $\mathbb{C}$ and $\ll$, if B is a parent of A then there is a ↑ (bottom up) link from A to B and a ↓ (top down) link from B to A. The weight on both these links equal #A/#B. The ↑ links are just used for spreading activation and do not have any evidential import. However, for convenience their weight is set equal to the corresponding ↓ link. All ↑ and ↓ links are incident at site **RELAY**. Figure 6.1 illustrates this situation. As the ↑ and ↓ links always occur in pairs, they will often be represented by a single undirected arc.

Each property is also encoded as a node. Such nodes are called φ-nodes, and each of these nodes has one input site: **QUERY**.

If $\delta(A,P) \in \Delta$ then for every value V_i of P there exists a node $[A,P \rightarrow V_i]$

that is connected to A, P and W_x as shown in figure 6.2.

A triangular node such as [A,P -> VJ is called a S^-node. 5^-nodes have two sites: ENABLE and EC. Each 5^-node [A,P -> VJ receives one input from node A and another from node P. Both these inputs are incident at site ENABLE, and the weight on these links is 1.0. The input from the concept node (A) is referred to as the £-input, while the input from the property node (P) is referred to as the <p-inpuL

Values are also concepts and hence they are also encoded as £-nodes. Links from 8^-nodes to £-nodes (for example, the link from [A,P -> VJ to Vj), are incident at site CP. The weights of such links are given by #A[P,V_i]/#V_i.

If B is a parent of A in the ordering induced by < on C/AJP, then there is a link from [AJP -> VJ to [B,P -> VJ. The weight on this link is given by #A[P,VJ/#B[A,VJ. Links from one S^-node to another 8^-node are incident at site EC of the destination S^-node. (Refer to figure 63).

Finally, if S(B,P) € A and there exists no C, such that B <^ C, and S(C,P) € A, then the link from [B,P -> VJ to V_x is incident at site HCP, instead of site CP. However, as before, the weight of this link equals #B[P,VJ/#V_i. (Refer to figure 6.4).

Besides the interconnections described above, *all nodes representing concepts, properties, and values* (£-nodes and <p-nodes) have an external input incident at the site QUERY, and the weight on this link is 1.0.

6.1.2 Description of network behavior

General computational characteristics

As described in section 2, each node in the network is an active element having an associated real valued potential that can take values in the interval [0, 1]. A node computes its potential based on its inputs. Each incoming link provides an input whose magnitude equals the output of the node at the source of the link times the weight on the link.

In our implementation, each node has two states: **active** or **inert.** The quiescent or normal state of each node is the **inert** state. In this state, nodes do not transmit any output A node switches to an **active** state under conditions that are specified below. In the **active** state, all nodes transmit an output equal to their potential.

There is a distinction between a node transmitting no output (NIL output), *and a node transmitting an output of magnitude 0.0.* Similarly, there is distinction between a NIL input (coming from a node not trasmitting an output), and an input of 0.0 (coming from a node transmitting an output of 0.0.

Computational characteristics of specific node type:

£-nodes:

State: Node is in the **active** state if it receives one or more inputs, otherwise it is in the **inert** state.

Potential: If no inputs at site **HCP** then

potential = the product of inputs at sites **QUERY**, **RELAY** and **CP**

else

potential = the product of inputs at sites **QUERY**, **RELAY** and **HCP**

{NIL inputs are ignored while computing the product}

δ_{inh}-nodes:

State: Node is in the **active** state if it receives both the φ-input and the ξ-input, otherwise it is in the **inert** state.

Potential: If node is in the **active** state then

potential = 1.0 * the product of inputs at sites EC

else

potential = NIL

φ-nodes:

State: Node is in the **active** state if it receives input at site QUERY, otherwise it is in the inert state.

Potential: Potential is equal to 1.0 in the **active** state, NIL otherwise.

In addition to what has been described above, the networks have an additional property in that the ↓ and ↑ links are special links. Unlike others, which always transmit the output of their source node, the ↓ and ↑ links normally remain disabled and transmit activity only when they are enabled. Each ξ-node has some additional control machinery associated with it whereby all ↓ or ↑ links emanating from it can be enabled. The enabling of ↑ and ↓ links has a chain effect. For example, if the ↑ (↓) links emanating from node A are enabled, then the ↑ (↓) links emanating from all nodes that are reachable from A via ↑ (↓) links, also get enabled. The control machinery is described in section 6.4.

6.1.3 Posing the inheritance problem and computing its solution

In the context of the network implementation, the problem of inheritance is recast as follows:

Given: i) a concept C and a property P, $P \in APL(C)$,

ii) an explicit enumeration of possible answers, i.e. a set V-SET $= \{v_1, v_2, \ldots v_n\}$ where each V-SET $\subseteq A(P)$, the set of values of P, and

iii) a reference concept REF for V-SET such that for all $V_j \in$ V-SET there exists a unique path from V_j to REF in the ordering graph defined by C and < (Typically, REF is a parent of V_j's. For example, if V_j's are RED, GREEN, BLUE ... then REF could be COLOR).

Find: $V \in$ V-SET such that relative to the values specified in V-SET, V is the most likely value of property P for concept C.

If the conceptual structure satisfies the conditions specified in section 6.1.4, the solution to the above problem may be computed using a network constructed according to the description given in section 6.1.1 and 6.1.2. The following algorithm describes how this may be done.

Phase-1

Set the external inputs, i.e. the inputs to the site QUERY, of nodes C and P to 1.0, and wait for three time steps,

Phase-2

If any $V_i \in$ V-SET reaches an active state (i.e. if any V_i receives any inputs):

then: Phase-2a

Set the external inputs to REF to 1.0, enable I links leaving REF, and wait d+3 time steps.

else { no $V_i \in$ V-SET received any activation}

Phase-2b

Set the external inputs to REF to 1.0, enable ↓ links leaving REF, enable ↑ links leaving C, and wait d+3 time steps.

The above will result in the potentials of nodes being such that the for any two nodes V_i and $V_j \in$ V-SET, the following holds:

$$(\text{potential of } V_i)/(\text{potential of } V_j) = \#C[P,V_i]/\#C[P,V_j]$$

It follows that the node V' ∈ V-SET with the highest potential will correspond to the value that is the solution to the inheritance problem.

The time required by the network to reach the desired state is $O(\max(l_1, l_2))$, where l_1 is the path length from C to some B such that, B is the highest node in the ordering graph for which $\delta(B,P) \in \Delta$, and l_2 is the maximum of the path lengths from REF to the nodes in V-SET. However, both l_1 and l_2 are bounded by the length of the longest path in the ordering graph defined by $\mathbb{C}$ and $\lll$. Thus, the time required to solve the inheritance problem is $O(d)$, where d is the depth of the conceptual structure. In the current implementation, the required time is $3*d$.

Let us consider an example to illustrate the working of the network.

Figure 6.5 depicts how the following information:

Quakers tend to be pacifists,

Republicans tend to be non-pacifists

Dick is a Quaker and a Republican.

is encoded in network form. We have interpreted the above information as follows (cf. section 4.1):

Has-bel is a property, and pacifism and non-pacifism are two values of this property. The nodes PAC and NON-PAC denote the concepts pacifism and non-pacifism respectively. The node BELIEFS is purported to represent a concept comprising of all beliefs - pacifism, non-pacifism, liberalism, nationalism ..... The node PERSON denotes a concept that is a common ancestor of the concepts Quaker (QUAK) and Republican (REPUB). It is assumed that δ[QUAK, has-bel], δ[REPUB, has-bel], δ[PERSON, has-bel] are known, while δ[DICK, has-bel] is unknown.

In order to decide - "Is Dick a pacifist or a non-pacifist"?, i.e. in order to solve the inheritance problem:

C = DICK, P = has-bel, V-SET = {PAC, NON-PAC} and REF = BELIEFS

the network is initialized by setting the external inputs of DICK and has-bel to 1.0. Because δ[DICK, has-bel] is not known, there will be no inputs reaching PAC or NON-PAC nodes. Hence, after three time steps, the external input to the node BELIEFS will be set to 1.0, and the ↑ links originating from DICK and the ↓ links originating from BELIEFS will be enabled.

The resulting potentials of some relevant nodes are as follows:

QUAK

This node receives an input at site RELAY from DICK. Hence its potential equals:

output of DICK * (1/#QUAK) = 1.0 * (1/#QUAK) = (1/#QUAK)

REPUB

This node receives an input at site RELAY from DICK. Hence its potential equals:

output of DICK * (1/#REPUB) = 1.0 * (1/#REPUB) = (1/#REPUB)

PERSON

This node receives two inputs at site RELAY, one from QUAK and another from REPUB. Hence its potential equals:

output of QUAK • (#QUAK/#PERSON) *

output of REPUB * (#REPUB/#PERSON)

« (1/#QUAK) * (#QUAK/#PERSON) * (1/#REPUB) *

(# REPUB/ # PERSON)

- 1/(#PERSON PERSON)**

[REPUB, has-bel •> PAC], [REPUB, has-bei •> NON-PAC]

These S^-nodes are in the active state because they receive their ^-inputs from REPUB and their <p-inputs form has-bel. As these nodes receive no inputs at site EC, their potential equals: 1.0

[QUAK, has-bel -> PAC], [QUAK, has-bel -> NON-PAC]

These 8^-nodes are in the active state because they receive their ^-inputs from QUAK and their cp-inputs form has-bel. As these nodes receive no inputs at site EC, their potential equals: 1.0

[PERSON, has-bel -> PAC]

This node reaches the active state because it receives its £-input from PERSON and its <p-input from has-bel. Furthermore, it receives inputs at site EC from [QUAK, has-bel -> PAC] and [REPUB, has-bei -> PAC]. Hence its potential equals:

1.0 * output of [QUAK, has-bel •> PAC] *

(#QUAK[has-bel, PAC] / #PERSON[has-bel, PAC]) *

output of [REPUB, has-bel -> PAC] *

(#REPUB[has-bel, PAC] / #PERSON[has-bel, PAC])

= (#QUAK[has-bel, PAC] / #PERSON[has-bel, PAC]) *

(#REPUB[has-bel, PAC] / #PERSON[has-bel, PAC])

[PERSON, has-bel -> NON-PAC]

The behavior of this node is analogous to the behavior of [PERSON, has-bel -> PAC], and its potential equals:

1.0 * output of [QUAK, has-bel -> NON-PAC] *

(#QUAK[has-bel, NON-PAC] / #PERSON[has-bel, NON-PAC]) *

output of [REPUB, has-bel -> NON-PAC] *

(#REPUB[has-bel, NON-PAC] / #PERSON[has-bel, NON-PAC])

= (#QUAK[has-bel, NON-PAC] / #PERSON[has-bel, NON-PAC]) *

(#REPUB[has-bel, NON-PAC] / #PERSON[has-bel, NON-PAC])

PAC

This node receives an input from BELIEFS at site **RELAY**, an input from [PERSON, has-bel -> PAC] at site **HCP**, and inputs from [QUAK, has-bel -> PAC] and [REPUB, has-bel -> PAC] at site **CP**. However, because site **HCP** receives an input, the inputs at site **EC** are ignored and the potential equals:

output of [PERSON, has-bel -> PAC] * (#PERSON[has-bel, PAC] / #PAC) *

output of BELIEFS * (#PAC / #BELIEFS)

= (#QUAK[has-bel, PAC] / #PERSON[has-bel, PAC]) *

(#REPUB[has-bel, PAC] / #PERSON[has-bel, PAC]) *

(#PERSON[has-bel, PAC] / #PAC) *

1.0 * (#PAC / #BELIEFS)

= (#QUAK[has-bel, PAC] * #REPUB[has-bel, PAC]) /

(#PERSON[has-bel, PAC] * #BELIEFS)

NON-PAC

The behavior of this node is analogous to the behavior of PAC], and its potential equals:

output of [PERSON, has-bel -> NON-PAC] *

(#PERSON[has-bel, NON-PAC] / #NON-PAC) *

output of BELIEFS * (#NON-PAC] / #BELIEFS)

= (#QUAK[has-bel, NON-PAC] / #PERSON[has-bel, NON-PAC]) *

(#REPUB[has-bel, NON-PAC] / #PERSON[has-bel, NON-PAC]) *

(#PERSON[has-bel, NON-PAC] / #NON-PAC)*

(#NON-PAC / #BELIEFS)

= (#QUAK[has-bel, NON-PAC] * #REPUB[has-bel, NON-PAC]) /

(#PERSON[has-bel, NON-PAC] * #BELIEFS)

Ignoring the common divisor (#BELIEFS), in the potentials of the nodes

PAC and NON-PAC, the potential of the node PAC corresponds to the best estimate of the number of people that are both quakers and republicans but subscribe to pacifism while the potential of the node NON-PAC corresponds to the best estimate of the number of people that are both quakers and republicans but subscribe to non-pacifism.

Hence, a comparison of the two potentials will give the most likely answer to the question: Is Dick a pacifist or a non-pacifist.

6.1.4 Network behavior: a proof of correctness.

In this section we prove that a network constructed according to the description given in sections 6.1.1 and 6.1.2, correctly[1] solves the inheritance problem provided the conceptual structure encoded by the network satisfies the conditions listed below.

The proof is based on establishing that if one focuses only on concept nodes (the ξ-nodes), then the structure of the network has the property that with reference to any query, it can be partitioned into two subparts such that in each subpart, activation spreads in a single direction - top to bottom or bottom to top. The two subparts interact via δ_{inh}-nodes. However, this interaction is again strictly unidirectional; the activation flows from one of the subparts (sat "netA") to the other (say "netB"), but not visa versa. Consequently, there are no cycles and the potentials of nodes stabilize at the appropriate value in time proportional to twice the depth of the conceptual hierarchy; the potentials of nodes in "netA" stabilize in one sweep of spreading activation and become available to nodes in "netB" which may now compute their potentials in a single sweep of spreading activation. Although the potentials of nodes in "netB" have to wait for the potentials of nodes in "netA" to stabilize, as there are no cycles and feedback effects, the activity in each subpart may proceed in parallel and *there is no need for any kind of synchronization.* The only requirement is that one wait for time proportional to twice the depth of the conceptual hierarchy before utilizing the potentials computed by the nodes.

1. Here correctness is defined relative to the solution presented in section 5.3.

The conditions imposed on the conceptual structure in order to compute the solution in parallel are:

COND-1: For any $P \in \Phi$, if $V \in \Lambda(P)$, then $P \notin \lambda(V)$. Furthermore, for all A such that $A \ll V$, $P \notin \lambda(A)$.

In other words, a property P is applicable neither to its values nor to the descendents of its values. For example, the property has-color does not apply to RED.

There are two obvious corollaries to COND-1.

a) $\forall$ A such that $V_i \ll A$ for some $V_i \in \Lambda(P)$, $P \notin \lambda(V_i)$. In simpler terms, a property is not applicable to any ancestors of its values. (For then it would be applicable to its values). With reference to the inheritance problem (cf. section 6.1.3), the above implies that $P \notin \lambda(REF)$.

b) If $P \in \lambda(C)$ then, for any $V_i \in \Lambda(P)$ it is not the case that $C \ll V_i$. In other words, C is not a descendent of any $V_i \notin \Lambda(P)$.

COND-1 requires that property values and the concepts they apply to, belong to distinct ontological Types and this appears to be a reasonable assumption. Consider the values of properties such as has-taste, has-shape, has-color on the one hand and the concepts they apply to on the other. It is easy to convince oneself that the values of has-taste (SWEET, SOUR ..), has-shape (ROUND, SQUARE ..), and has-color (RED, GREEN ...) are not subconcepts or superconcepts of objects that these properties apply to.

COND-2: For any V_i, $V_j \in \Lambda(P)$, neither $V_i \ll V_j$, nor $Vj \ll V_i$.

COND-3: The ordering graph defined by $\mathbb{C}$/C,P and $\ll$ is a tree.

This is a stronger condition than the one placed in section 5.3.5. Therein, only the portion of the ordering graph below the reference concept Ω of Γ(C,P) was required to be a tree. However, this condition trivially holds for the "multiple hierarchies" organization outlined in section 3.5, wherein, the type structure defined over Tokens consists of several distinct views.

In addition to the above constraints posed on the conceptual structure, there is an additional constraint on the manner in which the inheritance problem may be specified.

COND-4: It should not be the case that C $\ll$ REF.

One may now show that the networks compute the solution to the inheritance problem as outlined in section 5.3.5.

The proof can be broken down into two distinct cases: the local inheritance case, and the non-local inheritance case.

In the case of local inheritance, δ(C, P) is known, that is, for all possible values V_i of P, #C[P, V_i] is known, and there is a δ_{inh}-node [C,P -> V_i] connecting C, P and V_i. In the case of non-local inheritance, δ(C, P) is unknown and #C[P, V_i]'s have to be estimated based on the distributions available at concepts above C in the conceptual hierarchy. In terms of the parallel network, δ_{inh}-nodes associated with ξ-nodes that are above C in the conceptual hierarchy take part in the inheritance process.

The burden of the proof is to show that at the end of phase-2 (cf. section 6.1.3), of the inheritance process, the potentials of any pair of nodes V_i and V_j, $\in$ V-SET, are such that:

$$(\text{potential of } V_i)/(\text{potential of } V_j) = \#C[P,V_i]/\#C[P,V_j]$$

6.1.4.1 Proof for the local case of inheritance.

Lemma-L1: Besides C and P, the only other nodes active at the end of phase-1 (cf. section 6.1.3), are δ-nodes that encode $\delta(C, P)$, and ξ-nodes that represent members of $\Lambda(P)$.

Proof: At the onset of phase-1, nodes C and P are activated as a result of the input of magnitude 1.0 they receive at site QUERY. As per the potential function of ξ-nodes and φ-nodes described in section 6.1.2, C and P acquire a potential of 1.0. As none of the ↑ and ↓ links are enabled, activation from C and P spreads only along links leading into δ_{inh}-nodes. However, only δ_{inh}-nodes that encode $\delta(C, P)$, i.e. nodes labelled [C,P -> V_i], (where V_i is any member of $\Lambda(P)$), receive the ξ-input as well as the φ-input at site ENABLE, and hence, switch to the active state.

A δ_{inh}-node such as [C,P -> V_i] sends output to the node V_i and possibly to some other δ_{inh}-nodes of the form [B,P -> V_i] where $C \neq B$. However, the δ_{inh}-nodes that receive inputs from δ_{inh}-nodes such as [C,P -> V_i], do not get the appropriate ξ-input at site ENABLE, and hence, do not become active. Recall, that a δ_{inh}-node requires input from a ξ-node and a φ-node at site ENABLE to become active. Thus, at the end of the second time step of phase-1, the active nodes are:

C, P, and for every $V_i \in \Lambda(P)$, the ξ-node V_i and the δ_{inh}-node [C,P -> V_i].

The addition of nodes from $\Lambda(P)$ to the set of active nodes does not cause any new nodes to become active. To see this, consider an arbitrary node in $\Lambda(P)$, say V_i. Because the ↑ and ↓ links leaving V_i are not enabled, none of the ξ-nodes receive activation on acccount of V_i. Furthermore, as P is the only φ-node that is active, and P does not apply to V_i (by virtue of COND-1), no new δ_{inh}-nodes can become active on account of V_i.

Thus, at the end of phase-1, the set of active nodes would comprise of the nodes: C, P, and for every Vj € A(P), the £-node Vj and the S^-node [C,P -> VjJ.

This establishes Lemma-Ll.

Lemma-L2: At the end of phase-1, the potential of S^-nodes [C,P -> *V* is 1.0, while that of {-nodes Vj is given by #C[P, VJ / #*V*$_V$

Proof of Lemma-L2.

First consider the S^-nodes [C,P -> VJ. None of these nodes receive any inputs at site EC. This is because inputs to a node [C,P -> Vj] at site EC can only originate from other S^-nodes labelled [B,P -> Vj], B * C, and no such B is active. (Lemma-Ll).

Thus, the potential of each node [C,P -> V|], will be 1.0

Next consider the £-nodes Vj € A(P). Each £-node Vj receives an input from the S^-node [C,P -> VJ. The output of the S^-node is 1.0 and the weight on the link connecting [C,P -> *V* to Vj is #C[P, *V[]* / #Vj. Thus, the input to *V[* would be 1.0 * #C[P, Vj] / #Vj. As this is the only input received by Vj, its potential will be #C[P, Vj] /#V$_i$.

In calculating the potential of nodes Vj, we glossed over one detail If C is such that there exists a D such that C < D and S(D,P) is known, then the link from [C,P -> VJ would be incident on site CP of *V*$_v$ However, if such a D does not exist then the link would impinge on site HCP. In either case, the potential of *V[* computes to #C[P,Vj] / #Vj.

This establishes Lemma-L2.

At the end of phase-1, nodes *V[* would be active and hence phase-2a would ensue. This would lead to the activation of REF, and the enabling of the ↓ links emanating from REF.

Lemma-L3: As a result of phase-2a, each V_i member of V-SET, receives exactly one new input. This is incident along a ↓ link (at site **RELAY**), and has a magnitude of $\#V_i$ / $\#$REF.

In order to prove Lemma-L3, we first prove Lemma-L4.

Lemma-L4: No additional δ_{inh}-nodes become active during phase-2a.

Proof of Lemma-L4.

For a δ_{inh}-node to become active it must receive coincident inputs from an appropriate ξ-node and an appropriate δ_{inh}-node. COND-1 guarantees that P does not apply to REF or its descendants, and hence, none of the δ_{inh}-nodes that receive a ξ-input from REF or its descendants receive activation from P. But P is the only active φ-node. It follows that none of the δ_{inh}-nodes receiving activation from REF or its descendants, become active. This establishes Lemma-L4.

Proof of Lemma-L3.

As a consequence of Lemma-4, the only possible inputs to V_i's may come from ξ-nodes activated as a result of the activation of REF and the enabling of ↓ links. By definition, REF is such that, for each $V_i \in \Lambda(P)$, there exists a *unique* path from REF to V_i via ↓ links. Thus, activating REF and enabling the ↓ links results in a *single* input to each V_i along a ↓ link. Let the path form REF to V_i be via ξ-nodes D_1, D_2, ...D_k. The weights along this path would be $\#D_1/\#$REF, $\#D_2/\#D_1$, ... $\#D_k/\#D_{k-1}$, $\#V_i/\#D_k$. Thus, the input to V_i along this path would be:

output of REF * Π (the weights along the path to Vi)

that is:

$1.0 * \#D_1/\#REF * \#D_2/\#D_1 * .... \#D_k/\#D_{k-1} * \#V_i/\#D_k$,

the above equals: $\#V_i/\#REF$

Finally, as the input is along ↓ links, it would be incident at site RELAY.

This establishes Lemma-L3.

Combining the results of Lemma-L2 and Lemma-L3, and noting that the potential of each $V_i \in \Lambda(P)$, will be equal to the product of the inputs to V_i, we have:

The potential of each $V_i \in \Lambda(P)$ at the end of phase-2a is given by:

$\#C[P,V_i]/V_i * \#V_i/REF = \#C[P,V_i]/\#REF$

Ignoring the common factor in the denominator which would occur in the potential of each V_i, it is clear that each V_i has the desired potential. Specifically, for V_i, $V_j \in$ V-SET,

$$\text{(potential of } V_i)/(\text{potential of } V_j) = \#C[P,V_i]/\#C[P,V_j]$$

as was desired.

This completes the proof that the network computes the correct solution to the inheritance problem in the case of local inheritance.

6.1.4.2 Proof for the non-local case of inheritance:

As $\delta(C,P)$ is not know, there are no δ_{inh}-nodes encoding $\delta(C,P)$. Hence, the activation of C and P alone does not activate any δ_{inh}-nodes. Furthermore, because neither the ↑, nor the ↓ links are enabled, no ξ-node gets activated. Thus, at the end of phase-1, only nodes C and P are active, each with a potential of 1.0. In particular, none of the δ_{inh}-nodes in $\Lambda(P)$ are active. As a result phase-2b will ensue. This will lead to a) the activation of

REF and the enabling of the ↓ links at REF, and b) the enabling of ↑ links at C.

We will consider the effects of the two actions a) and b) separately, and then show that as far as the nodes in V-SET are considered these effects are independent.

Lemma-NL1: If C and P are active, then the following δ_{inh}-nodes are activated as a result of enabling the ↑ links at C.

$\forall\ X \in \mathbb{C}/C,P$ and $\forall\ V_i \in \Lambda(P)$, the δ_{inh}-nodes $[X,P \rightarrow V_i]$.

Proof of Lemma-NL1

By the definition of projection, each $X \in \mathbb{C}/C,P$ is such that $C \ll X$. Hence, each $X \in \mathbb{C}/C,P$ will be activated if C is active and the ↑ links at C are enabled. Furthermore, if $X \in \mathbb{C}/C,P$ then $\delta(X,P)$ is known, and there exist δ_{inh}-nodes of the form $[X,P \rightarrow V_i]$ that encode $\delta(X,P)$. As X becomes active, and P is already active, the nodes $[X,P \rightarrow V_i]$ also become active. Thus, all δ_{inh}-nodes that encode $\delta(X,P)$, for some $X \in \mathbb{C}/C,P$ become active.

This completes the proof of Lemma-NL1.

Lemma-NL2: If C and P are active, then the ξ-nodes that become active as a result of enabling the ↑ links at C are exactly those that either lie above C, or are members of $\Lambda(P)$.

The ξ-nodes activated as a result of enabling the ↑ link at C may be divided into two categories, the ones that receive activation from C along a chain of ξ-nodes connected by ↑ links (let this category be SET1), and those that receive activation along a path that includes one or more δ_{inh}-nodes (let this category be SET2).

The nodes in SET1 are exactly those ξ-nodes that are above C.

We now consider the category SET2.

At the onset, recall that the only active φ-node that is P. Consider the nodes in SET2 that are activated via a path that includes only *one* δ_{inh}-node. This implies that the solitary δ_{inh}-node in such a path must be one that it gets its ξ-input from a ξ-node in SET1, and hence it must be of the form [X,P -> V_i], where X $\in$ $\mathbb{C}$/C,P and V_i's are members of Λ(P). It follows that the nodes in SET2 that are activated via a path that includes only one δ_{inh}-node are exactly the ξ-nodes that are members of Λ(P).

However, as P is the only φ-node that is active, and as P does not apply to members of Λ(P), no new δ_{inh}-nodes become active on account of ξ-nodes that are members of Λ(P). Furthermore, by virtue of COND-1, no member of Λ(P) lies above C, and hence, the $\uparrow$ links emanating from the members of Λ(P) will not get enabled in spite of the enabling of the $\uparrow$ links at C. Consequently, no new ξ-nodes become active on account of ξ-nodes that are members of Λ(P). Thus, only ξ-nodes that are members of Λ(P) can be members of SET2.

This establishes Lemma-NL2.

Lemma-NL3: If C and P are active, then every δ_{inh}-node activated as a result of enabling the $\uparrow$ links at C is of the form:

[X,P -> Vi], where X $\in$ $\mathbb{C}$/C,P and $V_i \in \Lambda$(P)

Any active δ_{inh}-node will be receiving a ξ-input and a δ-input. By virtue of Lemma-NL2, the only ξ-nodes that are active are either those that are above C, or those that are members of Λ(P). Now, the only φ-node that is active is P. But by virtue of COND-1, the property P does not apply to any member of Λ(P), and hence, the only δ_{inh}-nodes that receive both the ξ-input and the φ-input must be those that get their ξ-input from ξ-nodes that lie above C, and their φ-input from P. But these are exactly the δ_{inh}-nodes of the form [X,P -> Vi], where $X \in \mathbb{C}/C,P$ and $V_i \in \Lambda(P)$.

This establishes Lemma-NL3.

Lemma-NL4: If C and P are active, then enabling the ↑ links at C causes *exactly* the following δ_{inh}-nodes to become active:

$\forall\ X \in \mathbb{C}/C,P$ and $\forall\ V_i \in \Lambda(P)$,

the δ_{inh}-nodes $[X,P \to V_i]$.

Proof of Lemma-NL4

Lemma-NL4 follows directly from Lemma-NL1 and Lemma-NL3.

Lemma-NL5: The set of ξ-nodes activated as a result of nodes C and P being active and the ↑ links at C being enabled, does not include REF, or any ξ-node that lies between REF and a member of V-SET.

Proof of Lemma-NL5

By virtue of Lemma-NL2, the only ξ-nodes that become active under the conditions defined in this Lemma are either those that lie above C or those that are members of $\Lambda(P)$.

By virtue of COND-4, C is not below REF. Hence, no node above C can either be REF, or any node below REF. Consequently, no node above C can either be REF, or any node that lies between REF and members of $\Lambda(P)$.

Therefore, the only ξ-nodes that lie between REF and members of $\Lambda(P)$, and may become active as a result of enabling the ↑ links at C, are nodes that are in $\Lambda(P)$. However, by virtue of COND-2, there exists no $V_j \in \Lambda(P)$ such that $V_i \ll V_j$. Therefore, there exists no $V_j \in \Lambda(P)$ that is above V_i, and hence, there exists no $V_j \in \Lambda(P)$ that lies between REF and V_i.

We have established that none of the £-nodes activated as a result of nodes C and P being active and the t links at C being enabled, lie between REF and members of A(P). (We have also shown that REF cannot be one of the nodes thus activated).

Now V-SET is a subset of A(P), therefore none of the nodes activated lies between REF and members of V-SET. This establishes Lemma-NL5.

Lemma-NL6: Given that C and P are active, the effect of activating REF and enabling the *i* links at REF, on members of V-SET, is not modified by the enabling of the t links at C.

Proof of Lemma-NL6

Lemma-L4 proved that activating REF and enabling *I* links at REF does not activate any 5^-nodes. Activation originating at REF reaches members of V-SET along a path consisting of £-nodes connected via *i* links.

Lemma-NL5 proved that none of the £-nodes activated as a result of enabling the t links at C lie between REF and members of V-SET (nor do the nodes thus activated include REF).

Thus, enabling the t links at C has no effect on the activation reaching member nodes of V-SET as a result of REF being active and the * links at REF being enabled. This establishes Lemma-NL6

Lemma-NL7: Given that C and P are active, the effect of activating the t links at C, on members of V-SET, is not modified by the activating REF and enabling the *i* links at REF.

Proof of Lemma-NL7

The effect of activating the t links at C on the members of V-SET is via £-nodes that lie above C. These in turn activate 5^-nodes that directly activate members of V-SET. Activating REF and enabling the ↓ links at REF does not activate any δ_{inh}-nodes (Lemma-L4). Furthermore, none of the ξ-nodes affected by activating REF and the enabling of ↓ links at REF lie above C. This follows from COND-4 which requires that C must not be below REF. Hence, none of the nodes that participate in affecting the members of V-SET as a result of C being active and the ↑ links at C being enabled, are modified by activating REF and enabling the ↓ links at REF. This establishes Lemma-NL7.

Lemma-NL8: Given that C and P are active, then, with respect to the member nodes of V-SET, the effect of activating REF and enabling the ↓ links at REF and the effect of enabling the ↑ links at C, are independent of each other.

Proof of Lemma-NL8

Lemma-NL8 follows directly from the conjunction of Lemma-NL6 and Lemma-NL7.

We have already established that each $V_i \in$ V-SET receives an input of magnitude $\#V_i/\#$REF at site **RELAY** as a result of activating REF and enabling the ↓ links at REF. (Lemma-L3). We now consider the effect of activating the ↑ links at C on the members of V-SET.

Lemma-NL9: For each member V_i of Λ(P), the interconnections between δ_{inh}-nodes of the form [X,P -> V_i] where X is a member of $\mathbb{C}$/C,P, are isomorphic to the ordering graph defined by $\mathbb{C}$/C,P and $\lessdot$.

Proof of Lemma-NL9

The above follows from the definition of projection and the encoding rules for δ_{inh}-nodes described in section 6.1.1. A δ_{inh}-node [X,P -> V_i] exists for every X that is a member of $\mathbb{C}$/C,P. Furthermore, there is a link from

$[X,P \rightarrow V_i]$ to $[Y,P \rightarrow V_i]$ iff Y is a parent of X in the ordering induced on $\mathbb{C}/C,P$ by $\lessdot$. This establishes Lemma-NL9.

Recall that the ordering diagram defined by $\mathbb{C}/C,P$ and $\lessdot$ is a tree (COND-3). In light of Lemma-NL9, the interconnections between δ_{inh}-nodes of the form $[X,P \rightarrow V_i]$ where X is a member of $\mathbb{C}/C,P$, also define a tree structure. By virtue of Lemma-NL4, all the δ_{inh}-nodes of the above form become active during phase-2b.

Lemma-NL10: For each $V_i \in$ V-SET, the tree of δ_{inh}-nodes referred to in Lemma-NL9 computes:

$$\text{BEST-ESTIMATE}(\Xi,P,V_i) \;/\; \#V_i$$

during phase-2b. In the above expression BEST-ESTIMATE is the function described in section 5.3.4.2 and, Ξ is such that there exists no D such that $\Xi \lessdot D$ and $\delta(D,P)$ is known.

Proof of Lemma-NL10

Except for the case where the recursion bottoms out, a call to BEST-ESTIMATE (B.E. for short), performs the following computations:

B.E. (ξ, φ, ν)

$$\underset{}{\#\xi(\varphi,\nu)} \quad \underset{\text{op3}}{*} \quad \underset{\text{op2}}{\Pi}[\ \text{B.E.}\ (\xi_i,\varphi,\nu)\ \underset{\text{op1}}{/}\ \#\xi(\varphi,\ \nu)]$$

The operation op1 follows each recursive call, one for each son of ξ, and it divides the value returned by the recursive call by $\#\xi(\varphi, \nu)$.
op2 computes the product over all the recursive calls,

and finally, op3 multiplies the result of op2 by $\#\xi(\varphi,\nu)$

In the ground case, the function simply returns $\#\xi(\varphi,\nu)$.

Now consider figure 6.6. We claim that the process enclosed in the dotted region surrounding the δ_{inh}-node [X,P -> V_i] corresponds to a call to B.E.; B.E.(X, P, Vi).

Each input link into [X,P -> V_i] corresponds to a recursive call to B.E. made from within B.E. (X,P,V_i). For example, the input link from [X_s,P -> V_i] into [X,P -> V_i] corresponds to a recursive call B.E. (X_s, P, V_i) made from within the call B.E.(X,P,V_i). However, there is one caveat: an input received along an incoming link not only corresponds to the value returned by the recursive call, but also includes the effect of performing op1. For example, the input coming into [X,P -> V_i] along the link [X_s,P -> V_i] corresponds to the value returned by B.E.(X_s,P,V_i) divided by #X[P,V_i]. Thus, instead of being performed by B.E.(X,P,V_i), op1 - which in this case is a division by #X[P,V_i], is being performed by B.E.(X_s,P,V_i). However, as is shown below, this preemptive computation of op1 is being done consistently.

To continue the correspondence between [X,P -> V_i] and B.E.(X,P,V_i), note that [X,P -> V_i] computes its potential by multiplying all its inputs and this step corresponds to op2 in the definition of B.E.

Finally, the output of [X,P -> V_i] is multiplied by #X[P,V_i]/ #X_f[P,V_i] before it appears as an input to its parent. The numerator corresponds to op3, and the denominator corresponds to the preemptive computation of op1. Normally, the division by #X_f[P,V_i] would have been performed as op1 by B.E.(X_f,P,V_i), which is the call that invokes B.E.(X,P,V_i).

The recursion bottoms out when a call is made to B.E. with a relevant concept as the first argument. In this case, the call B.E.(X,P,V_i) returns #X[P,V_i]. In the tree structure defined by δ_{inh}-nodes, the ground instances of calls to B.E. correspond to the lowest level δ_{inh}-nodes. A lowest level δ_{inh}-node [X,P -> V_i], does not receive any inputs at site EC and hence, its potential and output equals 1.0. However, before this output is incident at a higher level node, it is multiplied by a weight of the form:

$$\#X[P,V_i]/\#X_f[P,V_i]$$

where the numerator corresponds to the desired result, while the denominator corresponds to the preemptive computation of op1.

It remains to be shown where the op3 of the very first call to B.E. is performed. Thus far, the highest node in the tree of δ_{inh}-nodes does not perform op3. (op3 is performed when the output of a node is multiplied by the weight of the outgoing link). The problem is solved by the presence of the link that goes from the highest δ_{inh}-node to V_i, and is incident at site HCP. This link has a weight of $\#X[P,V_i]$ / $\#V_i$. The multiplication by $\#X[P,V_i]$ performs the required op3, although it introduces an unwanted division by the factor $\#V_i$.

Thus, the input to V_i at site HCP equals B.E.(Ξ,P,Vi) / #Vi, where Ξ is the highest ξ-node for which the distribution for P is defined; i.e. Ξ is a ξ-node such that there exists no D, such that $\Xi \lessdot D$ and δ(D,P) is known.

This completes the proof of Lemma-NL10.

In order to solve the inheritance problem we require:

(potential of V_i)/(potential of V_j) = B.E.(Ω,P,V_i)/B.E.(Ω,P,V_j)

and what we have thus far is:

1) as a result of activating REF and enabling the ↓ links at REF, each $V_i \in$ of V-SET receives an input of magnitude $\#V_i$ / #REF at site RELAY.

2) as a result of enabling the ↑ links at C, each $V_i \in$ of V-SET receives an input of magnitude B.E.(Ξ,P,V_i) at site HCP.

3) each $V_i \in$ V-SET, also receives inputs at site CP from active δ_{inh}-nodes

of the form $[X,P \to V_i]$ where $X \in \mathbb{C}/C,P$.

As per the potential function of ξ-nodes, inputs at site **CP** will be ignored because site **HCP** is active, and the potential of each meember node of V-SET will be given by the product of the inputs at site **HCP** and **RELAY**. (There are no inputs at site **QUERY**). Thus, the potential of each $V_i \in$ V-SET is:

$$B.E.(\Xi,P,V_i)/\#REF$$

The factor $1/\#REF$ occurs in the potential of all $V_i \in$ V-SET. Hence, for any V_i, $V_j \in$ V-SET

$$(\text{potential of } V_i)/(\text{potential of } V_j) = B.E.(\Xi,P,V_i)/B.E.(\Xi,P,V_j)$$

We now need to show that

$$B.E.(\Xi,P,V_i) = B.E.(\Omega,P,V_i)$$

for then we would have shown that

$$(\text{potential of } V_i)/(\text{potential of } V_j) = B.E.(\Omega,P,V_i)/B.E.(\Omega,P,V_j)$$

and hence, by the result derived in section 5,

$$(\text{potential of } V_i)/(\text{potential of } V_j) = \#C[P,V_i] \;/\; \#C[P,V_j]$$

Recall that COND-3 requires that the ordering induced by $\lessdot$ on $\mathbb{C}/C,P$ result in a tree. Hence, if there exists an Ω that is a reference concept for $\Gamma(C,P)$, it follows that the ordering induced on $\mathbb{C}/C,P$ by $\lessdot$ is as shown in figure 6.7. In particular, there is a single chain of nodes linking Ξ and Ω.

From the definition of B.E. it follows that if A has only one son say A_s, then,

$$B.E.(A,P,V_i) = \#A[P,V_i] * B.E.(A_s,P,V_i) \;/\; \#A[P,V_i] = B.E.(A_s,P,V_i).$$

As there is a linear sequence of nodes from Ξ to Ω, it follows that B.E.(Ξ,P,V_i) = B.E.(Ω,P,V_i).

This concludes the proof that the network decribed in section section 6.1.2 and 6.1.2 computes the solution to the non-local case of the inheritance problem in accordance with the solution developed in section 5.3.

The time taken by the network to solve the inheritance problem is O(*d*), where *d* is the depth of the conceptual hierarchy, i.e. the longest path in the ordering graph defined by $\lessdot$ and $\mathbb{C}$.

6.2 Categorization

This section focuses on categorization and describes how an agent's knowledge may be encoded in a network form and used to solve the categorization problem. The basic approach is analogous to the one employed in section 6.1

6.2.1 Encoding the conceptual structure

As in section 6.1.1, a concept is represented in the network by a ξ-node, and a property by a φ-node; each φ-node has a site named **QUERY**. However, the sites of ξ-nodes required for categorization are different from the sites of ξ-nodes described in section 6.1.1. ξ-nodes have four input sites: **QUERY**, **RELAY**, **INV**, and **PV**. The significance of these sites is explained below.

As in section 6.1.1, if B is a parent of A in the ordering graph defined by $\mathbb{C}$ and $\lessdot$, then there is a ↑ (bottom up) link from A to B and a ↓ (top down) link from B to A. The weight on both these links equal #A/#B. As before, the weight on all the ↑ links may uniformaly be set to 1.0 without effecting the outcome of any of the computations. All ↑ and ↓ links are incident at site **RELAY**.

If δ(A,P) $\in \Delta$ then for every value V_i of P there exists a δ_{cat}-node [P,V_i ->

A] that is connected to A, P and Vj as shown in figure 6.8. Notice the reversal of connections in the S_{cat}-nodes. The 8_{cat}-nodes described in section 6.1.1 had the form [C,P -> VJ and the output of such a node fed into the "value" node. However, the output of 5_{cat}-nodes that participate in categorzation is incident at the "concept" node. For instance, the output of the 8_{cat}-node [P,Vj -> A] is incident at the site PV of the £-node A, and the weight of this **link** is given by #A[P,V_i] / #A.

S_{cat}-nodes that participate in categorization have one site: **ENABLE.** Each S_{cat}-node [P,V^ -> A] receives one input from node A and another from node P. Both these inputs are incident at site **ENABLE,** and the weight on these links is 1.0. The input from the concept node (A) is referred to as the £-input, while the input from the property node (P) is referred to as the <p-input.

If B is a parent of A in the ordering induced by <^ on C/A,P, then there is **a** link from [P,Vi -> B] to A. This input is incident at site **INV** of the £-node A, and the weight on this link is given by #B[P,Vj]/#B. (Refer to figure 6.9).

As in section 6.1.1, all nodes representing concepts, properties, and values (£-nodes and <p-nodes) have an external input incident at the site QUERY, and the weight on this link is 1.0.

6.2.2 Description of **network** behavior

As in section 6.1.2, each node in the network is an active element with an associated potential and two states: active and **inert.** The basic behavior of the nodes is similar to the nodes described in section 6.1.2. The following describes the potential function and the state function of each node type:

£-nodes:

State: Node is in the **active** state if it receives one or more inputs, otherwise it is in the **inert** state.

Potential: potential = the product of inputs at sites **QUERY, RELAY, PV** and **INV**

{NIL inputs are ignored while computing the product}

8_{cat}-nodes:

State: Node is in the **active** state if it receives both the cp-input and the £-input, otherwise it is in the **inert** state.

Potential: If node is in **active** state then

potential = 1.0

else

potential = NIL

<p-nodes:

State: Node is in the **active** state if it receives input at site **QUERY,** otherwise it is in the **inert** state.

Potential: Potentialis equal to 1.0 in the **active** state, NIL otherwise.

Additional control machinery is associated with £-nodes whereby all I or t links emanating from it can be enabled. As explained in section 6.1.2, the enabling of t and *l* links has a chain effect.

6.2.3 Posing the categorization problem and computing its solution

In the context of the network implementation, the categorization problem is recast as follows:

Given: i) an explicit enumeration of possible answers, i.e. a set of concepts, C-SET = {C^, C2, ... C_n}, where either all members of C-SET are Types, or all members of C-SET are Tokens.

ii) a reference concept REF, such that for all C $\in$ C-SET, C $\preccurlyeq$ REF. For example, if C-SET = {APPLE, GRAPE, PEAR}, then REF could be FRUIT.

iii) a description consisting of a set of property value pairs, i.e. a set DISCR = { $[P_1,V_1]$, $[P_2,V_2]$, ... $[P_m,V_m]$ }, such that,

for each $[P_j,V_j] \in$ DISCR, $P_j \in \cap_{C \in \text{C-SET}} \lambda(C)$, and

$V_j \in \Lambda(P_j)$.

In other words, each property mentioned in the description should apply to every concept in C-SET, and the values specified for these properties should be appropriate.

Find: C' $\in$ C-SET such that relative to the concepts specified in C-SET, C' is the most likely concept described by DISCR.

If the conceptual structure satisfies the conditions specified in section 6.2.4, the solution to the above problem may be computed using a network constructed according to the description given in section 6.2.1 and 6.2.2. The following algorithm describes how the categorization problem is to be posed to the network:

For each $[P_j,V_j] \in$ DISCR, set the inputs to the site QUERY of nodes P_j and V_j to 1.0. At the same time, set the input to the site QUERY of REF to 1.0, and enable the ↓ links emanating from REF. Wait d + 3 time steps, where d is the longest path in the ordering graph defined by $\mathbb{C}$ and $\preccurlyeq$.

The above will result in the potentials of nodes being such that, for any two nodes C_i and $C_j \in$ C-SET, the following holds:

(potential of C_i)/(potential of C_j) equals:

the best estimate of $\#C_i[P_1,V_1][P_2,V_2] \ldots [P_m,V_m]$ divided by the best estimate of $\#C_j[P_1,V_1][P_2,V_2] \ldots [P_m,V_m]$.

The best estimates referred to above are computed based on the result derived in section 5.1.

It follows that the node C' ∈ C-SET with the highest potential will correspond to the concept that is the solution to the categorization problem.

The time required by the network to reach the desired state will be $O(\max(l_i))$, where each l_i is the path length, from REF to some C_i ∈ C-SET, in the ordering graph defined by $\mathbb{C}$ and $\lessdot$. However, each l_i is bounded by *d*, the length of the longest path in the ordering diagram. Thus, the time required to solve the categorization problem is O(*d*). In the current implementation the time required is 3**d*.

We illustrate how the network computes a solution to the categorization problem with reference to the network in figure 6.10. The network is intended to depict the following information:

Fruits and vegetables are a kind of edible thing.

Grapes and apples are a kinds of fruit.

Root-vegetables are a kind of vegtable, and

Beet is a root-vegetable.

Red and green are two values of has-color, while sweet and sour are two values of has-taste.

Edible things have the property has-color and has-taste associated with them. The distribution for the property has-taste is known for fruits, grapes, and vegetables, while the distribution for the property has-color is known for fruits, and beets. In other words, S[FRUIT, has-color], 5[FRUIT, has-taste], S[GRAPE, has-taste], 8[VEGGIE, has-taste] and S[BEET, has-color] are known.

The network in figure 6.10 encodes the above information as per the description given in section 6.2.1, except that the 5_{cat}-nodes and links associating SOUR and GREEN nodes to appropriate nodes in the hierarchy have been omitted as they do not play a role in the example, and would unnecessarily complicate the diagram.

Notice that for GRAPE, information about color has to be inherited from FRUIT, but specific information about taste is available locally, in addition to the more general information available at FRUIT. For APPLE, the information about taste as well as color has to be inherited from FRUIT, while for beets, the information about color is available locally, but the information about taste is to be inherited from VEGGIE which is two levels away in the conceptual hierarchy.

In order to decide - "Is a red and sweet edible thing an apple, a grape, or a beet[11]?, i.e. in order to solve the inheritance problem:

C-SET = {APPLE, GRAPE, BEET}, REF = ED-THING, DISCR = { [has-color, RED], [has-taste, SWEET]}

The network is initialized by setting the external inputs of has-taste, has-color, RED, SWEET, ED-THING to 1.0, and enabling the i links at REF.

After d + 3 time time steps the resulting potentials of some relevant nodes are as follows:

ED-THING = 1.0

All 5_{cat}-nodes shown in figure 6.10 will be active and their potential will be

1.0.

FRUIT = potential of ED-THING * (#FRUIT / # ED-THING) *

(#FRUIT[has-col, RED] / #FRUIT) *

(#FRUIT[has-taste, SWEET] / #FRUIT)

= 1.0 * (#FRUIT / #THING-TH) *

(#FRUIT[has-col, RED] / #FRUIT) *

(#FRUIT[has-taste, SWEET] / #FRUIT)

= (#FRUIT[has-color, RED] * (#FRUIT[has-taste, SWEET]) /

(#FRUIT * #ED-THING)

VEGGIE = potential of ED-THING * (#VEGGIE / #ED-THING) *

(#VEGGIE[has-taste, SWEET] / #VEGGIE)

= 1.0 * (#VEGGIE / #ED-THING) *

(#VEGGIE[has-taste, SWEET] / #VEGGIE)

= (#VEGGIE[has-taste, SWEET] / #ED-THING

GRAPE = potential of FRUIT * (#GRAPE / #FRUIT) *

(#GRAPE[has-taste, SWEET] / #GRAPE) /

(#FRUIT[has-taste, SWEET] / #FRUIT)

= (#FRUIT[has-color, RED] * #GRAPE[has-taste, SWEET]) /

(#FRUIT * #ED-THING)

APPLE = potential of FRUIT * (#APPLE / #FRUIT)

= (#FRUIT[has-color, RED] *

(#FRUIT[has-taste, SWEET]* #APPLE) /

(#FRUIT* #FRUIT* #ED-THING]

ROOTV = potential of VEGGIE * (#ROOTV / # VEGGIE)

= (#VEGGIE[has-taste, SWEET]* #ROOTV) /

(#VEGGIE* #ED-THING)

BEET = potential of ROOTV * (#BEET / #ROOTV) *

(#BEET[has-color, RED] / # BEET)

= (#VEGGIE[has-taste, SWEET] * #BEET[has-color, RED]) /

(#VEGGIE* #ED-THING])

Ignoring the common divisor (#ED-THING) in the potentials of nodes GRAPE, APPLE, and BEET, the potential of the node GRAPE corresponds to the best estimate of the number of red and sweet grapes, the potential of node APPLE corresponds to the best estimate of the number of red and sweet apples, while the potential of the node BEET corresponds to the best estimate of the number of red and sweet beets.

Hence, a comparison of the three potentials will give the correct answer to the question: IS a red and sweet edible thing is an apple, a grape, or a beet?

In order to understand the significance of the potentials of the nodes GRAPE, APPLE, and BEET, we elaborate on the potential of GRAPE.

The best estimate of the number of red grapes is:

(#FRUIT[has-color, RED] * #GRAPE / #FRUIT) ; *via direct inheritance*

The number of sweet grapes is:

(#GRAPE[has-taste, SWEET]) ; 5(GRAPE, has-taste) *is known.*

Therefore, by the result derived in section 5.5, the best estimate of the number of red and sweet grapes is:

((#FRUIT[haS-COlor, RED] * # GRAPE / # FRUIT) *

#GRAPE[haS-taste, SWEET]) / # GRAPE

which may be simplified to yield:

(#FRUiT[has-color, RED]*#GRAPE[has-taste, SWEET]) / #FRUIT

which is exactly the potential of the node GRAPE if we ignore the common denominator #ED-THING.

A similar analysis of the potentials of nodes APPLE, and BEET leads to similar result.

6.2.4 Network behavior: a proof of correctness.

In this section we prove that a network constructed according, to the description given in section 6.2.1 and 6.2.2 correctly[2] solves the categorization problem, provided the conceptual structure encoded by the network satisfies the conditions listed below. The basic strategy is the same as the one adopted in section 6.1.4.

2. Here correctness is defined relative to the solution presented in section 5.5.

COND-1: The conceptual structure is as defined in section 3.5, and satisfies the well-formedness rules WFR-mv-1 and WFR-mv-2.

COND-2: For any $P \in \Phi$, if $V \in \Lambda(P)$, then $P \notin \lambda(V)$. Furthermore, for all A such that, $A \ll V$, $P \notin \lambda(A)$.

This condition is the same as the one imposed in section 6.1.4. According to it, the values of a property, and the concepts that the property applies to, belong to distinct ontological Types.

In addition to the above constraints on the conceptual structure, there is an additional constraint on the manner in which the categorization problem may be posed.

COND-3: Let ω be the leaf of the ontological tree that is required by WFR-cat-1 (cf. section 5.5.3) to be the ancestor of all member of C-SET. Then REF should be such that $\omega \ll$ REF.

In other words, REF should be at least as general as the most specific ontological Type that is the common ancestor of members of C-SET.

Condition COND-3 is required only if members of C-SET have multiple parents, i.e. there are multiple views defined with respect to ω. If members of C-SET have multiple parents and COND-3 is not satisfied, then, when computing the solution to the inheritance problem, the network will be unable to incorporate information available in views other than the one that includes REF.

As discussed in section 5.5.3, there are two cases of categorization; Type categorization and Token categorization.

6.2.4.1 Proof for Type categorization

In order to prove that the network computes the correct solution, we have to show that after $O(d)$ time steps the potential of each C_i member of C-SET equals:

$C_i * \Pi_{j=1,m}(\#B_{ij}[P_j,V_j] \;/\; \#B_{ij})$

where P_j, $1 \le j \le m$, are properties mentioned in DISCR, and B_{ij} is the concept that is relevant to C_i w.r.t P_j.

Instead, we will prove that the potential of each C_i member of C-SET equals:

$(1/CONST) * C_i * \Pi_{j=1,m}(\#B_{ij}[P_j,V_j] \;/\; \#B_{ij})$

where (1/CONST) is a common factor in the potential of all $C_i \in$ C-SET.

Proof:

Because each C_i is a Type, COND-1 entails that there can be *at most* one concept B_{ij} that is relevant to C_i w.r.t. P_j. It also entails that there is a unique path Ψ from REF to C_i. Let this path be REF, D_1, D_2, ... D_s, C_i. Note that Ψ has been defined so as to include REF and C_i.

COND-1 in conjunction with COND-3 entails that for each P_j, there must be *at least* one node X such that, $C_i \ll X \ll$ REF, and $\delta(X,P_j)$ is known. The latter follows from the fact that $\omega \ll$ REF, and for each leaf ω of the ontological tree, $\delta(\omega,P)$ is known if $P \in \lambda(\omega)$. (WFR-mv-1, Cf. section 4.1).

Hence, each B_{ij} is on Ψ.

Lemma-TC1: The only δ_{cat}-nodes activated during categorization are of the form $[P_j,V_j$ -> Y], for all $[P_i,V_j] \in$ DISCR.

Proof of Lemma-TC1

All δ_{cat}-nodes of the form $[P_j,V_j$ -> Y] become active when the categorization problem is posed to the network because the initialization involves activating all the nodes P_j and V_j such that $[P_j,V_j] \in$ DISCR. The only other node activated during intialization is the ξ-node REF. COND-2 implies that none of the £-nodes reachable from REF via i links can be a node in A(Pj) for any P_j mentioned in DISCR. But the only <p-nodes active are Pj's that are mentioned in DISCR. Thus no 5_{cat}-node receiving its <p-input, receives its |-input because of REF, or nodes activated by REF. Hence, no new 8_{cat}-nodes become active because of REF, or nodes activated by REF. COND-2 also implies that none of the £-nodes activated by 5_{cat}-nodes of the form [Pj,Vj -> Y] can be a member of A(Pj) for any P_j mentioned in DISCR. Thus, the only 8_{cat}-nodes that become active during categorization are nodes of the form [Pj,Vj -> Y].

This concludes the proof of Lemma-TCl

Lemma-TC2: The only £-nodes transmitting activation to *C[* are nodes that lie on *.

Proof of Lemma-TC2

The only way a £-node may transmit activation to Cj is via *i* links activated as a result of activating REF and enabling the I links at REF. These are exactly the nodes on the path *.

This concludes the proof of Lemma-TC2.

The only nodes that can transmit activity to *C[* are i) 8_{cat}-nodes of the form [Pj,Vj -> Z], where Z is a parent of REF in the ordering defined by < on C/CpPj, ii) S_{cat}-nodes of the form [Pj,Vj -> X], and iii) £-nodes X, where X lies on *. Note that, each By is one such X. With reference to the 5_{cat} nodes of the form [Pj,Vj -> X], the links from these nodes to £-nodes at site INV ensure that only the inputs from nodes that are of the form [Pj,Vj -> By] contribute to the potential of C_v Thus, the net potential of C[is given by the product of three terms:

The first term is contributed by the nodes of the form [Pj,Vj -> Z]. The inputs from these nodes are incident at REF and hence, their effect is identical on all C_i's and may be ignored.

The second term is the effect of the activation travelling along the path from REF to C_i and is given by:

$$1.0 * \#D_1/\#REF * \#D_2/\#D_1 * \ldots \#D_s/\#D_{s-1} * \#C_i/\#D_s$$

The above reduces to $\#C_i/\#REF$.

The third term is the contribution of all the δ_{cat}-nodes of the form $[P_j, V_j \text{-}> B_{ij}]$ and equals:

$$\Pi_{j=1,m}(\#B_{ij}[P_j,V_j] \;/\; \#B_{ij})$$

Thus, the net potential of C_i is:

$$(1/\#REF) * C_i * \Pi_{j=1,m}(\#B_{ij}[P_j,V_j] \;/\; \#B_{ij})$$

The factor $1/\#REF$ is also common to all members of C-SET and may be ignored.

Furthermore, the time taken for the appropriate activation to reach C_i would equal the length of the path from REF to C_i. Notice that all relevant δ_{cat}-nodes would be activated in the very first step and hence, the total time required to compute the solution remains proportional to the length of the path from REF to C_i. The latter can be at most *d*, where *d* is the longest path in the conceptual hierarchy.

This concludes the proof for the case of Type categorization.

6.2.4.2 Proof for Token Categorization

For Token categorization one must show that the potential of each C_i equals:

$$\#C_i * \Pi_{j=1,m;k=1,q_{ij}}(\#B_{ij}^k[P_j,V_j] \;/\; \#B_{ij}^k)$$

where B_{ij}^k's are the concepts that are relevant to C_i with respect to property P_j. Each B_{ij}^k lies in a different view defined over C_i.

However, since C_i's are Tokens, $\#C_i = 1$, and the above expression equals:

$$\Pi_{j=1,m;k=1,q_{ij}}(\#B_{ij}^k[P_j,V_j] \ / \ \#B_{ij}^k)$$

One can show that the potential of each C_i equals:

$$\alpha * \Pi_{j=1,m;k=1,q_{ij}}(\#B_{ij}^k[P_j,V_j] \ / \ \#B_{ij}^k)$$

where α equals:

$$(1/\#REF)^q * \Pi_{j=1,m;k=1,q}(\beta_j^k)$$

In the above expression, ω is the leaf of the ontological tree that is a common ancestor of all members of C-SET and q is the number of views defined w.r.t. ω.

β_j^k equals 1 if there exist B_{ij}^k's relevant to C_i's, w.r.t. P_j's, in the paths from REF to C_i's via view H_k, otherwise β_j^k equals $\#\omega[P_j,V_j] \ / \ \#\omega$.

The proof is essentially the same as that for Type categorization, except that there are q paths from REF to each C_i. The activation along each of these paths contributes the factor $1/\#REF$, in addition to the appropriate factors of the form $\#B_{ij}^k[P_j,V_j] \ / \ \#B_{ij}^k$ introduced by the δ_{cat}-nodes activated during the initialization of the network. Furthermore, if a view H_k is such that there are no concepts X in H_k for which $\delta(X,P_j)$ is known, then the activation propagated from REF to each C_i along each path in H_k would also include the factor $\#\omega[P_j,V_j] \ / \ \#\omega$. Thus, the potential of each C_i is given by:

$$(1/\#REF)^q * \Pi_{j=1,m;k=1,q}(\beta_j^k) *$$
$$\Pi_{j=1,m;k=1,q_{ij}}(\#B_{ij}^k[P_j,V_j] \ / \ \#B_{ij}^k)$$

Recall that either all members of C-SET have relevant concepts w.r.t. to a property P_j in a view H_k, or none of them have (WFR-mv-2 cf. section 5.5.3). Therefore, $(1/\#REF)^q * \Pi_{j=1,m;k=1,q}(\beta_j{}^k)$ is a common factor for each member of C-SET. Hence, the relative potentials of C_i's may be used to find the answer to the categorization problem.

As the propagation time from REF to members of C-SET is not a function of the number of paths from REF to members of C-SET, the time required to find a solution to the categorization problem is still bounded by *d*, the depth of the conceptual structure.

6.3 A single network for inheritance and categorization

This section explains how a single network can be constructed to perform both inheritance and categorization tasks. Such a network is formed by combining into one network, the computational capabilities of the networks described in sections 6.1 and 6.2. The separate descriptions in the last two sections were strictly for pedagogical purposes.

6.3.1 Encoding the conceptual structure

In the combined network, each concept is represented by a ξ-node that has six sites: **QUERY**, **RELAY**, **CP**, **HCP**, **PV**, and **INV**. These are a combination of the sites that were present in ξ-nodes of the inheritance and the categorization networks.

The φ-nodes in the combined network are exactly like the φ-nodes described in sections 6.1 and 6.2.

If $\delta(C,P)$ is known, then for every $V_i \in \Lambda(P)$ the network has δ_{inh}-nodes of the form [C,P -> V_i] and δ_{cat}-nodes of the form [P,V_i -> C]. The interconnections between δ-nodes (i.e. δ_{inh}-nodes or δ_{cat}-nodes), ξ-nodes, and φ-nodes are *exactly* as described in section 6.1.1. and 6.2.2. (Notice that

ξ-nodes in the combined network have all the requisite sites).

In addition to the ξ-nodes, φ-nodes and the δ-nodes, the combined network has two more nodes: INHERIT and CATEGORIZE. Each of these nodes has one input site: QUERY, at which it receives an external input whose weight is 1.0. Each δ_{inh}-node receives an input from the node INHERIT at site ENABLE, and each δ_{cat}-node also receives an input from node CATEGORIZE at site ENABLE. The weight o these links are 1.0.

6.3.2 Description of network behavior

The following describes the computational characterstics of each node type in the combined network.

ξ-nodes:

State: Node is in **active** state if it receives one or more inputs.

Potential: If no inputs at site HCP then

potential = the product of inputs at sites QUERY, RELAY, CP, PV and INV.

else

potential = the product of inputs at sites QUERY, RELAY, HCP

{NIL inputs are ignored while computing the product}

δ_{inh}-nodes:

State: Node is in **active** state if and only if it receives the φ-input, the ξ-input and the input from INHERIT node. (The three inputs at site ENABLE).

Potential: If node is in **active** state then

potential = 1.0 * the product of inputs at sites EC

else

potential = NIL

δ_{cat}-nodes:

State: Node is in **active** state if and only if it receives the φ-input, the ξ-input and the input from CATEGORIZE node. (The three inputs at site ENABLE).

Potential: If node is in **active** state then

potential = 1.0

else

potential = NIL

φ-nodes:

State: Node is in **active** state if it receives input at site QUERY.

Potential: Potential is always equal to 1.0 in the **active** state.

ENABLE and CATEGORIZE nodes:

State: Node is in **active** state if it receives input at site QUERY.

Potential: Potential is always equal to 1.0 in the **active** state.

As mentioned in sections 6.1.3 and 6.1.4, unlike other links that always transmit the output of their source node, the ↑ and ↓ normally remain disabled, and transmit activity only when they are enabled. This control is affected via additional control machinery associated with £-nodes whereby all **i** or t links emanating from it can be enabled. Furthermore, the enabling of t and I links has a chain effect The implementation details are specified in the section 6.4.

6.3.3 Posing the problems and computing their solution

The inheritance and categorization problems are posed exactly as described in sections 6.1.3 and 6.2.3, except that when an inheritance (categorization) problem is posed to the network, the external input incident at site **QUERY** of node INHERIT (CATEGORIZE) is set to 1.0. The answers to the inheritance and categorization problems are obtained exactly as described in sections 6.1.3 and 6.2.3 respectively.

Inheritance problem

Given: i) a concept C and a property P, P € \(C),

ii) an explicit enumeration of possible answers, i.e. a set V-SET = {^{V}l> ^{V}2* - ^{v}n } where each V-SET C_ A(P), the set of values of P, and

iii) a reference concept REF for V-SET such that for all Vj € V-SET there exists a unique path from Vj, to REF in the ordering graph defined by C and < (Typically, REF is a parent of Vj's. For example, if *V{s* are RED, GREEN, BLUE ... then REF could be COLOR).

Find:NT € V-SET such that relative to the values specified in V-SET, V is the most likely value of property P for concept C.

If the conceptual structure satisfies the conditions specified in section 6.1.4, the solution to the above problem may be computed by as follows:

Phase-1

Set the external inputs, i.e. the inputs to the site **QUERY**, of nodes C, P and INHERIT to 1.0, and wait for three time steps,

Phase-2

If any Vj € V-SET reaches an **active** state {i.e. if any Vj receives any inputs):

then: Phase-2a

Set the external inputs to REF to 1.0, enable I links leaving REF, and wait d+3 time steps.

else { no Vj 6 V-SET received any activation}

Phase-2b

Set the external inputs to REF to 1.0, enable I links leaving REF, enable t links leaving C, and wait d+3 time steps.

The above will result in the potentials of nodes being such that for any two nodes Vj and Vj € V-SET, the following holds:

(potential of ^/(potential of Vj) = #C[P,Vi]/#C[P,Vj]

It follows that the node V € V-SET with the highest potential will correspond to the value that is the solution to the inheritance problem.

The time required by the network to reach the desired state will be $O(d)$, where *d* is the depth of the conceptual structure. In the current implementation the time required is *3*d*.

Categorization

Given: i) an explicit enumeration of possible answers, i.e. a set of concepts C-SET $\subseteq \mathbb{C}$.

ii) a reference concept REF, such that for all C $\in$ C-SET, C $\ll$ REF. For example, if C-SET = {APPLE, GRAPE, PEAR}, then REF could be FRUIT.

iii) a description consisting of a set of property value pairs, i.e. a set DISCR = { $[P_1,V_1]$, $[P_2,V_2]$, ... $[P_m,V_m]$ }, such that,

for each $[P_i,V_i] \in$ DISCR, $P_i \in \cap_{C \in \text{C-SET}} \lambda(C)$, and $V_i \in \Lambda(P_i)$.

In other words, each property mentioned in the description should apply to every concept in C-SET, and the values specified for these should be appropriate.

Find: C' $\in$ C-SET such that relative to the concepts specified in C-SET, C' is the most likely concept described by DISCR.

If the conceptual structure satisfies the conditions specified in section 6.2.4, the solution to the above problem may be computed as follows:

For each $[P_j,V_j] \in$ DISCR, set the inputs to the site QUERY of nodes P_j and V_j to 1.0. At the same time, set the input to the site QUERY of CATEGORIZE and REF to 1.0, and enable the $\downarrow$ links emanating from REF. Wait $d + 3$ time steps, where d is the longest path in the ordering graph defined by $\mathbb{C}$ and $\ll$.

The above will result in the potentials of nodes being such that, for any two nodes C_i and $C_j \in$ C-SET, the following holds:

(potential of C_i)/(potential of C_j) equals:

the best estimate of $\#C_i[P_1,V_1][P_2,V_2] \ldots [P_m,V_m]$ divided by the best estimate of $\#C_j[P_1,V_1][P_2,V_2] \ldots [P_m,V_m]$.

The best estimates referred to above are computed based on the result derived in section 5.3.

It follows that the node C' $\in$ C-SET with the highest potential will correspond to the concept that is the solution to the categorization problem.

The time required by the network to reach the desired state will be $O(d)$. In the current implementation the time required is $3*d$.

6.3.4 Network behavior: a proof of correctness

The combined network essentially operates in two modes: inheritance and categorization. It can be shown that the computations performed by the combined network, (henceforth simply the network), in the inheritance mode are exactly the computations performed by the network described in section 6.1, and similarly, the computations performed by the combined network in the categorization mode are exactly the computations performed by the network described in section 6.2.

6.3.4.1 Inheritance mode

In this mode, the node INHERIT is active while the node CATEGORIZE is inert. We show that in this mode, all the ξ-nodes, δ-nodes, and φ-nodes behave exactly as they did in the inheritance networks described in sectio 6.1.

φ-nodes: As was the case with the inheritance network, a φ-node receives exactly one input, namely, an external input at site QUERY, and has the same state and potential functions. In both the networks, a φ-node becomes active under exactly the same conditions, namely, when the inheritance problem specifies a property P encoded by the φ-node.

δ-nodes: During inheritance, the CATEGORIZE node remains inert, and hence, *all the* δ_{cat}-nodes *remain inert throughout the inheritance mode.* Furthermore, since the INHERIT node remains active throughout the inheritance mode, all δ_{inh}-nodes receive an input from INHERIT node at site **ENABLE** throughout the inheritance mode. Consequently, the enabling conditions, and the state and potential functions of the δ_{inh}-nodes become exactly identical to that of the δ_{inh}-nodes in the inheritance networks described in section 6.1. Thus, each δ_{inh}-node behaves exactly like the corresponding δ_{inh}-node in the inheritance network.

ξ-nodes: The ξ-nodes now have six sites: **QUERY**, **RELAY**, **CP**, **HCP**, **PV**, and **INV** instead of the four sites: **QUERY**, **RELAY**, **CP** and **HCP** present in the inheritance networks. Furthermore, the state and potential functions now depend on the inputs received at all these sites.

However, in spite of these changes, the behavior of ξ-nodes remains the same as the behavior of ξ-nodes in the inheritance network. This is because in the inheritance mode, none of the δ_{cat}-nodes ever become active, and these are the only nodes that send inputs to ξ-nodes at sites **PV** and **INV**. Consequently, only the appropriate sites of the ξ-nodes namely, **QUERY**, **RELAY**, **CP** and **HCP**, take part in computations performed during the inheritance mode. It is easy to see that under these conditions, the state and potential functions of ξ-nodes reduce to the state and potential functions of ξ-nodes specified in section 6.1.2. Thus, each ξ-node behaves exactly like the corresponding ξ-node in the inheritance network.

We have established that during the inheritance mode, all the ξ-nodes, δ_{inh}-nodes, and φ-nodes of the combined network behave exactly like the corresponding nodes in the inheritance network. We have also shown that the δ_{cat}-nodes do not play any role in this mode. Finally, because of the manner in which the inheritance problem is posed, the initial state of all the ξ-nodes, δ_{inh}-nodes, and φ-nodes in the combined network corresponds to the initial state of the corresponding nodes in the inheritance network. Thus, the combined network solves the inheritance problem.

6.3.4.2 Categorization mode

In this mode, the node CATEGORIZE is active while the node INHERIT is inert. It can be shown by arguments similar to those given in section 6.3.4.1, that in the categorization mode, all the ξ-nodes, δ-nodes, and φ-nodes behave exactly as they did in the categorization networks described in section 6.2.

φ-nodes: By an argument analogous to that put forth in the previous section, it can be shown that in the categorization mode, φ-nodes behave exactly like the φ-nodes in the categorization network described in section 6.2.

δ-nodes: During categorization, the INHERIT node remains inert, and hence, all the δ_{inh}-nodes also remain inert. Furthermore, since the CATEGORIZE node remains active throughout this mode, all δ_{cat}-nodes receive an input from the CATEGORIZE node at site **ENABLE**. Consequently, the enabling conditions, and the state and potential functions of the δ_{cat}-nodes become identical to that of the δ_{cat}-nodes in the categorization networks described in section 6.2. Thus, each δ_{cat}-node behaves exactly like the corresponding δ_{cat}-node in the categorization network.

ξ-nodes: The ξ-nodes now have six sites: **QUERY**, **RELAY**, **CP**, **HCP**, **PV**, and **INV** instead of the four sites: **QUERY**, **RELAY**, **PV** and **INV** present in the categorization networks. However, the behavior of ξ-nodes remains the same as the behavior of ξ-nodes in the categorization network. This is because, in the categorization mode, none of the δ_{inh}-nodes ever become active, and these are the only nodes that send inputs to ξ-nodes at sites **CP** and **HCP**. Consequently, only the appropriate sites of the ξ-nodes namely, **QUERY**, **RELAY**, **PV** and **INV**, take part in computations performed during the categorization mode. It is easy to see that under these conditions, the state and potential functions of ξ-nodes reduce to the state and potential functions of £-nodes specified in section 6.2.2. Thus, each £-node behaves exactly like the corresponding £-node in the categorization network.

We have established that during the categorization mode, all the £-nodes, S_{cat}-nodes, and <p-nodes of the combined network behave exactly like the corresponding nodes in the categorization network. We have also shown that the Sjj^-nodes do not play any role in this mode. Finally, because of the manner in which the categorization problem is posed, the initial state of all the |-nodes, 5_{cat}-nodes, and <p-nodes in the combined network corresponds to the initial state of the corresponding nodes in the categorization network. Thus, the combined network solves the categorization problem.

6.4 Implementation of t and I links

It was mentioned in sections 6.1.2 and 6.2.2 that unlike other links that always transmit the output of the source node, the t and I links normally remain disabled and transmit activity only when they are enabled. Furthermore, the effect of enabling the t (4) links at a £-node has a chain effect; if the t (;) links emanating at a £-node C are enabled, then, the t (i) links at all nodes that are reachable from C via t (i) links also get enabled. This section describes how this is implemented.

A t o r i link is not encoded as a simple link between two nodes, instead, it is encoded via relay nodes. Figure 6.11 illustrates the encoding; figure 6.11b shows the actual implementation of the .t and 4 links for the network described in figure 6.11a.

Each £-node C, has two relay nodes: C-l and C-t, associated with it; C-i for encoding the I links, and C-t for encoding the t links.

A relay node such as C-t receives an external input at site **ENABLE,** an input at site **OWNER** from the £-node C, and inputs at site **UPSTREAM** from all relay nodes X-↑ such that, C is a parent of X in the ordering graph defined by ℂ and ⋖. All these inputs have a weight of 1.0. The output of C-↑ goes to the site RELAY of *all* ξ-nodes Y, and to the site UPSTREAM of *all* relay nodes Y-↑ such that, Y is a parent of C in the ordering graph defined by ℂ and ⋖. The weight on the link to Y-↑ is 1.0, while the weight of the link to Y is #C/#Y.

The interconnections of a node such as C-↓ are analogous to that of the node C-↑. Thus, C-↓ receives an external input at site ENABLE, an input at site OWNER from C, and inputs from nodes X-↓, such that X is a parent of C in the ordering defined by ℂ and ⋖. The output of C-↓ goes to the site RELAY of all ξ-nodes Y, and the site UPSTREAM of all relay nodes Y-↓ such that, C is a parent of Y in the ordering graph defined by ℂ and ⋖. The weight on the link to Y-↓ is 1.0, while the weight on the link to Y is #Y/#C.

The state function and potential functions of relay nodes are as follows:

State: Node is in **active** state if it receives input at site ENABLE, or if it receives one or more inputs at site UPSTREAM. Otherwise it is in the **inert** state.

Potential: Potential equals the input at site OWNER.

6.5 Simulation

Section 6.1 through 6.4 specified the design of a massively parallel (connectionist) network that can solve the inheritance and categorization problem in time proportional to the depth of the conceptual hierarchy. A rigorous proof of correctness was also provided. In order to explicate the behavior of these networks and demonstrate the nature of inferences drawn by them, several networks of the kind described in section 6.3 have been simulated. These networks encode examples that are often cited in the knowledge representation literature as being problematic. The results of simulations demonstrate how the approach developed in this thesis deals with inheritance and categorization in an uniform manner, and solves some of the classic problems related to inheritance in presence of exceptions and conflicting information. This section describes the results of some of these simulations.

The simulation involves three stages. During the first stage, a high level description of the information to be encoded in the network is processed by a compiler (SNAIL) and translated into a set of commands to a general purpose connectionist network builder (SPIDER).The high level input to SNAIL does not depend on any aspect of the parallel implementation. During the second stage, SPIDER constructs a network in accordance with the commands generated by SNAIL. Finally, in the third stage, the activity of the network constructed by SPIDER is simulated using CISCON - a connectionist network simulator.

SNAIL and SPIDER are written in LISP, while CISCON is written in C. All the examples presented in this section were run on a VAX-750. SPIDER and the current version of CISCON have been coded by Mark Fanty and are described in [Fanty 85a][Fanty 85b].

The first example is an extension of the "quaker example" discussed in section 6.1.3. It demonstrates how the network performs inheritance in the presence of conflicting evidence. Figure 6.12 depicts the information to be encoded. There are two properties, has-bel (has-belief) with values PAC (pacifist) and NON-PAC (non-pacifist), and has-eth-org (ethnic-origin) with values AFRIC (african) and EURO (european). In broad terms, the information encoded is as follows:

"Most persons are non-pacifists"

"Most quakers are pacifists"

"Most republicans are non-pacifists"

"Most persons are of european descent"

"Most republicans are of european descent"

"Most persons of african descent are democrats"

The input to SNAIL consists of four lists:

i) a list of concepts,

ii) a list of properties and their associated values.

iii) a list specifying the $\preccurlyeq$ partial ordering where each element of the list consists of a tuple of the form (A B #A/#B), where concepts A and B are such that B is a parent of A in the ordering induced by $\preccurlyeq$ on $\mathbb{C}$, and

iv) a list specifying the distributions δ(C,P)'s that are known to the agent. Each element in this list is a tuple of the form (C P V #C[P,V]/#V #C[P,V]/#C), where V $\in$ Λ(P) for some P such that δ(C,P) is known.

The input to SNAIL based on the information depicted in figure 6.12 is as follows:

```
(NB-concept '(   PERSON P-GRP R-GRP DEMOC REPUB
                 CHRIST ZORAS QUAK MORM
                 BELIEFS PAC NON-PAC
                 ETH-ORG AFRIC EURO
                 DICK RICK SUSAN PAT))

(NB-property '((has-bel PAC NON-PAC) (has-eth-org AFRIC EURO))

(NB-is-a '(      (R-GRP PERSON 1.0)        (P-GRP PERSON 1.0)
                 (ZORAS R-GRP 0.7)         (CHRIST PERSON 0.3)
                 (QUAK CHRIST 0.1667)      (MORM CHRIST 0.8333)
```

```
                    (DICK QUAK 0.1)          (DICK  REPUB  0.0125)
                    (RICK MORM 0.02)         (RICK  REPUB  0.0125)
                    (SUSAN QUAK 0.1)         (SUSAN  DEMOC  0.0083)
                    (PAT MORM 002)           (PAT  DEMOC  0.0083)
                    (PAC BELIEFS 0.3)        (NON-PAC  BELIEFS  0.7)
                    (AFRIC ETH-ORG 0.2)      (EURO  ETH-ORG  0.8)))

(NB-delta'(         (PERSON has-bel PAC 1.0 0.3)
                    (PERSON has-bel NON-PAC 1.0 0.7)

                    (CHRIST has-bel PAC 0.4 0.4)
                    (CHRIST has-bel NON-PAC 0.26 0.6)

                    (QUAK has-bel PAC 0.12 0.7)
                    (QUAK has-bel NON-PAC 0.02 0.3)

                    (REPUB has-bel PAC 0.27 0.2)
                    (REPUB has-bel NON-PAC 0.46 0.8)

                    (DEMOC has-bel PAC 0.73 0.37)
                    (DEMOC has-bel NON-PAC 0.54 0.63)

                    (PERSON has-eth-org AFRIC 1.0 0.2)
                    (PERSON has-eth-org EURO 1.0 0.8)

                    (REPUB has-eth-org AFRIC 0.125 0.0625)
                    (REPUB has-eth-org EURO 0.47 0.9375)

                    (DEMOC has-eth-org AFRIC 0.875 0.29)
                    (DEMOC has-eth-org EURO 0.53 0.71)))
```

On the basis of its input, SNAIL generates commands to SPIDER to create the required ξ-nodes, φ-nodes, δ_{inh}-nodes, δ_{cat}-nodes, and relay nodes. SNAIL also generates commands that instruct SPIDER to connect various nodes in accordance with the interconnections described in sections 6.1 through 6.4.

In addition to the nodes discussed so far, the simulations also make use of *driver* nodes. A driver node is associated with each ξ-node, φ-node, and relay node, and each of these nodes receive an input from their associated driver node at the site QUERY. This input acts as an *external* input that can be turned ON or OFF by simply turning the driver node ON or OFF. Thus, a query may be posed to the network by turning ON the appropriate driver nodes.

As our first example, consider the inheritance query:

"Is Dick a pacifist or a non-pacifist"

In other words: V-SET = {PAC, NON-PAC}

C = DICK; P = has-bel; REF = BELIEFS

This query is posed by activating (turning ON) INHERIT, d-DICK, and d-has-bel and the activity of PAC and NON-PAC is observed for 5 time steps (it was specified in section 6.1.3 that one must wait for 3 time steps, the extra delay is caused by the presence of driver nodes.) Both PAC and NON-PAC remain **INERT** with a potential of -1.0. (A potential of -1.0 means that the node is inert.) Consequently, the nodes d-bu-DICK, d-BELIEFS, and d-td-BELIEFS are activated.

The trace of potential of some relevant nodes is shown in the following table:

CLOCK:	9	10	11	12	13	14	15
PERSON	<------------- 0.0 ----------------->				0.005	0.005	0.25×10^{-4}
CHRIST	<--- 0.0 ---->		<----------------- 0.017				--------------------->
QUAK	<---------------------------			0.01	-------------------------------->		
REPUB	<---------------------------			0.0125		--------------------------->	
PAC	0.0	0.3	<---0.00972---->		<---0.12×10^{-2}-->		0.00972
NON-PAC	0.0	0.7	<---0.00644---->		<---0.13×10^{-3}-->		0.00644

The potentials remain unchanged after the 15th cycle, and the final potentials of PAC and NON-PAC for DICK are:

DICK

Value	Raw potentials	Normalized potentials
PAC	0.00972	1.00
NON-PAC	0.00644	0.66

Thus, the ratio of the likelihoods of Dick being a pacifist and Dick being a non-pacifist is about 3:2. Thus, on the basis of available information , Dick who is a republican and a quaker is more likely to be a pacifist.

Similar simulations for RICK, PAT, and SUSAN lead to the following results.

RICK

Value	Raw potentials	Normalized potentials
PAC	0.03240	0.39
NON-PAC	0.08372	1.00

Rick who is a mormon republican is more likely to be a non-pacifist.

PAT

Value	Raw potentials	Normalized potentials
PAC	0.08760	0.89
NON-PAC	0.09828	1.00

Pat who is a mormon democrat is also more likely to be a non-pacifist, but only marginally so.

SUSAN

Value	Raw potentials	Normalized potentials
PAC	0.02628	1.00
NON-PAC	0.00756	0.29

Finally, Susan who is a quaker democrat is a pacifist with a very high probability.

As an example of categorization, consider the query:

"among the following persons, who is most likely to be a pacifist and of african descent: DICK, RICK, SUSAN, PAT".

in other words: C-SET = {DICK, RICK, SUSAN, PAT}

REF = PERSON

DISCR = { [has-bel PAC] [has-eth-org AFRIC] }

The query is posed by activating (turning ON) the nodes CATEGORIZE, d-PERSON, d-td-PERSON, d-has-bel, d-has-eth-org, d-PAC, and d-AFRIC. {The nodes with prefix "d-" refer to driver nodes. For example, d-PERSON is the driver node for PERSON and d-td-PERSON is the driver node for td-PERSON, which in turn is the top-down (↓) relay node associated with PERSON).

The network computes the solution in 12 steps (the depth of the network is 4). The most likely person who is a pacifist and is of african descent is SUSAN followed by DICK, PAT, and RICK.

The final potentials of DICK, RICK, SUSAN, and PAT are:

[has-bel PAC][has-eth-org AFRIC]

Name	Raw potentials	Normalized potentials
DICK	0.00000004	0.11
RICK	0.00000002	0.05
SUSAN	0.00000037	1.00
PAT	0.00000021	0.57

As would be expected, Susan who is a quaker and a democrat best matches the description "person of african descent with pacifist beliefs". The least likely person turns out to be Rick. The latter seems intuitively correct given that Rick is a republican and a mormon (Rick is neither a democrat which correlates with african origin, neither he is a quaker which correlates with pacifism).

In a similar fashion, the query for finding the most likley person who is a non-pacifist and is of european descent results in the following potentials:

[has-bel NON-PAC][has-eth-org EURO]

Name	Raw potentials	Normalized potentials

DICK	0.00000450	0.50
RICK	0.00000900	1.00
SUSAN	0.00000267	0.30
PAT	0.00000535	0.59

A query for finding the most likely person who is a pacifist leads to the following potentials:

[has-bel PAC]

Name	Raw potentials	Normalized potentials
DICK	0.00000350	0.54
RICK	0.00000200	0.31
SUSAN	0.00000645	1.00
PAT	0.00000369	0.57

As a second example, the information depicted in figure 6.13 was encoded in a network. This corresponds to a popular example that is often cited in the knowledge representation literature as a difficult problem of inheritance [Etherington & Reiter 83][Fahlman 81]. In this example, the concepts have the property epidermis-type associated with them. The values of this property are: SHELL, SKIN, FUR, and FEATHER.

This information about S(C,P)'s may be paraphrased thus:

Most Molluscs are shell-bearers.
Cephalopods are Molluscs, but most Cephalopods are not shell-bearers
Nautili are Cephalopods, and all Nautili are shell bearers.

The final potentials of the nodes SHELL and SKIN as a result of the inheritance of the property epidermis-type for MOLLUSC (or MOLLI), CEPHALOPOD (or PODI), and NAUTILUS (or CREFTI) are given below (the potentials of FUR and FEATHER were consistently 0.0 in each case):

MOLLUSC (or MOLL1)

Value	Raw potentials	Normalized potentials
SHELL	0.087500	1.00
SKIN	0.037485	0.43

Thus MOLL1 is more likely to be a shell-bearer which is consistent with the agent's belief that most molluscs are shell-bearers

CEPHALOPOD (or PODi)

Value	Raw potentials	Normalized potentials
SHELL	0.006250	0.25
SKIN	0.025020	1.00

That is, POD1 is not likely to be a shell-bearer, which is in agreement with the agent's belief about cephalopods.

NAUTILUS (or CREFT1)

Value	Raw potentials	Normalized potentials
SHELL	0.006250	1.00
SKIN	0.000000	0.00

CREFT1 is definitely a shell-bearer. Notice that the likelihood of CREFT1 having an epidermis-type other than Shell-bearer computes to 0.00 which is exactly what shold be expected given that ALL nautilus are shell bearers.

Finally, a large network was constructed that included the two smaller network described above as sub networks. The large network had 75 concepts, 5 properties, and 30 S(C,P)'s. The depth of the network was 11. A total of 632 nodes and 1591 links were required to encode this network. The time taken to construct this network was around 40 minutes and the time taken to perform a single step of simulation was just under a second (all times are elapsed times). A description of the conceptual hierarchy underlying this example is given in figures 6.14a, 6.14b, and 6.14c, while the known distributions are specified in figures 6.15a through 6.15e.

When any of the queries posed to the "quaker network" and the "mollusc network", were posed to the large network, the answers obtained were the same as those that were obtained by posing the query to the smaller networks.

The following results were obtained when the query "which of the following ANIMAL is most likely to have epidermis type SKIN and habitat LAND: MLUSC REPTILE BIRD ELEPHNT PERSON"

[epidermis-type SKiN][has-habitat LAND]

Name	Raw potentials	Normalized potentials
MOLLUSC	0.01000000	0.02
REPTILE	0.09332800	0.21
BIRD	0.03547398	0.08
ELEPHANT	0.02755139	0.06
PERSON	0.44444799	1.00

Thus, a land dweller with epidermis type skin is most likley to be a person.

When the query sought the answer for epidermis type SKIN and habitat WATER, the following results were obtained:

[epidermis-type SKIN][has-habitat WATER]

Name	Raw potentials	Normalized potentials
MOLLUSC	0.05333334	1.00
REPTILE	0.01110867	0.21
BIRD	0.00682228	0.13
ELEPHANT	0.00091110	0.02
PERSON	0.00000000	0.00

Thus, a water dweller with epidermis-type skin is most likely to be a mollusc.

In evaluating these answers it must be borne in mind that they are based on the information encoded in the network and if the information does not capture the readers intuitions then neither will the answers computed by the network.

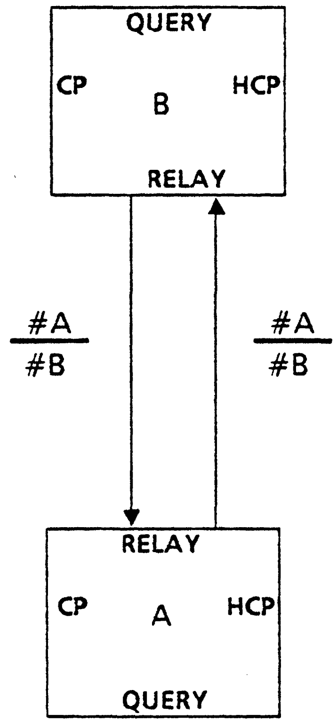


FIGURE 6.1 Parallel encoding for inheritance - 1

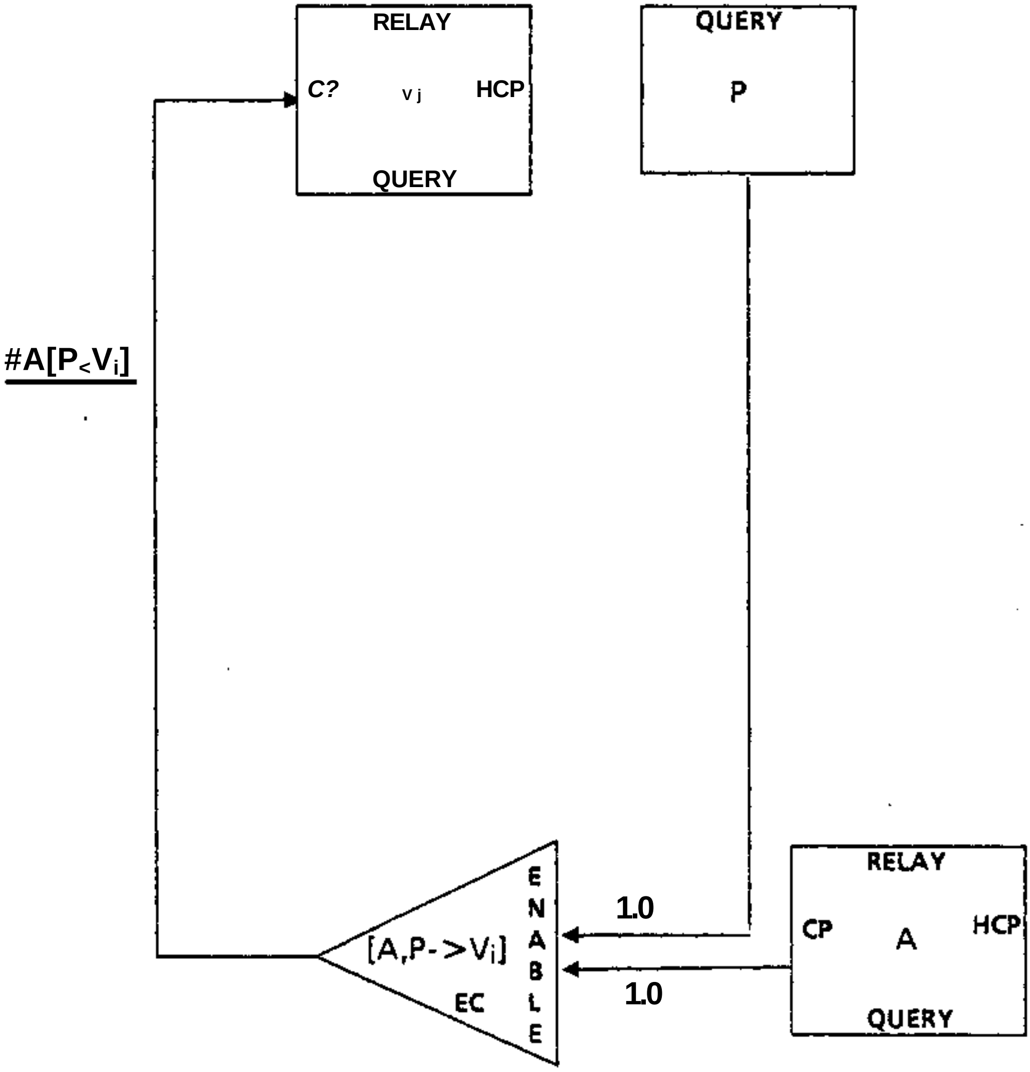


FIGURE 6.2 Parallel encoding for inheritance-II

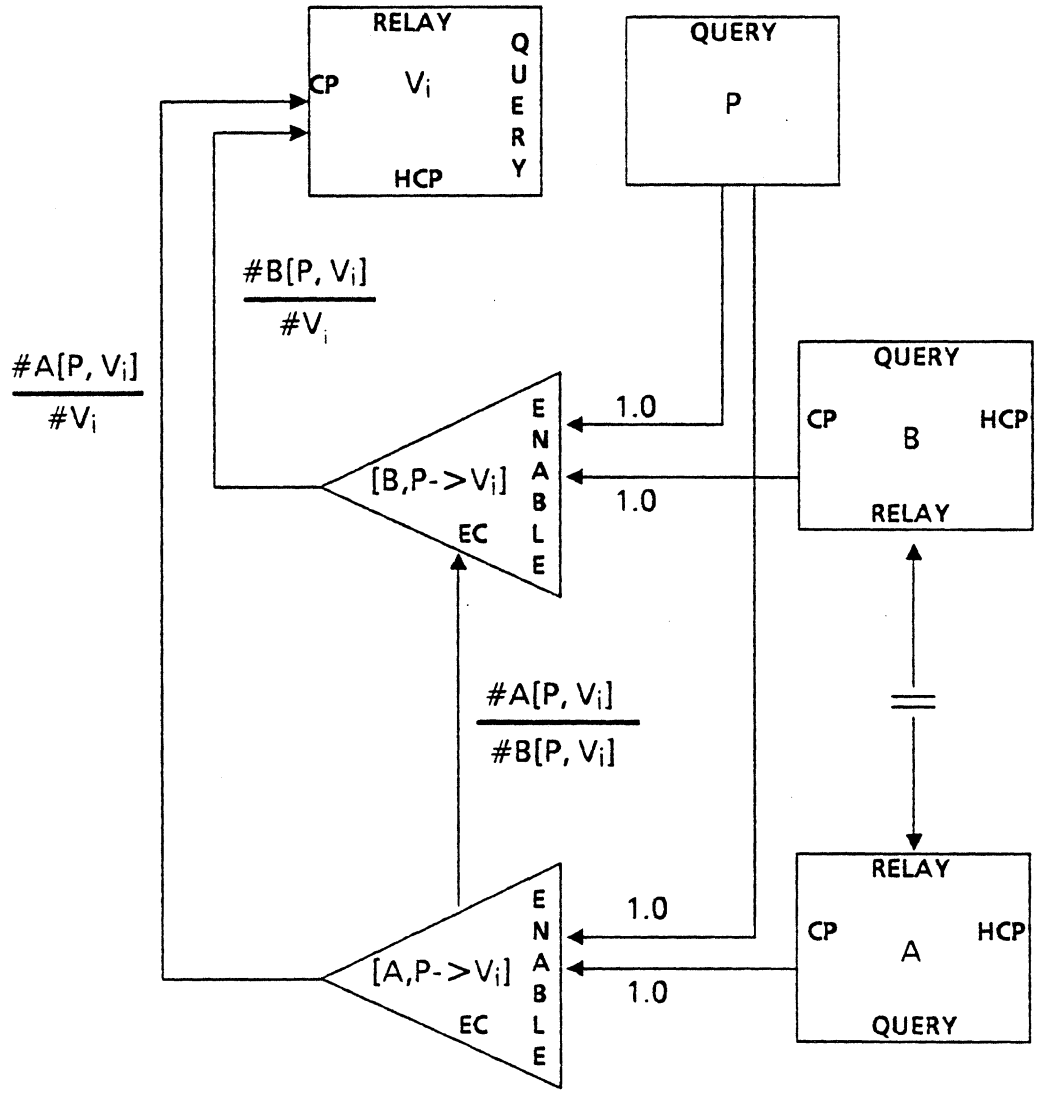


FIGURE 6.3 Parallel encoding for inheritance - III

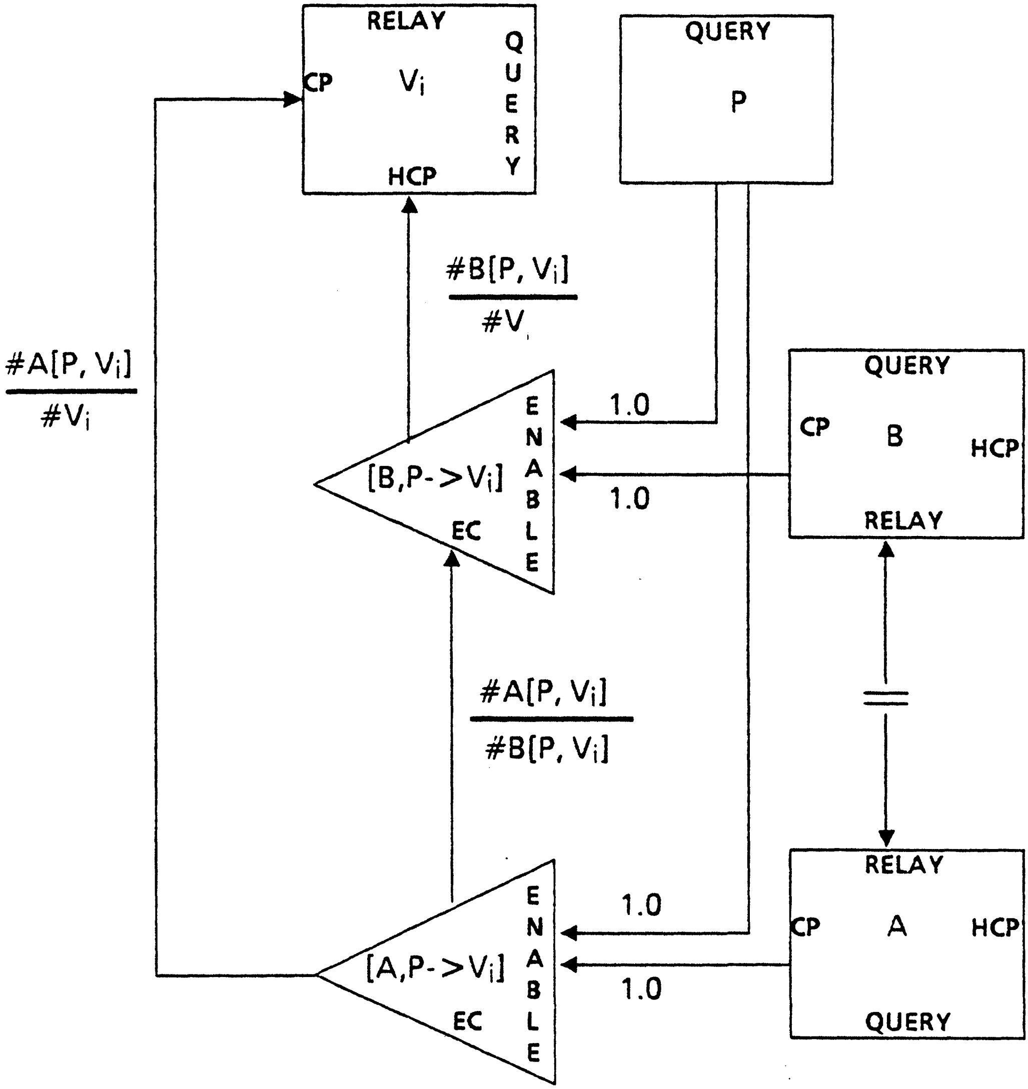
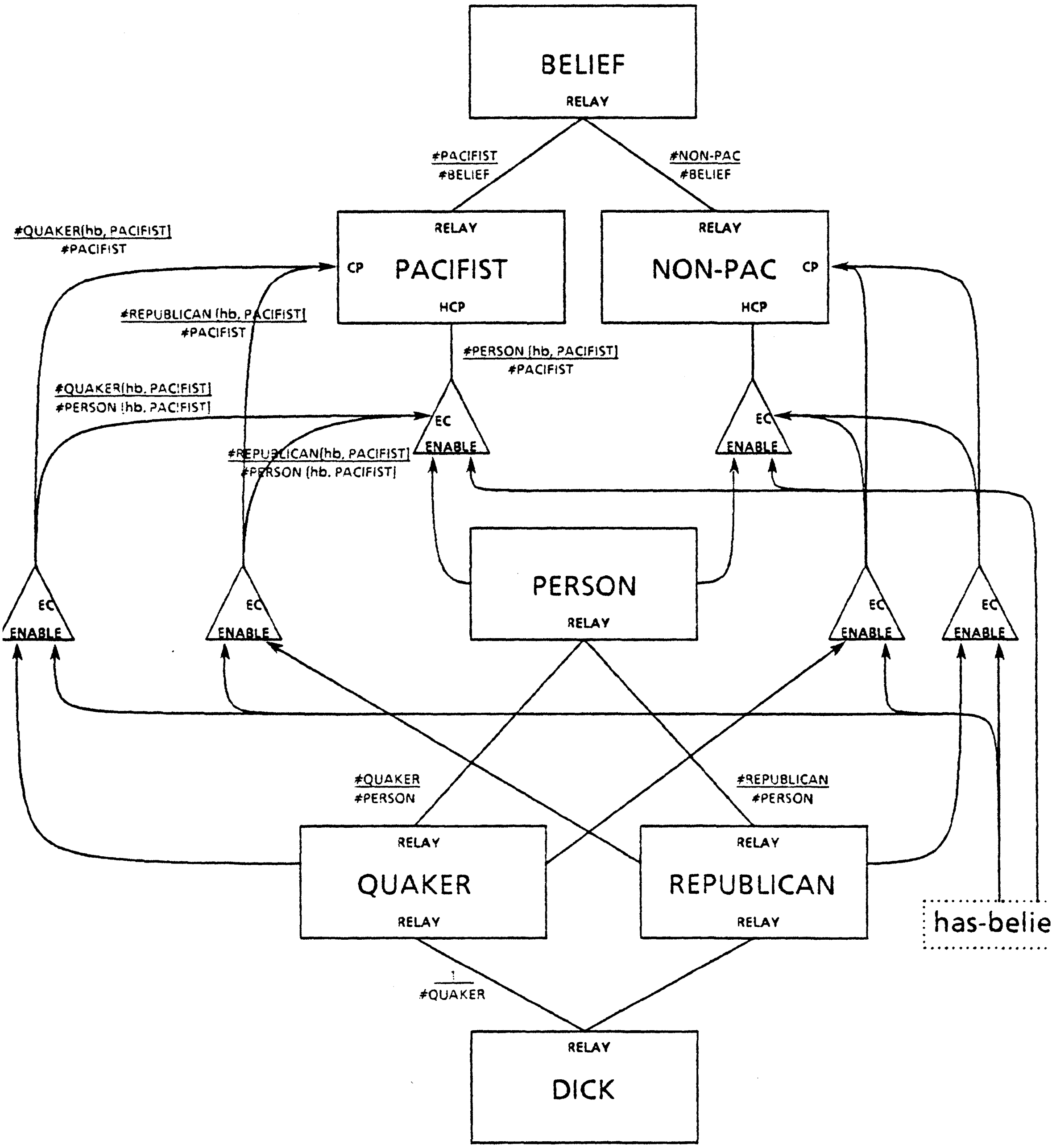


FIGURE 6.4 Parallel encoding for inheritance - IV



All inputs incident at site enable of δ-node have a weight of 1.0.
Not all sites and weights have been shown.
hb = has-belief

FIGURE 6.5 An example of inheritance

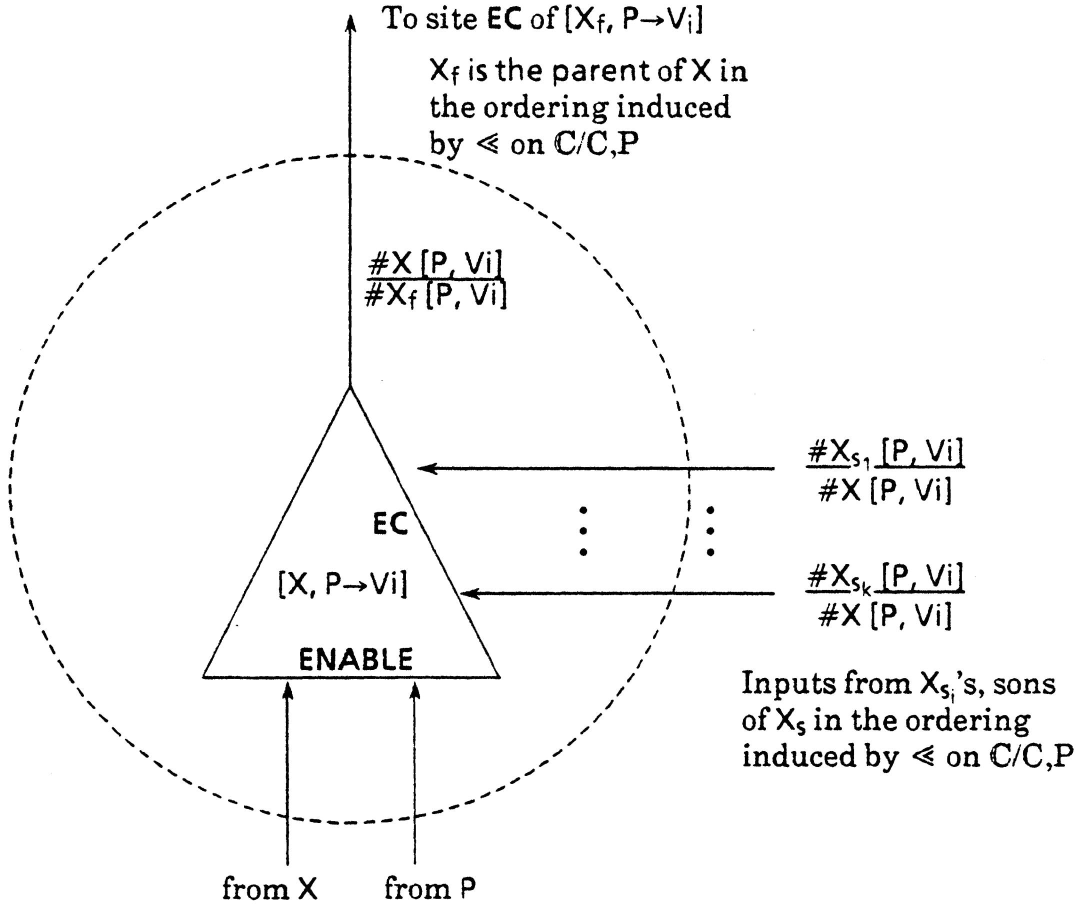
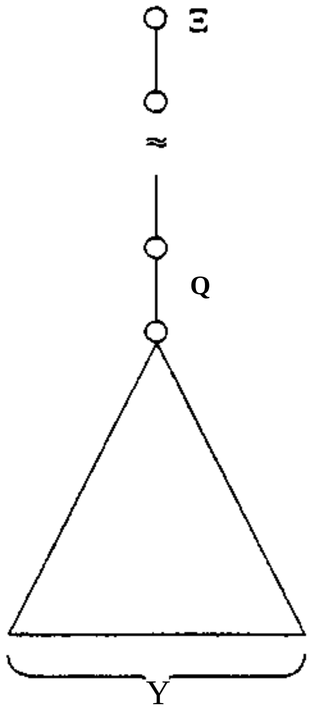


FIGURE 6.6 Computation at a δ_{inh} node



Z is such that there is no D for which Z <3D and 8 (D, P) is known.

The lowest level consists of Bj € *T* (C, P).

FIGURE 6.7 Relevant concept

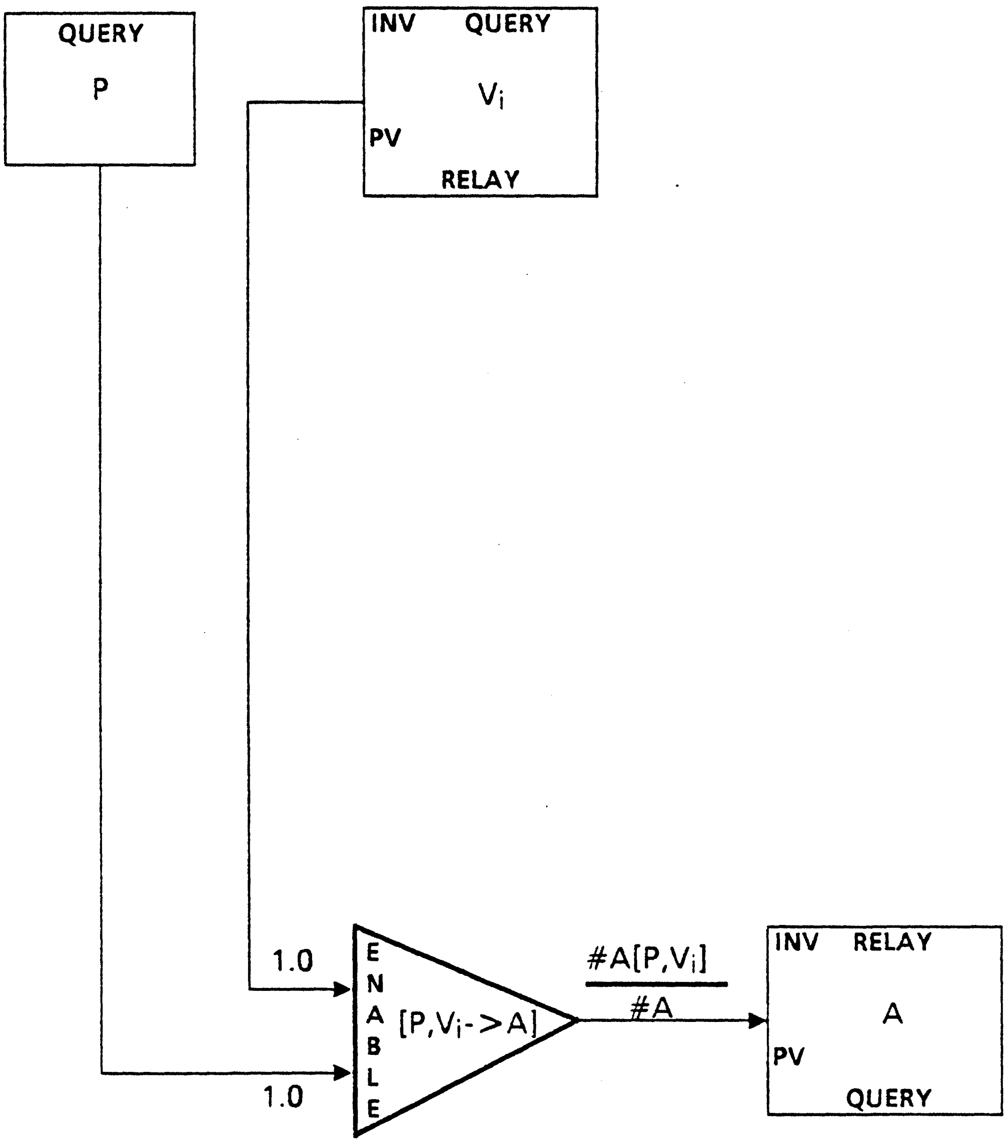


FIGURE 6.8 Parallel encoding for categorization - I

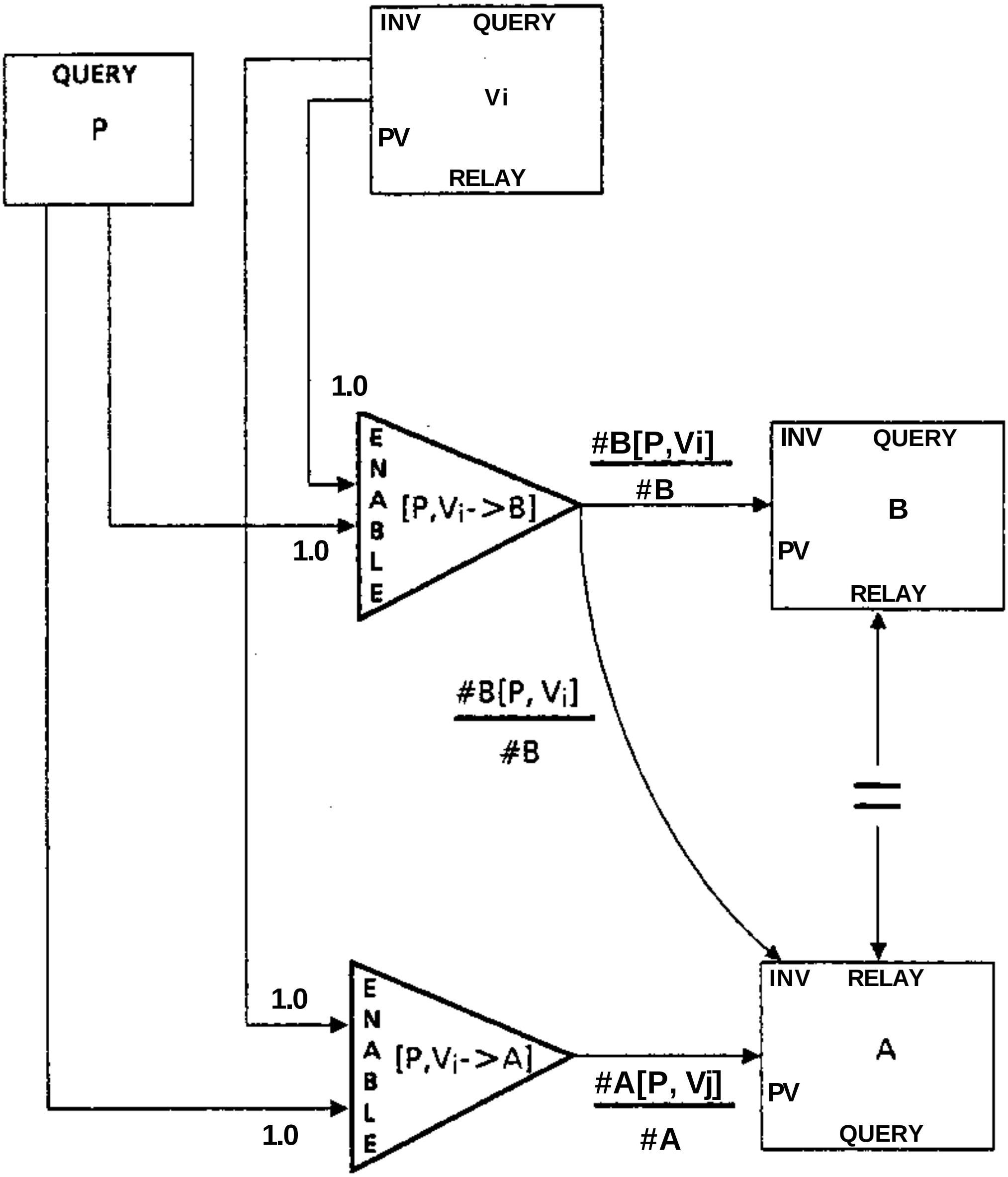
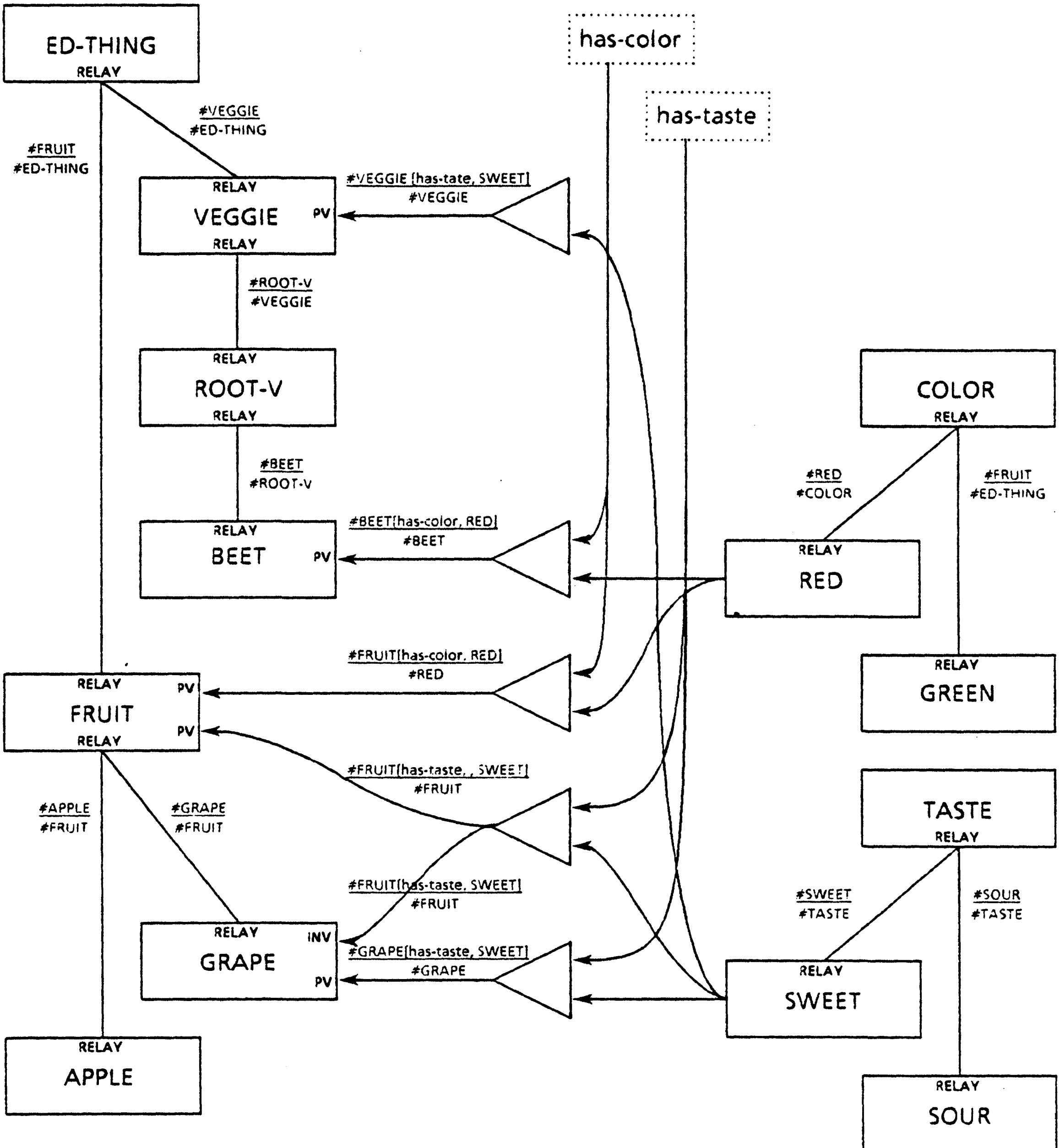


FIGURE 6.9 Parallel encoding for categorization - II



Not all sites have been marked.

FIGURE 6.10 An example of categorization

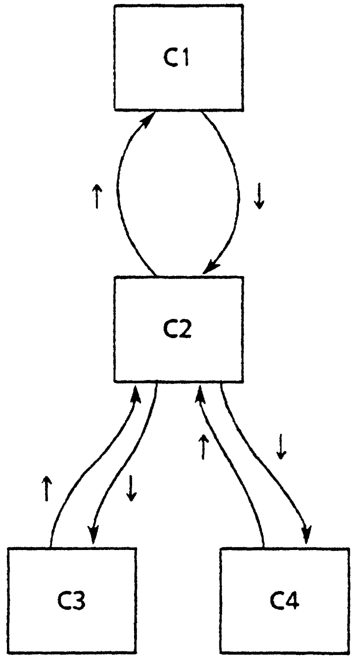
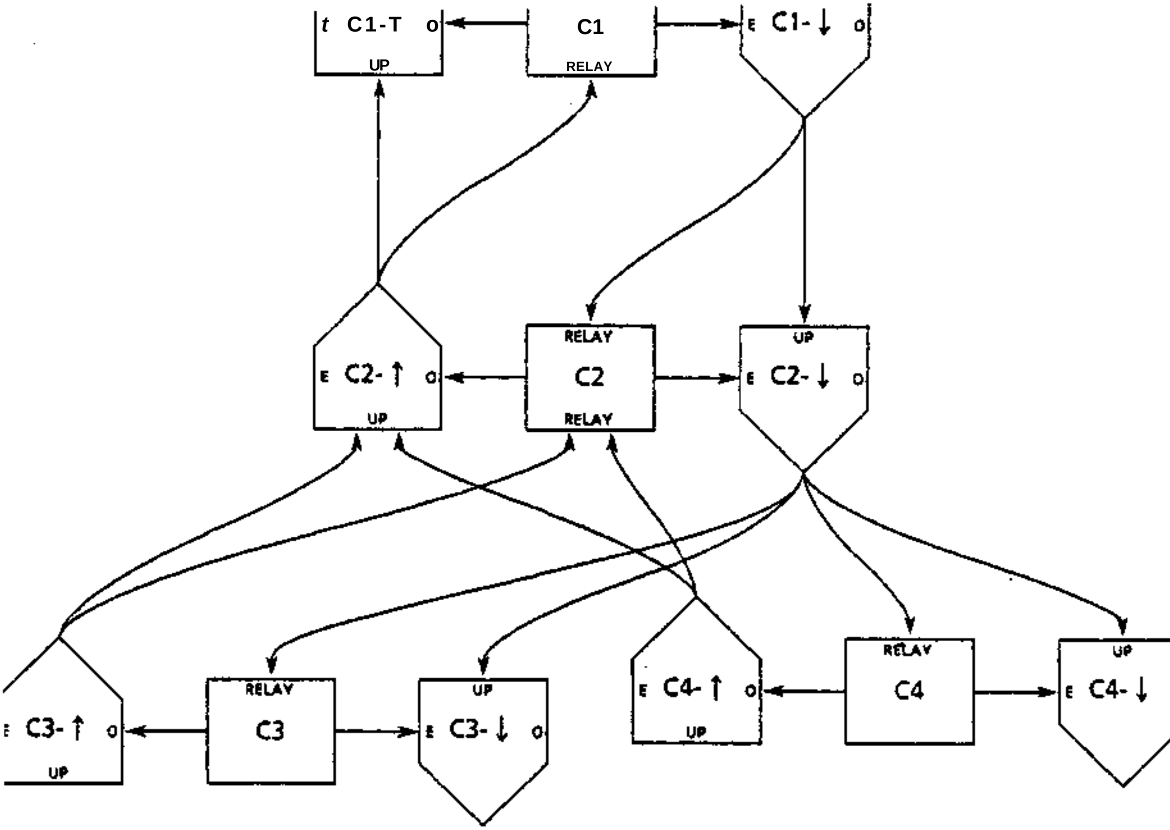


FIGURE 6.11a Examples of ↑ and ↓ links



E: **ENABLE**
O: **OWNER**
UP: **UPSTREAM**

The site RELAY in node C2 has been duplicated for ease of illustration.

FIGURE6.11b Encodingof t and 1 links

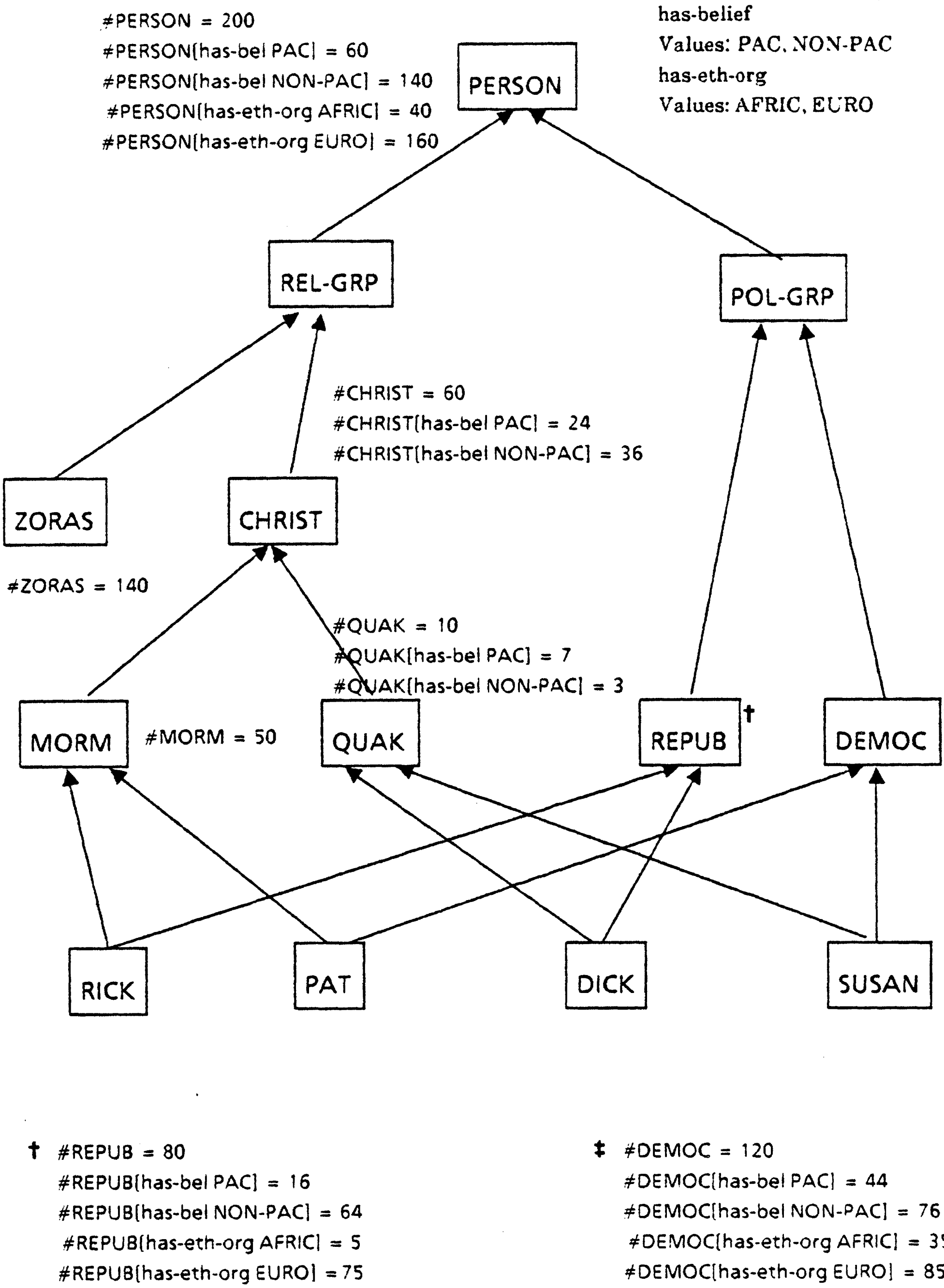


Figure 6.12 The quaker example

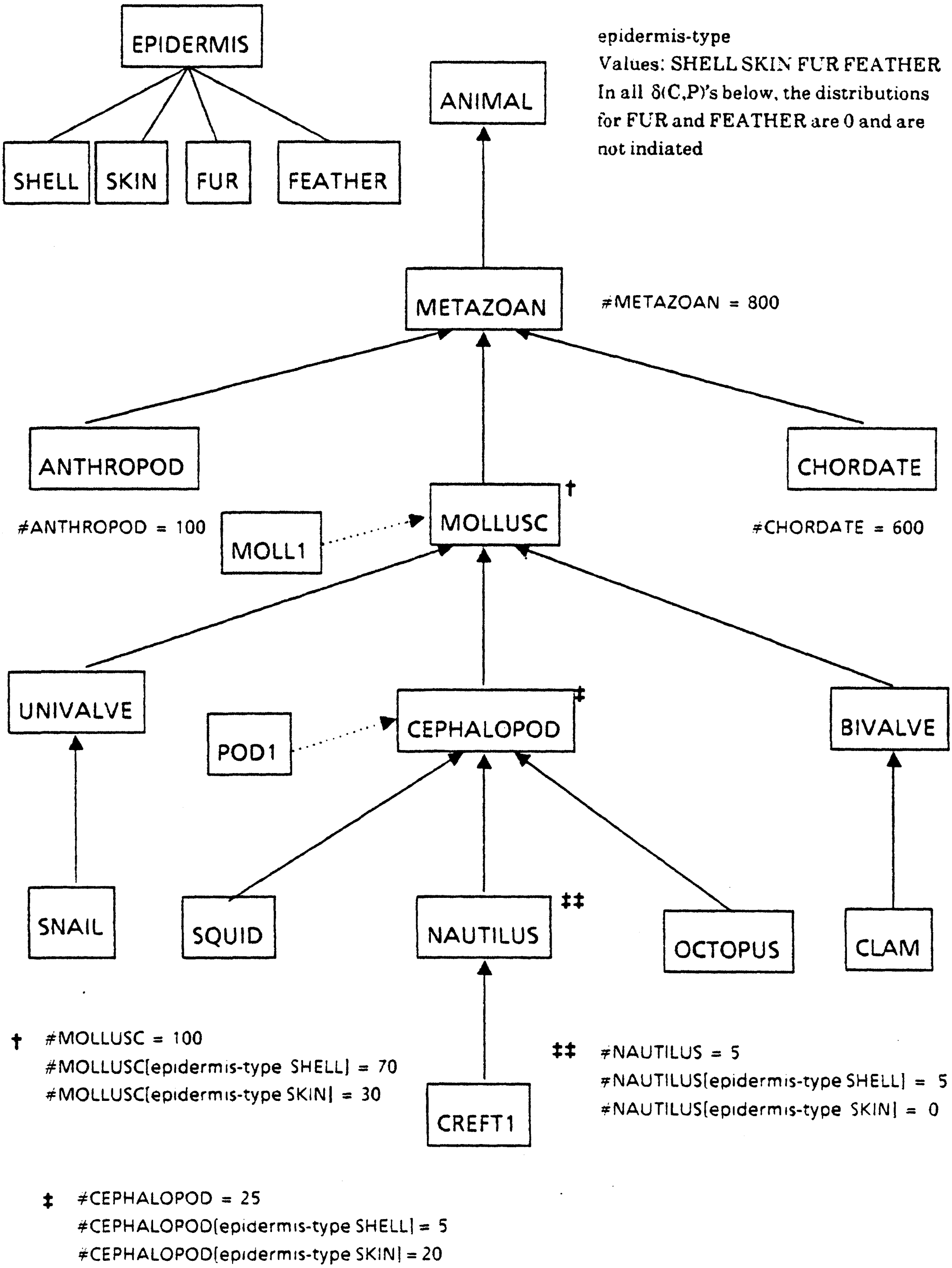
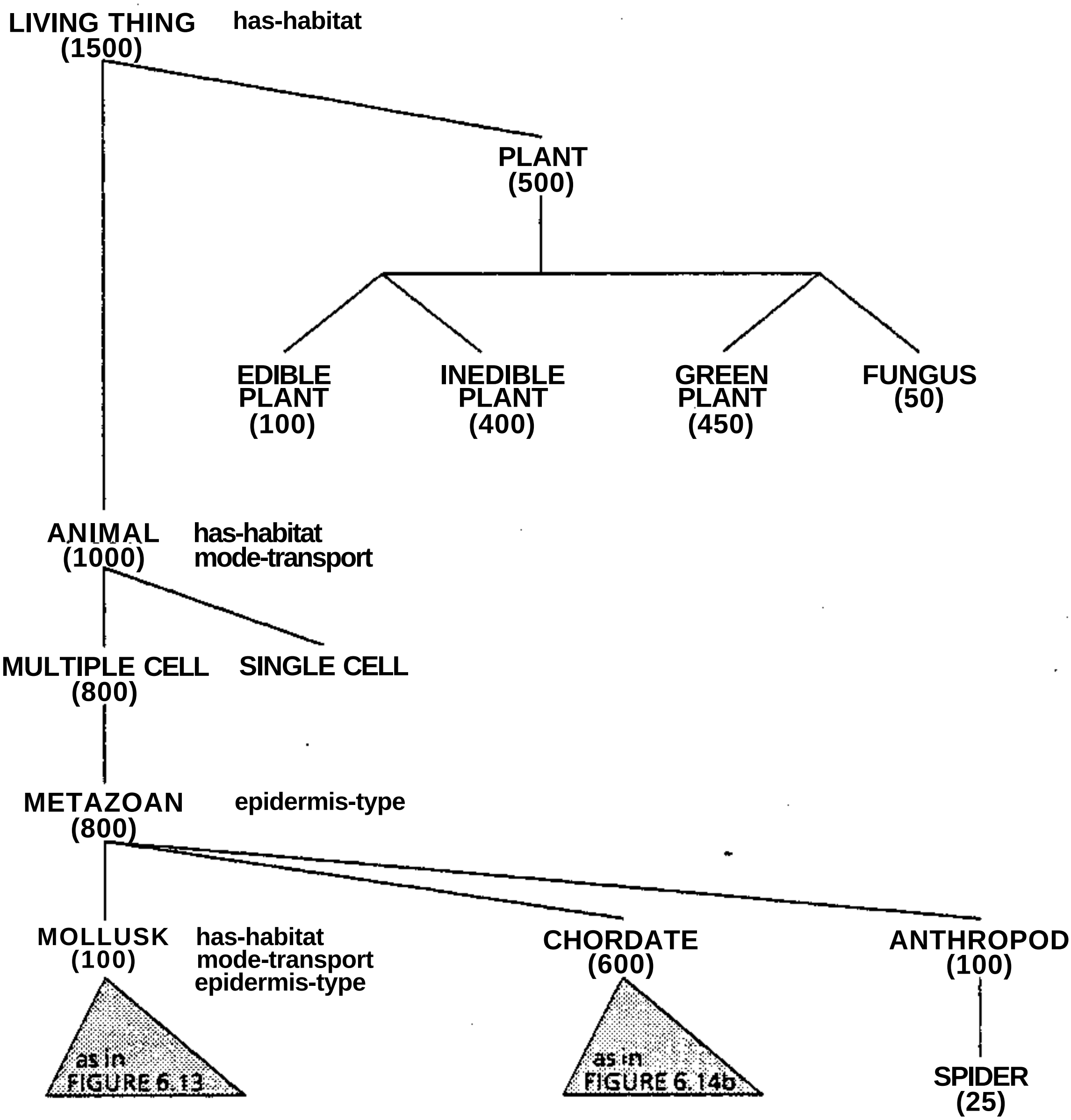


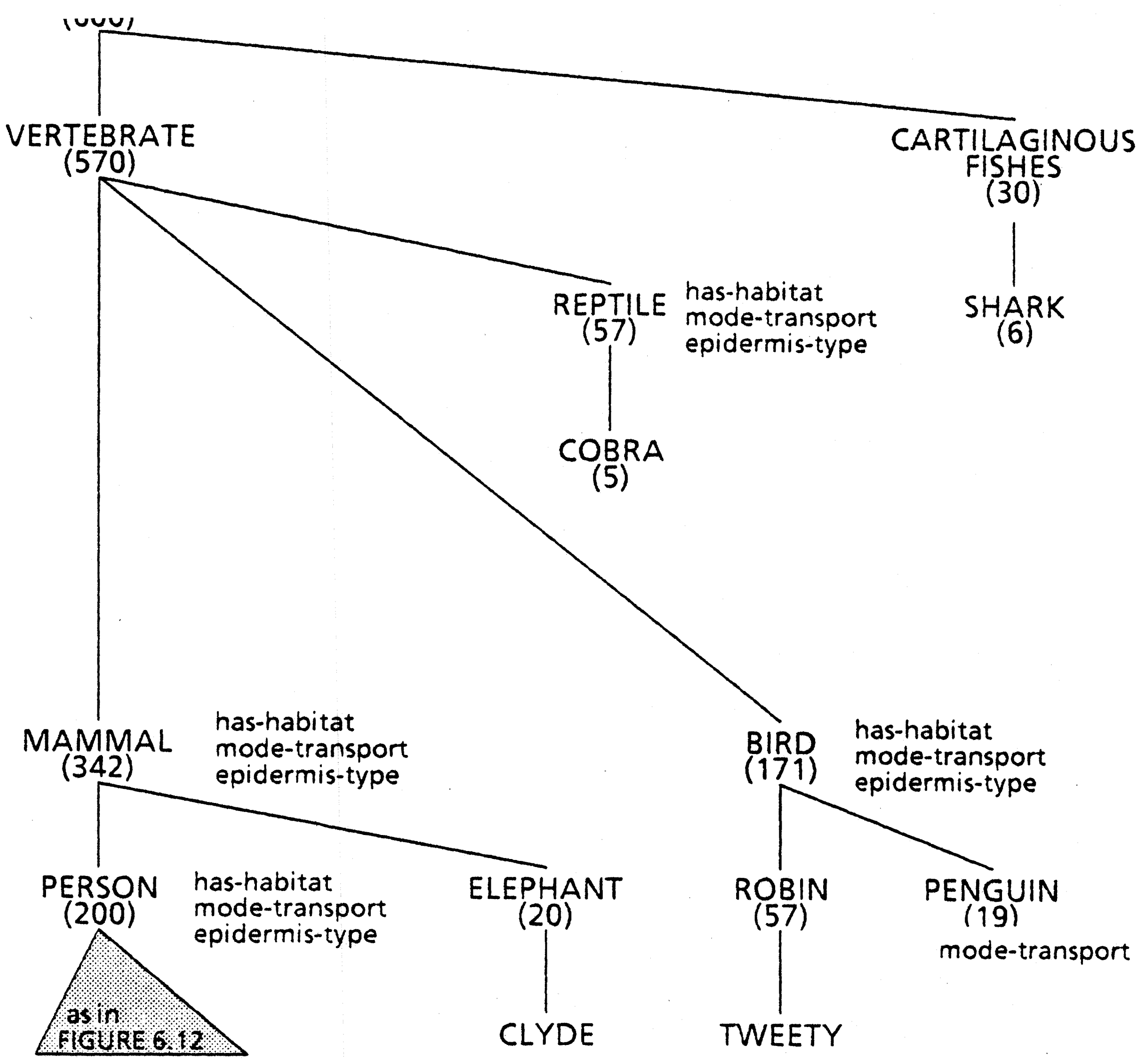
Figure 6.13 The mollusc example



Properties listed with concepts are those for which the distributions are known.

Numbers listed with concepts refer to #C.

FIGURE 6.14a Organism hierarchy -I



Properties listed with concepts are those for which the distributions are known.
Numbers listed with concepts refer to #C.

FIGURE 6.14b Organism hierarchy - II

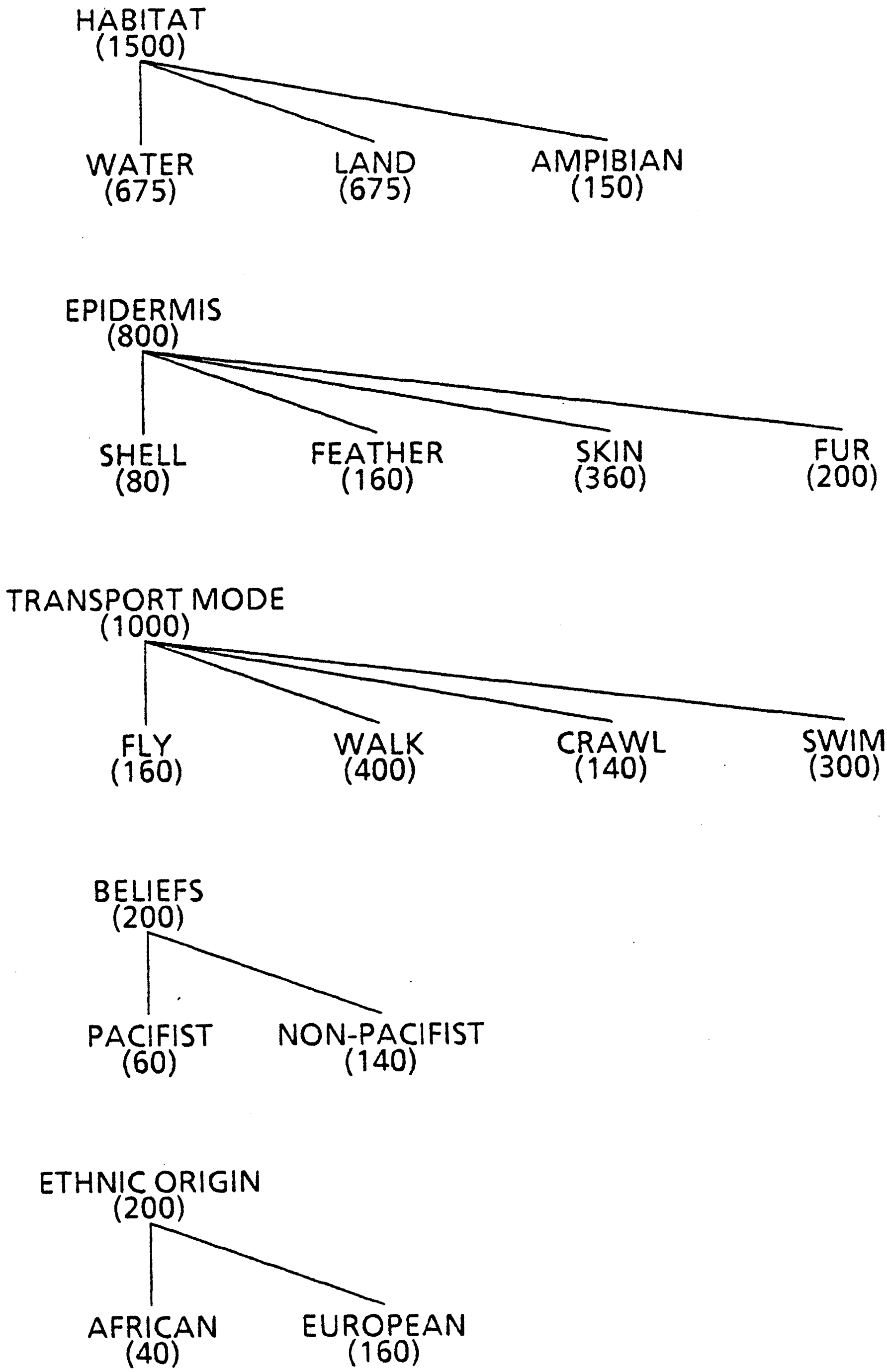


FIGURE 6.14c Organism hierarchy - III

	SHELL	FEATHER	SKIN	FUR
METAZOAN	30	160	360	200
MAMMAL	0	0	240	102
PERSON	0	0	200	0
BIRD	0	150	21	0
REPTILE	0	0	57	0
MOLLUSC	70	0	30	0
CEPHALOPOD	5	0	20	0
NAUTILUS	5	0	0	0

FIGURE 6.15a Distribution w.r.t. epidermis type

	FLY	WALK	CRAWL	SWIM
ANIMAL	160	400	140	300
MOLLUSC	0	0	15	85
REPTILE	5	17	25	10
BIRD	140	16	0	15
PENGUIN	0	9	0	10
MAMMAL	0	300	0	42
PERSON	0	180	0	20

FIGURE 6.15b Distribution w.r.t. mode of transport

	WATER	LAND	AMPHIBIAN
LIVING THING	675	675	150
MOLLUSC	80	15	5
BIRD	25	130	16
MAMMAL	10	302	30
PERSON	0	200	0
ANIMAL	450	450	100
REPTILE	5	42	10

FIGURE 6.15c Distribution w.r.t. habitat

	PACIFIST	NON-PACIFIST
PERSON	60	140
CHRISTIAN	24	36
QUAKER	7	3
REPUBLICAN	16	64
DEMOCRAT	44	76

FIGURE 6.15d Distribution w.r.t. beliefs

	AFRICAN	EUROPEAN
PERSON	40	160
REPUBLICAN	5	75
DEMOCRAT	35	85

FIGURE 6.15e Distribution w.r.t. ethnic origin

Chapter 7

Discussion

This chapter discusses some limitations of the research described in the previous chapters and lists some issues that remain unresolved. It also indicates possible directions that one may take in pursuing the line of research described in this thesis.

7.1 Representation issues

In this thesis I have focused on the nature of information underlying inheritance and categorization, proposed an evidential framework for formalizing this form of reasoning, and presented an efficient and massively parallel implementation of the resulting formalism. In other words, the traditionally distinct areas of representation, inference, and computation have been dealt with simultaneosuly. In order to maintain rigor, my strategy has been to consider only a restricted class of representational issues and there remain several issues that have not been addressed. These include representation of actions, events, complex shapes, definition of composite relations, finer structure of properties and values, and the relationship between property values of a concept (analogous to structural descriptions of KLONE[Brachman 79]). This section discusses some of the questions that are important from an evidential viewpoint.

7.1.1 Relationship between property-values of a concept

The question of representing interrelations between property values is particularly relevant to our concerns. How does one represent the belief that "most red apples are sweet" and use it when reasoning about the taste of an apple given that it is red in color. In other words, if the correlation between two property-values of a concept C is known (i.e. if $\#C[P_1,V_1][P_2,V_2]$ is known), then how can this information be utilized in reasoning about C. The evidential formulation developed in section 5.3 demonstrates that as long as the knowledge about concepts and their property-values is in the form of #C[P,V]'s, there exists an efficient way of utilizing this knowledge. In section 5.2.2 we saw how a limited amount of knowledge outside this form may be incorporated. However, the computations soon become too complex and computationally intractable. One way of representing such information is to posit the existence of "intermediate concepts[11]. For example, in order to use the information "most red apples are sweet", one would either posit the concept "red apple" and attach the appropriate information about the taste of such apples, or alternately posit the existence of the concept "sweet apple" and attach the information about the color of such apples. This does not imply that all such intermediate concepts have to be linguistic terms, however it does entail that appropriate computational machinery (concept nodes and binder nodes) has to be dedicated to make this information accesssible during inheritance and categorization. An extension of the above idea directly leads to the following conjecture: if information about correlations between property values of a concept is available, and if this information is significant and relevant, then *new* concepts are created in order to use this information *effectively*. This provides a computational explanation for the creation of new concepts and suggests that the goal of a concept formation (learning) mechanism should be to create concepts - and the ensuing Type structure, such that most of the distribution information may be expressed in terms of #C[P,V]'s of existing concepts. In more general terms, the goal of a concept formation mechanism should be to create concepts that allow significant information to play a role in the inference process. This also explains why one may have a lot of information that does not get used in actual reasoning or decision making. *Having information is not enough to guarantee its utilization; for it to be used in the reasoning process, it must occupy an appropriate place in the conceptual structure.*

The above requirement should not seem unreasonable if one observes that in real life, if a large amount of *significant* information about some concept is available, the concept is subdivided in order to better encode this information. All specialized domains have numerous concepts to express that which is a single concept in common parlance. Thus, whereas we may be happy to apply the term "rock" to a large class of "relatively hard naturally formed mass of mineral or petrified matter", a geologist has numerous concepts to capture the subtle distinctions and interrelations in the properties of such substances. These distinctions would be difficult to express and equally difficult to conceptualize unless they were encapsulated into appropriate concepts.

What remains to be specified is a theory of concept formation that can identify the concepts (Types) that must arise if relevant information is to be utilized. It was stated in section 3.4.2 that concepts record similarities between objects and their properties, and make this information accessible for categorizing novel objects, and predicting their property-values (i.e. concepts provide the basis for categorization and inheritance). This suggests that a criteria for creating concepts may be based on an information-theoretic measure that takes into account the effectiveness of a particular conceptual structure in solving categorization and inheritance problems, (in other words, a measure that rates different conceptual structures vis a vis their ability to categorize objects given their property-values, and their ability to predict the property-values of an object given its category). For some ongoing work in this direction refer to [Gluck & Corter 85].

7.1.2 Finer structure of property-values

A simplifying assumption made in this work is that all property values are disjoint, for example, "red" and "green" are assumed to be disjoint values of the property has-color. This is not always the case, as there are property-values that do not fit this simplistic picture. Some property-values may be composites of two or more property-values. For example, the taste "sweet and sour" is a combination of "sweet" and "sour".

Furthermore, property-values may be organized in an hierarchical manner; an artist would tell us that there are many different kinds of "red", each of which may be treated as a subtype of "red". Even in common usage, "pink" is at a lower level of abstraction than "red". It is not clear how evidence for "red" should be distributed among its various subtypes. This seems to be more of a question of *typicality* than frequency of occurrence. If one observes the range of the visual spectrum that is classified as "red", it may range from what would be described as "blood-red" to "pink", and colors within this range will be more typical or less typical instances of "red". One would like to distribute evidence for "red" according to some measure of typicality. Among existing proposals, the work on fuzzy logic by Zadeh [Zadeh 84] seems relevant for dealing with the notion of typicality. In the framework of fuzzy logic, RED would be a fuzzy predicate and different instances of "red" would have varying degrees of "redness" associated with them; thus Rose may be red with degree 0.8 while Brick may be red only with a degree 0.7. The major criticism of fuzzy logic is the manner in which certainty factors are combined when analyzing compound terms. It handles disjunctions by taking the maximum, and conjunctions by taking the minimum. Such a combination rule often leads to conuterintuitive results. For example, a guppy may be a very atypical fish, and also a very atypical pet, however this should not necessarily entail that guppy is a very atypical pet fish.

An alternative to the use of fuzzy predicates is the use of "exploded values". It may be argued that property values are much more fine grained than their "names" might suggest. Normal usage of language often belies the complexity of the information being communicated. In some cases detailed information may not be articulated as it is not relevant to the situation.

However, oftentimes, a speaker does not make certain distinctions because he relies upon the hearer to make these by using his world knowledge. For instance, while refering to the color of an apple and that of a brick as "red" one seldom means that they are one and the same color. One assumes that the hearer is aware of the difference between the two colors and hence will be able to interpret the two usages of "red" appropriately. In view of the above we ought to use exploded color values such as APPLE-RED, ROSE-RED and BRICK-RED. It is important to make these distinctions in a knowledge representation scheme inspite of the surface uniformity of language. Traditional knowledge representation systems do not have to represent these distinctions explicitly as they can shift this burden to the interpreter; the interpreter may be programmed to treat differently the value "red" when it is associated with distinct objects. The absence of an interpreter in the present formulation, however, makes it necessary to explicitly represent concepts in a finer grain. The relationship between concepts such as APPLE-RED and RED may be the same as that between RED and COLOR, and the properties associated with color - *HUE*, *BRIGHTNESS* and *SATURATION* - may be used to make classifications like RED and GREEN and also to make finer distinctions like BRICK-RED and APPLE-RED. These issues are addressed within a connectionist framework in [Cottrell 85].

7.1.3 Representation of relations and events

In section 4.1 it was explained how non-evidential property values are special cases of the general evidential construct 5. Thus, relations may be represented using the graphical notation described in section 3.6 and suppressing the evidential weights on links (for example all weights may be assumed to have the weight 1.0). Figure 7.1 shows the representation of the predicate LOVES. It is easy to see the similarity in the notion of properties as used in this formulation and case roles that denote relations between predicates and noun phrases [Bruce 75; Fillmore 68]. The simplified representation in Figure 7.1 suggests that a PREDICATE has two case roles namely, *HAS-AGENT* and *HAS-PATIENT*. For the more specific predicate LOVES these case roles get mapped into *HAS-LOVE-AGENT* and *HAS-LOVE-PATIENT* which in turn are filled by JOHN MARY in the representation of "John loves Mary".

In a similar fashion, the network in figure 7.2 encodes the following information:

"ON is a kind of spatial relation.

ON has two arguments: the thing on top and the thing at the bottom.

A is a ball and B is a cube.

A is on B."

In general, a relation or its instance is similar to a concept such as an apple or bird, and the arguments of the relation are analogous to properties of an object. Thus, the representation of a two place relation such as ON may be characterized as an object with two properties (arguments): *on-top* and *on-bottom.*

Many relations such as PARENT-OF and ON either hold or do not hold and hence, their representation does not require an evidential treatment. However, there are many relations that are best viewed as graded relations (in a manner similar to various grades or degrees of "redness" or "bigness"). AN example of such a relation is LIKES; as in "John likes Mary". There are at least two ways in which a degree of strength may be associated with the representation of this relation. First, "liking" itself may have a degree of strength associated with it; John may "like Mary a lot" or "like her just a little". Second, an agent's belief in the various degrees of John's liking of Mary may also vary. Thus, one may *strongly* believe that "John likes Mary a *little*[1]'. Representation of such distinctions may serve as a point of contact between a probabilistic or evidential approach and a formalism such as fuzzy logic.

Finally, figure 7.3 shows a simple example encoding the event described by "Jim made John hit Tom yesterday". As before, the figure is meant to convey a general idea of how we intend to approach these problems.

7.2 Treatment of evidential information

The representation language described in section 4.1 assumes that if an agent knows $\delta(C,P)$ then he knows $\#C[P,V]$ for all $V \in \Lambda(P)$. This eliminates the possibility of representing one form of ignorance. Assume that the agent knows that 40% apples are red, 30% are green and 20% are yellow, but he is unsure about the remaining 10%. In the evidential treatment outlined in chapter 5, there is no explicit way of taking such information into account (although such a situation was discussed in section 3.6, recall the use of ?COLOR node). One possibility is to posit a special value Θ_P for every property P. One may now include $\#C[P,\theta_P]$ in the specification of $\delta(C,P)$. This raises the question of how to distribute the "ignorance" represented by θ_P among the values of P. Should the count assigned to $\#C[P,\theta_P]$ be distributed equally among all $V \in \Lambda(P) - \theta_P$, or should it be distributed equally to all V such that $\#C[P,V] = 0$. The answer would depend on whether the count $\#C[P,\theta_P]$ denotes the observations made by the agent about miscellaneous values of P not represented in δ because they were insignificant, or whether the count denotes the agent's belief that there exist other instances of C which may have values other than those he has observed. In the former case the count $\#C[P,\theta_P]$ would be distributed to all V such that $\#C[P,V] = 0$, while in the latter case it would be distributed to all $\#C[P,V]$.

In addition to the kind of ignorance discussed above, there is another form of ignorance that ought be considered[1]. This form of ignorance may arise because the agent may not have observed sufficient instances of a class to be confident about its distributions recorded by him. In other words, #C may be a so small that the agent may not want to use #C[P,V]'s to infer property values of instances of C. It seems plausible to assume that in such situations an agent may prefer to employ distribution information of a more general class if such information is available and is thought to be more reliable. An explicit representation of ignorance may help in modelling this situation. For example one may posit that if the ignorance 9_{CP} associated with S(C,P) is very high, then #C[P,V] be estimated on the basis of S(C,P) as well as 5(D,P), where D is some concept higher up in the conceptual hierarchy for which G_{DP} is very low. It remains to be seen how such a strategy may be incorporated within the framework developed in this thesis.

1. The significance of this form of ignorance was pointed out by Gary Dell.

A related issue is that of treating likelihoods as intervals rather than point values. There are many arguments in favor of using intervals [Loui et al. 85]. One of the most forceful being that intervals enrich our capacity to express ignorance. If probabilites express uncertainty, then intervals allow us to express uncertainty about probabilities. For example, if one were certain of one's belief in the probability of some event as being p, then one would express this probability as [p,p]. However, if one's belief about this probability were itself uncertain, one might express this probability of the event as [p-e, p + e]- With reference to the example about apples and grapes in section 5.1 (figure 5.1), there could be at most 60 and at least 30 red and sweet apples. Thus, Pr(red & sweet | apples) lies in the interval [0.3, 0.6]. Using the maximum entropy formalism we reduced this to a single point value of 0.42 (42 out of 100 apples were red and sweet). Given the information about apples and grapes in that example, the value 0.42 is the most likely value from among the set of possible values in the interval [0.3, 0.6]. The rationale for this has been provided in section 5.1. However, the introduction of 0_p in the set of values of P introduces a complication that needs to be resolved.

Another suggestion due to Rollinger [Rollinger 83] involves using a dimensional representation for uncertainty. One of the dimension represents positive evidence while the other represents negative evidence. It is argued that a single value does not distinguish between "strong positive evidence in conjunction with some negative evidence" and "some positive evidence and no negative evidence". Thus, [1. 0] means the proposition is true, [0, 1] means that the proposition is false, while [1, 1] means that there is a contradiction.

The problem of evidential reasoning becomes extremely complex if the nature of information available to the agent includes inequality constraints. For example, an agent may know that there are more red and sweet apples than there are green and sour ones, without knowing how many such apples there are. In the presence of inequaltiy constraints, the maximum entropy computations become as complex as general optimization problems. Needless to say, developing computationally tractable solutions to such problems is beyond the scope of this thesis. For a promising approach for computing approximate solutions to optimization problems using massive parallelism refer to [Hopfield & Tank 85].

7.3 Extended inference

As stated in section 1.3, it was my contention that we must first identify the kinds of inference that an agent needs to perform very fast and provide a computational account of how this may be achieved. Consequently, this thesis focuses on a form of reasoning that is performed effortlessly and automatically by human agents. In contrast, a more conscious, directed, and sometimes even painstakingly slow and belabored form of reasoning underlies many cognitive activities. Examples abound; planning a trip, planning a talk, solving puzzles, troubleshooting a VLSI circuit, playing chess and the like. Some of these "higher forms" of reasoning are best modelled after standard deductive reasoning, some others by qualitative reasoning [Bobrow 85], and yet others by a hybrid approach. Considerable effort in AI has been directed towards embodying this form of reasoning in computer programs.

One may be lead to believe that it may not be possible to employ the proposed parallel architecture to handle more elaborate forms of reasoning. Although many problems remain unresolved, some progress has been made. For a description of how routines may be employed to perform more elaborate inferences refer to section 3.4 in [Shastri & Feldman 84].

With reference to evidential reasoning, the discussion in this thesis may lead one to conclude that it is being suggested that decisions should be based only on the likelihoods of possible outcomes, and the agent must always choose the most likely outcome. This is not the case. In most situations the agent may want to take into account the *utilities* of various outcomes, then again he may choose to take risks or be conservative, or he may adopt some other strategy. My contention is that any complicated or elaborate strategy (unless it simply involves making random choices) will perforce require the knowledge of the likelihoods of the possible outcomes.

7.4 Learning

There has been an undercurrent of issues related to learning throughout this thesis. The knowledge encoded in δ was supposed to be based on the observations made by the agent, it was mentioned time and again that Types "evolve" when certain things happen, hence, it is only natural to discuss some issues related to learning. If one examines the encoding of knowledge as described in sections 6.1 and 6.2, one would notice, that most of the weights drawn between links have a very simple explanation. If we view nodes in the network to be active elements - as indeed they are, then the weights on the links emanating from δ nodes (both δ_{inh} and δ_{cat}) and incident on other ξ-nodes and δ-nodes have the following interpretation:

"the weight on a link is a measure of *how often when the destination node was active, was the source node also active.*"

This interpretation relates extremely well to a Hebbian interpretation of synaptic weights in neural nets. These weights are based on purely local information. Thus, the weight on a link from a δ_{inh} node [C,P->V] to V is precisely the fraction: "how often when V was firing was [C,P->V] also firing". If V is RED, P is has-color and C is APPLE, then the weight on the link from APPLE to RED via the binder node would equal:

"what fraction of red colored things were apples"

The above explanation may sound plausible for computing the weights on individual links, but it does not answer how structures such as concepts evolve. The following is a preliminary attempt at answering this question. Although the problem is far from solved, it does provide a general idea of how learning may occur in connectionist semantic networks. The emphasis is on identifying how pre-existing (innate) structure may give rise to new concepts.

The proposed mechanism for learning in semantic networks is based on the notions of recruitment and chunking [Feldman 82a; Wickelgren 79] and these issues are discussed in brief before a plausible mechanism of concept formation is outlined. Broadly speaking, the idea of chunking may be described as follows: At a given time, the network consists of two classes of nodes:

1. ***Committed*** Nodes. These are nodes that have acquired a distinct "meaning" in the network. By this we mean that given any committed node, one can clearly identify sets of other committed nodes, whose activation will result in the former becoming activated. Committed nodes are connected to other committed nodes by "strong" links, and to a host of other *free* nodes, (see below), via "weak" links.

2. ***Free*** Nodes. These are nodes that have a multiplicity of weak links to other nodes, both free and committed. These form a kind of "primordial network" of uncommitted nodes within which the network of committed nodes is embedded.

Chunking involves strengthening the links between a cluster of committed nodes and a free node. Thereafter, the free node becomes committed and functions as the chunking node for the cluster i.e., the activation of nodes in the cluster results in the activation of the chunking node and conversely, the activation of the chunking node activates all the nodes in the cluster. The process by which a free node is transformed to a committed node is called *recruitment*. The mechanics of recruitment in connectionist networks is described in detail in [Feldman 82a]. The basic insight in the solution to the problem of learning through weight change is that certain classes of random connection graphs have a very high probability of containing the sub-network needed for learning a new concept.

The notion of chunking in its generic form only suggests a mechanism whereby nodes can be associated and is not sufficient for explaining how structured relationships arise. In the proposed solution we wish to exploit the non-trivial structure resulting from assuming that knowledge is organized in terms of properties and values thereof. We postulate that learning takes place within a network that is already organized to reflect this structure. For instance, in the context of vision, we specifically assume that concepts that correspond to primitive properties like color, shape, texture and motion are already present in the semantic network of an agent together with concepts that represent some basic values of these properties. Simple forms of learning result in the formation of concepts that represent coherent collections of existing properties and values, while more complex forms of learning lead to generalization of concepts and the formation of complex properties that in turn lead to development of more complex concepts.

We will consider a toy example of a semantic network interacting with a very simple visual system that is capable of detecting the colors blue and green and the primitive shapes round and oval. The initial organization of the semantic network takes into account these characterstics of the visual system. Figure 7.4a is an oversimplified representation of the initial organization of the semantic network. The network has four pre-existing concepts namely, the property *HAS-COLOR* and its values BLUE and GREEN and the property *HAS-SHAPE* and its values ROUND and OVAL. In other words, the nodes representing the properties and values are already connected to the visual system and may be activated by it under appropriate conditions. The nodes representing the four concepts are committed nodes embedded in a "primordial network" of free nodes that may be roughly partitioned into three diffused sub-networks X, Y and Z. Network X consists of nodes that are primarily connected to the nodes *HAS-COLOR*, BLUE and GREEN along with a host of free nodes in network Z. Nodes in network Y receive most of their connections from the nodes *HAS-SHAPE*, ROUND and OVAL and also from numerous free nodes in network Z. Finally, the nodes in network Z are connected to a large number of nodes throughout the semantic network. The existence of networks X and Y indicates that the semantic network is pre-wired to "know" that BLUE and GREEN are values of *HAS-COLOR* while ROUND and OVAL are the values of *HAS-SHAPE*.

Figure 7.4b depicts the result of learning an instance of a blue and round object. The figure only shows the committed units and their interconnections. Learning an instance involves two stages of recruitment; the binder nodes Bl and Rl are recruited first, followed by the concept node BRL When the visual system detects the color blue in the stimulus it activates the node *HAS-COLOR* and BLUE. The coincident activation results in the recruitment of a free node (Bl) from the pool of free nodes in network X. The node Rl is recruited in an analogous manner from the pool of nodes in network Y. The simultaneous activity in Bl and Rl leads to the recruitment of the node

(BR1) from network Z. Thereafter, the nodes B1 and R1 act as binder nodes and BR1 represents the newly acquired concept. B1 is activated by the coincident activity of *HAS-COLOR* and BLUE while R1 is activated by the coincident activity of *HAS-SHAPE* and ROUND. The activity of the concept node BR1 is strongly correlated with the activity of B1 and R1.

The working of the scheme depends on the assumptions we made about the pre-existing structure of the semantic network. It was crucial to assume the existence of property and value nodes with appropriate connections to the visual system. The organization of free nodes as networks X, Y and Z was equally important. Networks X and Y provided binder nodes in order to associate properties with their values, and the network Z provided a pool of nodes that could be recruited to "chunk" binder nodes in order to form concepts.

Figure 7.4c depicts the semantic network with three instances (BR1, BR2 and BR3) of blue round objects and one instance (GO1) of a green oval object. In this situation a second kind of concept formation may occur and result in the formation of the concept "blue and round object" which is a generalization defined over BR1, BR2 and BR3. The resulting network is shown in Figure 7.4d. The new concept is represented by the node BR that owns the binders B and R that indicate its property values. These property values correspond to the shared property values of the instances.

The transformation from the network in Figure 7.4c to that in Figure 7.4d is best explained with the help of the simpler networks shown in Figure 7.5. The network shown in Figure 7.5b is the result of a similar transformation of the network in Figure 7.5a. The three instances A, B and C have the same value (V) for the property P and this forms the basis for the formation of the more general concept D. The transformation occurs in two stages.

I. A chunking node for b1, b2 and b3 is recruited from a pool of free nodes that serves the same function as network Z in the previous example, i.e. provides a potential concept node.

II i) Over a longer period of time, the multiple paths between P and V via b1, b2 and b3 collapse into a single path via b, where b is one of the existing binder nodes b1, b2 or b3. The collapsing of links does not mean that the links disappear, but rather that the weights of links get reduced in such a way that all binder nodes besides b gradually become free nodes (are released).

ii) The connection between b and D remain strong but the connections between other binder nodes and D become weak.

iii) The links x,y and z (in effect) now emanate from D rather than the binder nodes.

(All changes described in stage II happen during the same time interval).

The net effect of I and II is that the network shown in Figure 7.5a behaves like the network shown in Figure 7.5b. The scheme that we have just described characterizes learning as network transformations that minimize the complexity of the network (number of links and nodes) while maintaining the cause effect relationships between existing concept nodes. Thus, the nodes P, V, A, B and C have roughly the same effect on each other in the two networks shown in Figures 7.5a and 7.5b. The complexity of networks is substantially reduced by formation of more general concepts although this may not be evident from this simple example. In general, if the generalization takes place over p properties and c instances (the values of p and c were 1 and 3 in the example of Figure 7.5 and 2 and 3 in the example of Figure 7.4d), then the savings in the number of links and nodes is of the order of pxc.

Referring back to Figure 7.4d, BR, a node in network Z, will be recruited as a chunking node of Bl, B2, B3 as well as Rl, R2 and R3. The release of binder nodes and the collapsing of links will occur separately for the two properties *HAS-COLOR* and *HAS-SHAPE*. Thus, B1, B2 and B3 will collapse into B while R1, R2 and R3 will collapse into R.

The above is intended to be a crude description of how recruitment of free nodes and release of committed nodes gives rise to representation of new instances and development of concepts that are generalizations of existing concepts. The latter kind of concept formation is accompanied by a substantial reduction in the number of committed nodes and links.

7.5. Conclusion

This thesis demonstrates that certain problems in knowledge representation and reasoning have elegant solutions within an evidential framework. I hope that this work provides a point of contact between researchers who adopt traditional approaches - (i.e. various non-monotonic logics), and researchers who adopt an evidential approach to deal with partial and uncertain knowledge. I also hope that this will lead to a greater interaction between the two groups that have largely worked independently.

This thesis also demonstrates the efficacy of developing efficient connectionist solutions to problems that are considered to be "too hard" for an apparently "low-level" approach. My experience has been otherwise - "thinking" in connectionist terms gave rise to the intuitions that lead to this thesis. I believe that a deep understanding of what is intelligence, why is it that we view the world to be structured as we do, and why are we good at certain tasks while bad at some others, will only accrue if we seriously examine the information processing characterstics and the computational properties of the biological machine (i.e. the human brain) that embodies intelligence.

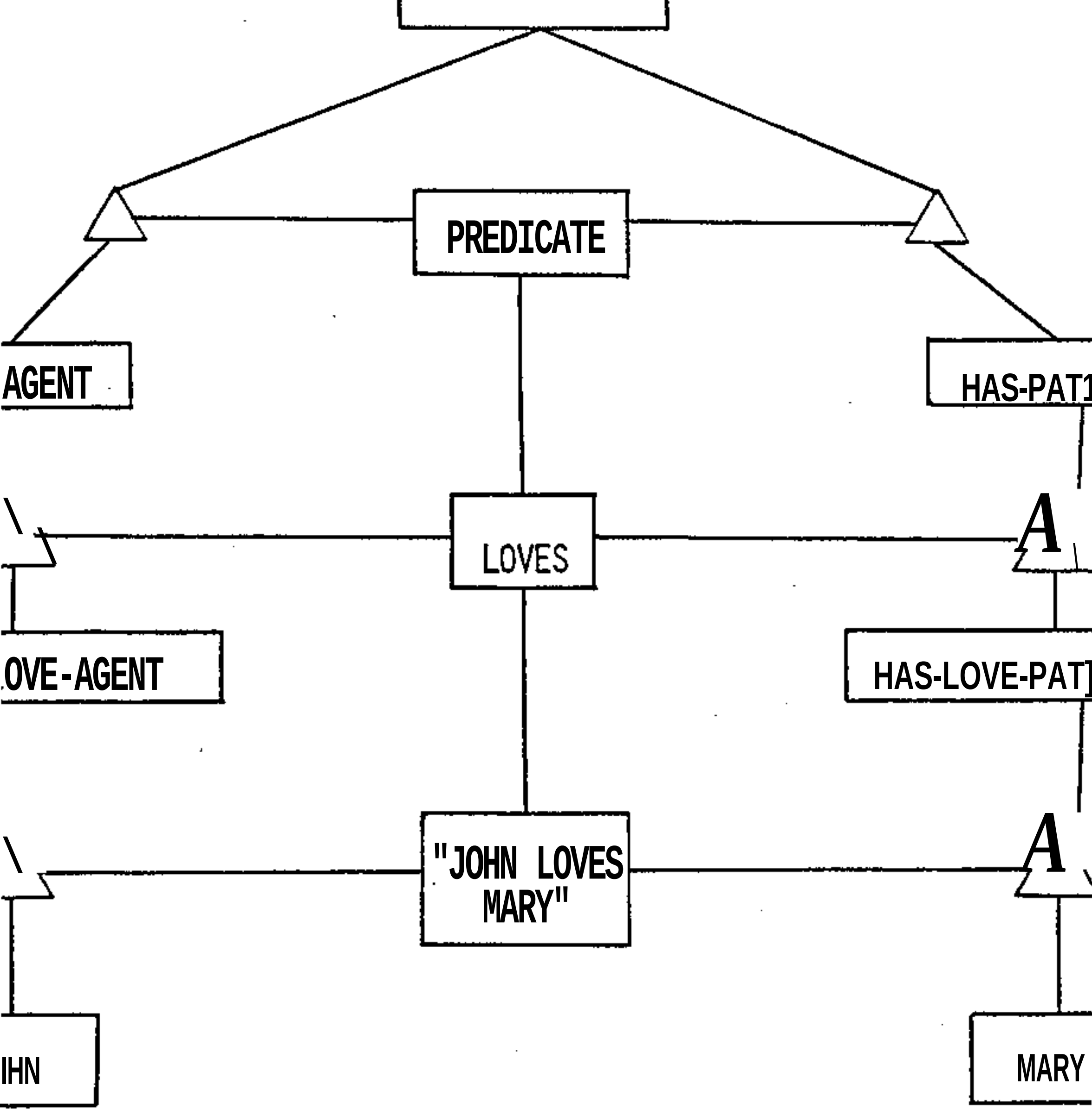


FIGURE 7.1 "John loves Mary"

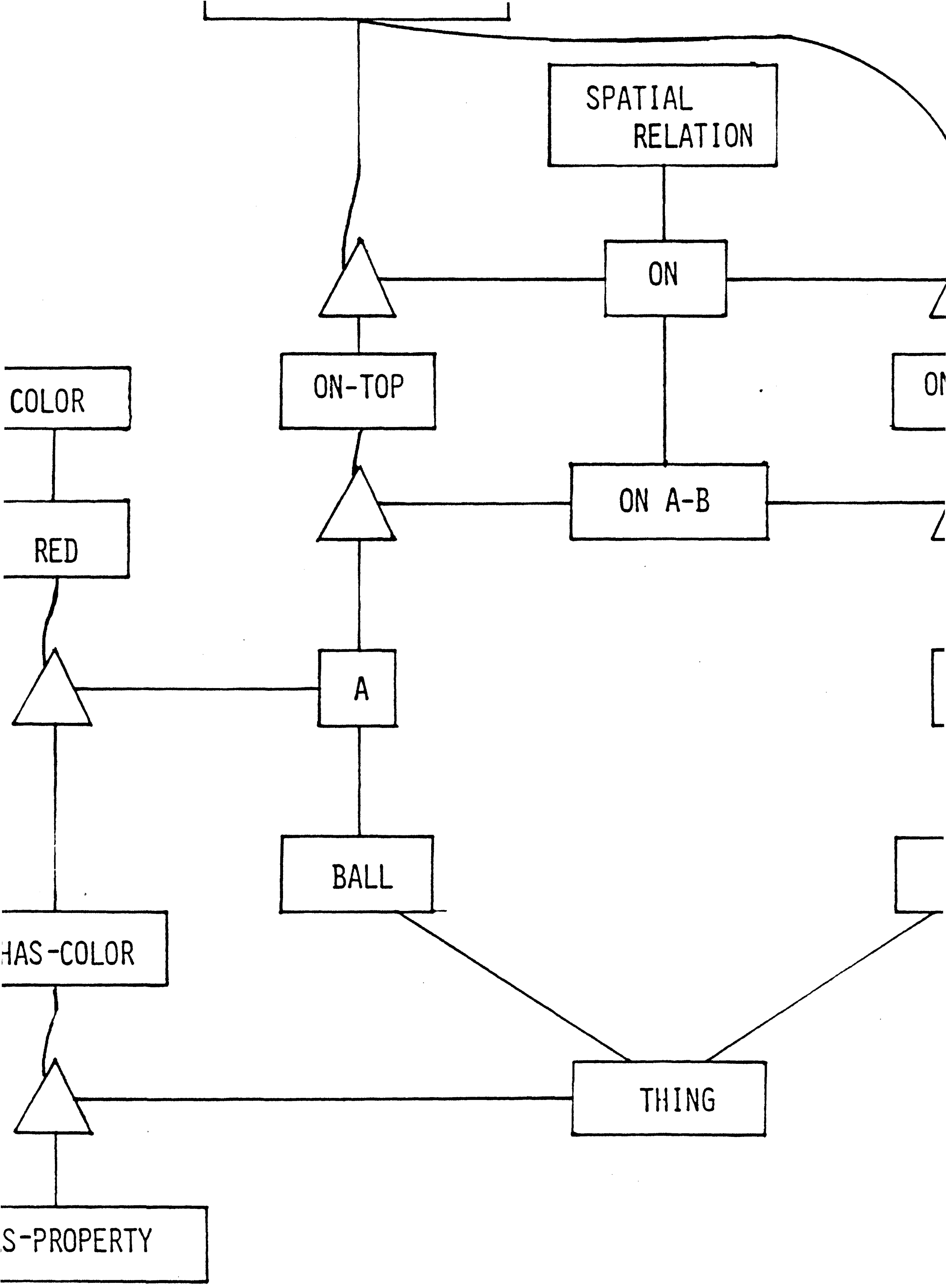


FIGURE 7.2 "A is on B"

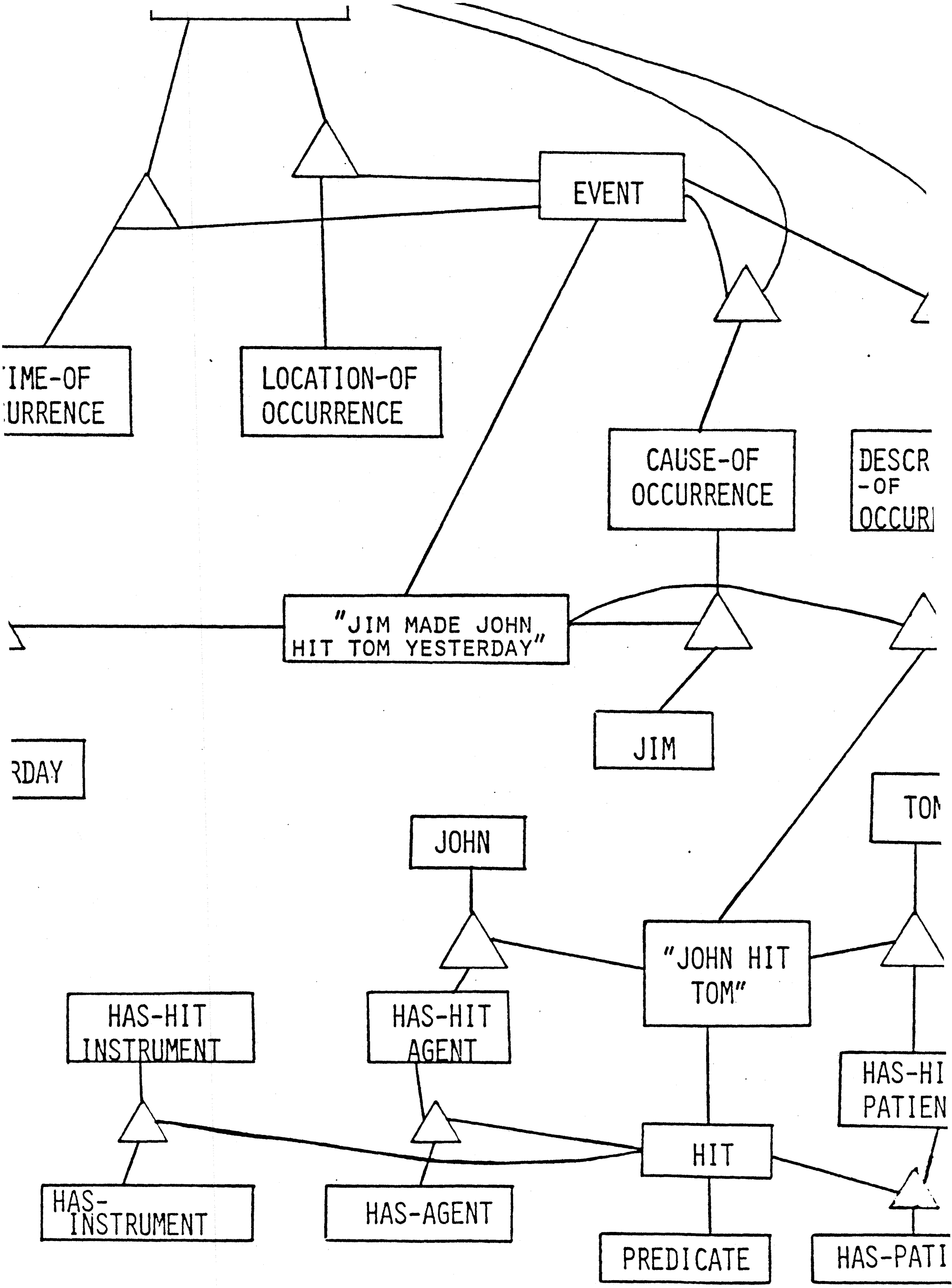
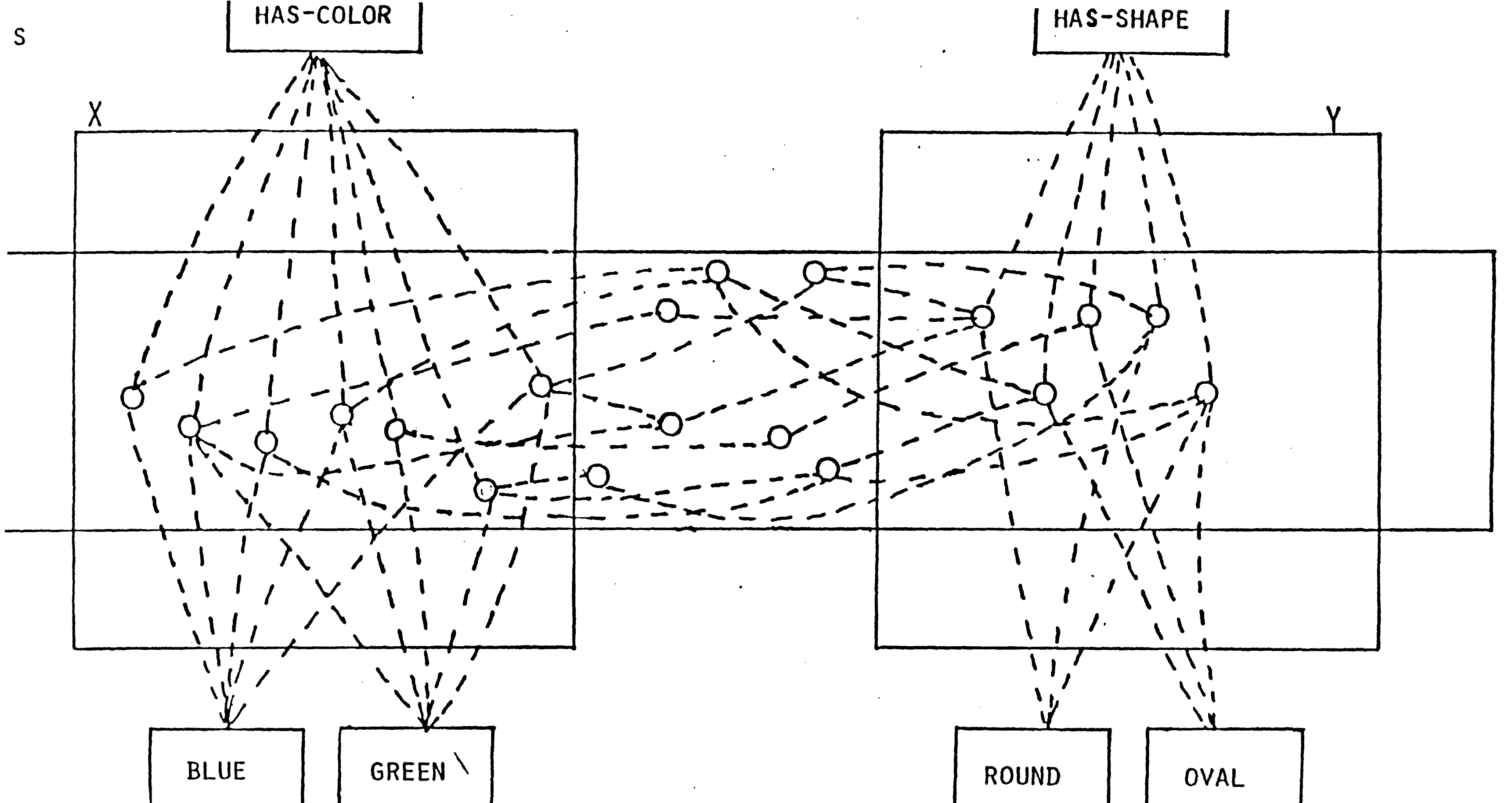
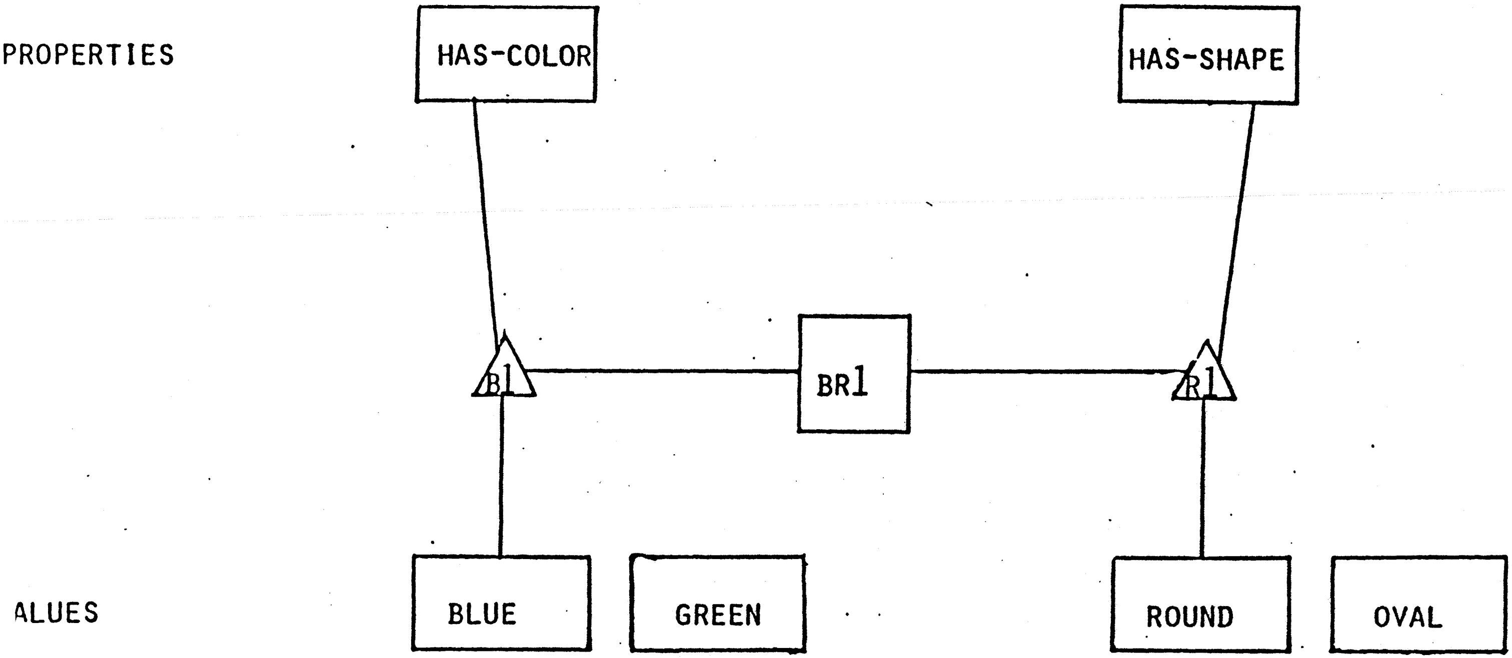


FIGURE 7.3 "Jim made John hit Tom yesterday"



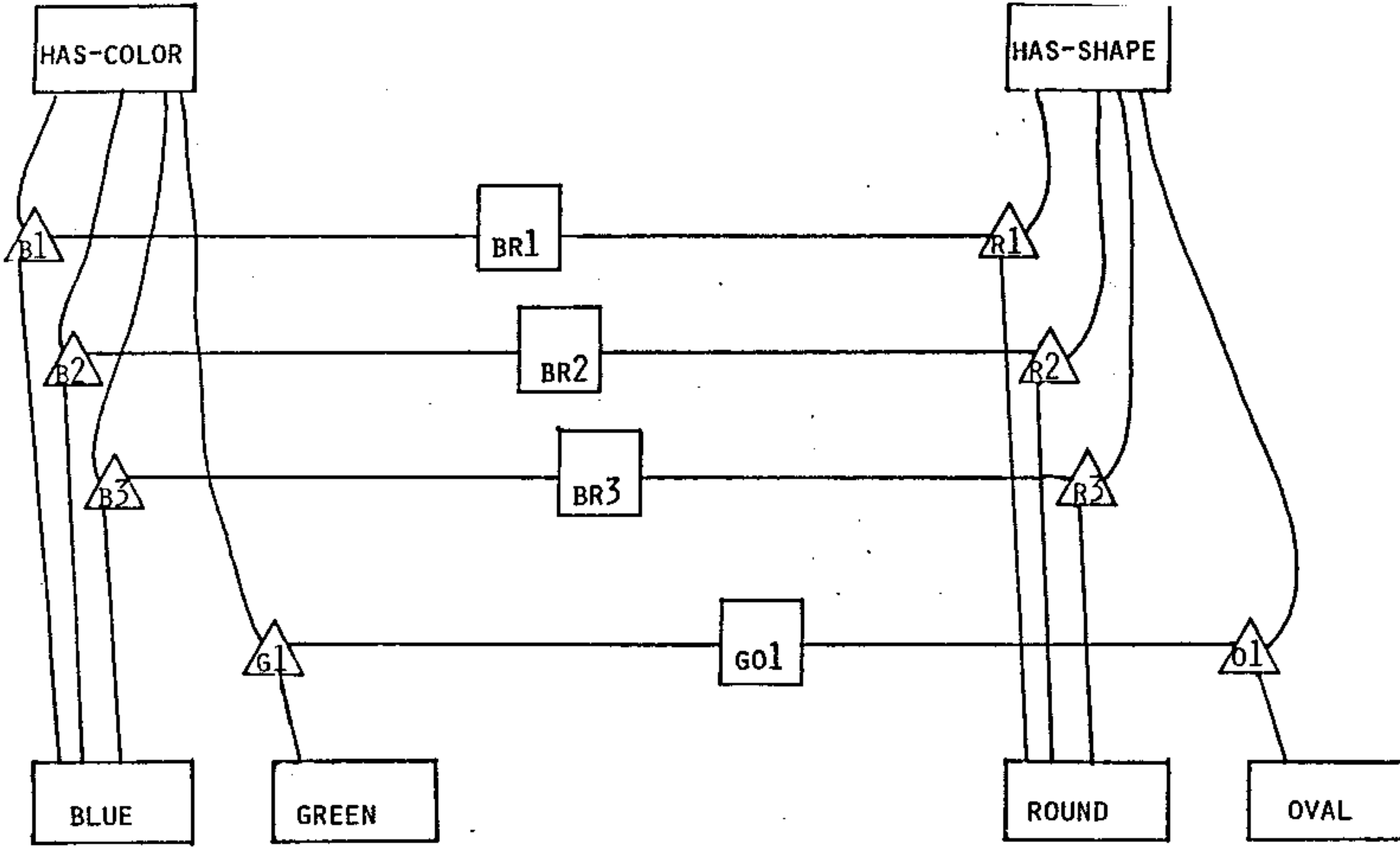
- WEAK LINKS

FREE NODES



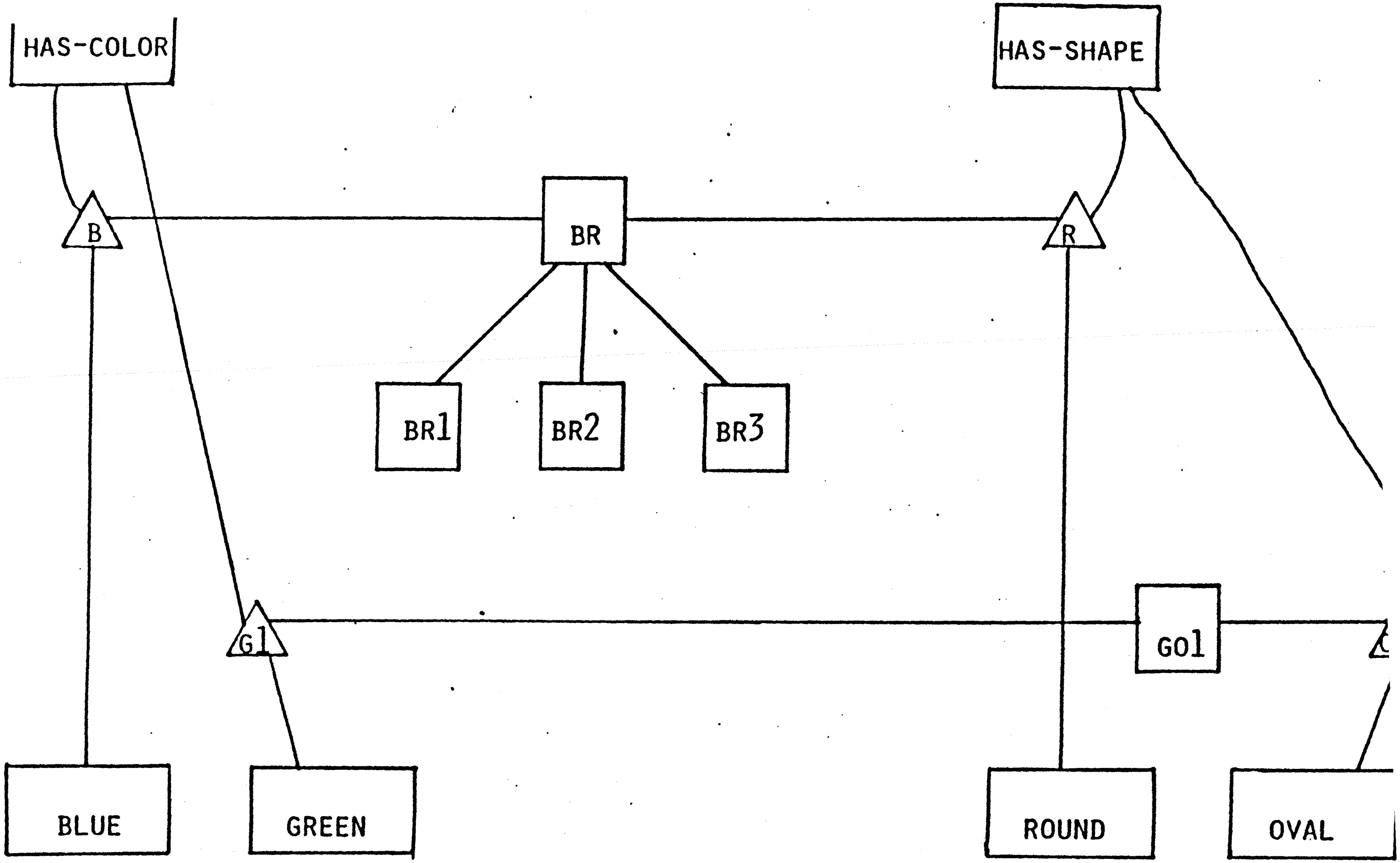
REE NODES AND WEAK LINKS ARE NOT SHOWN.

FIGURE 7. 4b A blue and round object represented in the semantic network



Free nodes and weak links are not shown

FIGURE 7.4c Multiple blue and round objects



Free nodes and weak links are not shown

FIGURE 7. 4d A new type represents blue and round objects

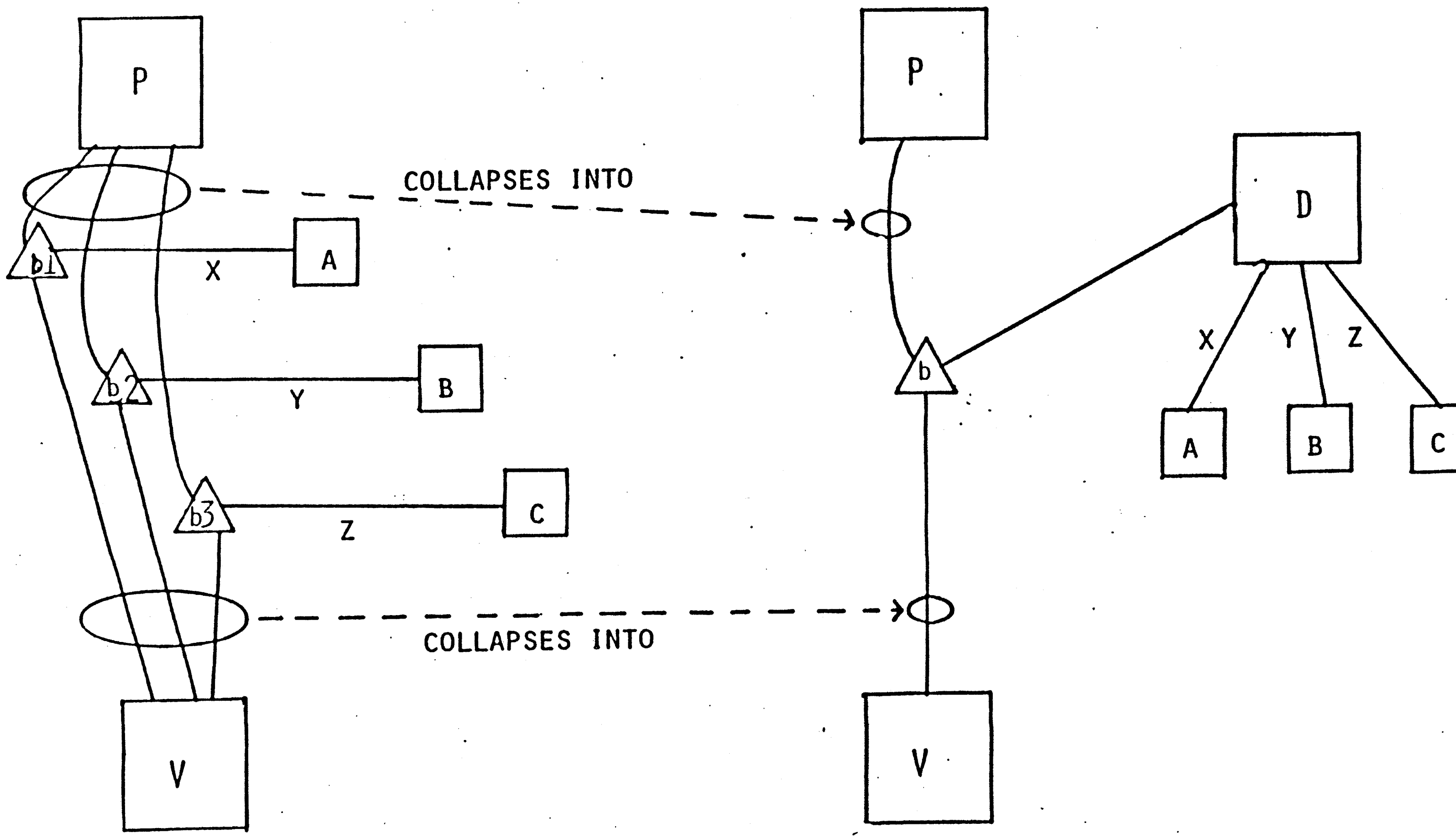


Figure : 7. 5a

Figure : 7. 5b

FIGURE 7.5 Network in figure 7.5a collapses into network in figure 7.5b

BIBLIOGRAPHY

Even though some publications listed below have not been referenced, they have influenced the work described in this thesis.

[Ackley & Berliner 83] Ackley D.H. and Hans J. Berliner. The QBKG System: Knowledge Representation for Producing and Explaining Judgenments. Technical Report Carnegie-Mellon University, March 1983.

[Allen 83] Allen JLF. Maintaining knowledge about Temporal Intervals. *Commun.* ACM 26, 1983, 832-843.

[Allen & Frisch 82] Allen J.F. and A.M. Frisch. What's in a Semantic network? Proc. 20th Annual Meeting, Assoc, of Computational Linguistics, Univ. of Toronto, 1982.

[Addanki 84] Addanki Sanjay. Applications of Connectionist Modeling for motor control systems, Ph.D. Dissertation. Dept. of Computer Science. University of Rochester, 1984.

[Anderson 83] Anderson John R. *The Architecture of Cognition.* Harvard University Press. Cambridge Massachusetts. 1983.

[Ballard 84] Parameter Networks, *Artificial Intelligence,* 22, 1984, 235-267

[Ballard & Brown 82] Ballard Dana H., Christopher M. Brown, Computer ' Vision Prentice Hall, Englewood Cliffs, New Jersey. 1982.

[Barnett 81] Barnett, Jeffery A. Computational methods for a mathematical theory of evidence. In *Proc. UCAI-81,* Vancouver, B.C., 1981.

[Bobrow 85] Bobrow D.G. (Ed.) *Qualitative Reasoning about Physical Systems.* A Bradford Book, The MIT Press. Cambridge, 1985.

[Bobrow & Winograd 76] Bobrow D.G., Winograd Terry. An overview of KRL: A Knowledge Representation Language. CSL-76-4. Xerox Palo Alto Research Centre.

[Brachman 82] Brachman RJ. What "ISA" IS and Is'nt In the proceedings of **the *Fourth National Conference of the Canadian Society for the Computational Studies of Intelligence,* Saskatoon, Canada. 1982. pp 212-**221.

[Brachman 79] Brachman RJ. On the epistemological status of semantic **networks. In *Associative Networks: Representation and use of Knowledge*** *by Computers*. N.V. Findler (ed.), Academic Press 1979.

[Brachman 77] Brachman RJ. A Structural Paradigm for Representing Knowledge, Ph.D. Thesis. Division of Engineering and Applied Physics, Harvard University, 1977.

[Brachman et al. 83] Brachman RJ., R.E. Fikes, and HJ. Levesque. KRYPTON: A functional approach to Knowledge Representation. IEEE Computer - 16(10), October 1983.

[Bruce 75] Bruce B.C. Case systems for natural language. *Artificial Intelligence,* 1975, 6, 327-360.

[Cercone & Schubert 75] Cercone N, Schubert Len. Toward a State Based Conceptual Representation. *In Proc. IJCAI-75,*. pp 83-90.

[Charniak 83] Charniak E. The Bayesian Basis of Common Sense Medical Diagnosis, In proc. AAAI-83, pp 70-73. 1983.

[Charniak 81] Charniak E. A common representation for problem solving and language comprehension information. *Artificial Intelligence,* 16 225-255, 1981.

[Cheeseman 83] Cheeseman, Peter. A Method of Computing Generalized Bayesian Probability Values For Expert Systems. In *Proc. IJCAI-83,* Karlsruhe, West Germany, 1983.

[Cognitive Science 85] Cognitive Science, Vol. 9, No. 1, Jan.-March 1985.

[Collins & Loftus 75] Collins Allan M. Loftus Elizabeth F. A Spreading

Activation Theory of Semantic Processing. In *Psychology Review*, vol. 82, No. 6, pp 409-428, 1975.

[Cottrell 85] Cottrell Garrison W. A connectionist approach to word-sens disambiguation. Ph.D. Dissertation. Dept. of Computer Science. University of Rochester, 1985.

[Dell 80] Dell Gary S. Phonological and Lexical Encoding in Speech Production. Ph. D. dissertation, Dept. of Psychology, University of Toronoto, 1980.

[Dosher 83] Dosher, Barbara Anne. Effect of sentence size and Network Distance on Retrieval Speed. In Journal of Experimental Psychology: Learning, Memory, and Cognition. vol. 8, No. 3, 173-207. 1983.

[Doyle 83] Doyle Jon, Some theories of reasoned assumptions: an essay in rational psychology. CS-83-125, Technical Report Carnegie-Mellon University, Pittsburgh PA.

[Duda et al. 76] Duda, R., P.E. Hart, and N.J. Nilsson. Subjective Bayesian Methods for Rule-Based Inference Systems. Technical Note 124, SRI International, 1976.

[Etherington & Reiter 83] Etherington, David W. and R. Reiter. On Inheritance Hierarchies with Exceptions. In *Proc. AAAI-83*, Washington D.C. 1983.

[Fahlman 82] Fahlman Scott E. Three flavors of Parallelism. In *The proceedings of the 4th National Conference of the Canadian Society for Computer Studies of Intelligence*, 1982.

[Fahlman, Touretzky & van Roggen 81] Fahlman Scott E., Touretzky David S., van Roggen Walter. Cancellation in a Parallel Semantic Network. In *Proc. IJCAI-81* Vancouver. B.C. 1981.

[Fahlman 79] Fahlman Scott E. *NETL: A System for Representing and Using Real-World Knowledge.* The MIT Press, 1979.

[Fanty 85a] Fanty M.A. Connectionist network building tools. Internal document, Computer Science Department, University of Rochester. June 1985.

[Fanty 85b] Fanty M.A. Connectionist network simulation tools. Internal document, Computer Science Department, University of Rochester. June 1985.

[Feldman & Shastri 84] Evidential inference in activation networks. In *Proc. of the Cognitive Science Society Conference*, Boulder, Co. June 1984.

[Feldman 82a] Feldman Jerome A. Dynamic connections in neural networks, Bio-Cybernetics, 46, 27-39, 1982.

[Feldman 82b] Feldman Jerome A. Four Frames Suffice: A provisionary model of vision and space. TR99. Dept. of Computer Science, University of Rochester, 1982.

[Feldman & Ballard 82] Feldman Jerome A., Ballard Dana H. Connectionist Models and their Properties. *Cognitive Science*, 6, pp 205-254, 1982.

[Fillmore 68] Fillmore C.J. The Case for Case. In *Universals in Linguistic Theory*, E.W. Bach and R.T. Harms (Eds.), pp 1-88. Holt, Rinehart & Winston, New York. 1968.

[Findler 79] Findler Nicholas V. (Ed.) *Associative networks: Representation and use of knowledge by computers.* Academic Press. New York San Francisco London. 1979.

[Fox 79] Fox Mark S. On Inheritance in Knowledge Representation. In *Proc. IJCAI-75*, 1975.

[Fox 81] Fox, Mark S. Reasoning with incomplete knowledge in a resource-limited environment: Integrating reasoning and knowledge acquisition. In *Proc. IJCAI-81*, Vancouver B.C. 1981.

[Frisch & Allen 82] Frisch, A.M. and J.F. Allen, Knowledge retrieval as limited inference in D.W. Loveland (Ed.) *Lecture notes in Computer Science: 6th Conference on Automated Deduction.* Springer-Verlag, New York. 1982.

[Garvey et al. 81] Garvey, Thomas D., John D. Lowrance, and Martin A. Fischler. An inference technique for integrating knowledge from disparate sources. In *Proc. IJCAI-81,* Vancouver B.C. 1981.

[Geman & Geman 83] Geman Stuart, Geman Donald. Stochastic Relaxations, Gibbs Distribution and the Baysian Restoration of Images. Unpublished manuscript September 1983.

[Ginsberg 84] Ginsberg, Mattew L. Non-Monotonic Reasoning using Dempster's Rule. In *Proc. AAAI-84*, Austin Texas, 1984.

[Gluck & Corter 85] Gluck M.A. and J.E. Corter. Information and Category Utility. In preparation. Department of Psychology, Stanford University.

[Halpern & McAllester 84] Halpern, Joseph Y. and David A. McAllester. Likelihood, Probability and Knowledge. In *Proc. AAAI-84*, Austin Texas, 1984.

[Hart & Duda 77] Hart, P.E. and R.O. Duda. PROSPECTOR - A computer-based consultant system for meineral exploration. TN 155, SRI International, Menlo Park, California. 1977.

[Hayes 77] Hayes Patrick J. The logic of frames. In D. Metzing (Ed). *Frame conception and Text Understanding.* Walter de Gruyter & Co. 1979.

[Hayes 77] Hayes Patrick J. In defense of Logic. In *Proc. IJCAI-77*, Cambridge Massachusetts 1977.

[Hayes 79] Hayes Patrick J. The Naive Physics Manifesto. In Donald Michie (Ed), *Expert Systems in the Micro-Electronic Age.* Edinburgh University Press, 1979.

[Hillis 81] Hillis W.D. The Connection Machine (Computer architecture for the New Wave). Memo #646, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1981.

[Hinton & Anderson 81] Hinton G.E., Anderson J.A. (Eds) *Parallel Models of AssociativeMemory.* Lawrence Erlbaum Associates, 1981.

[Hinton 79] Hinton G.E. Relaxation and its Role in Vision. Ph. D. Thesis, Cognitive Studies Program. University of Edinburgh, 1979.

[Hinton & Sejnowski 83] Hinton G.E., Sejnowski T. Optimal perceptual **inference. In *Proceeding of the IEEE conference on Computer Vision and*** *Pattern Recognition,* pp 448-453. IEEE Computer Science Press, Siver Spring Maryland. 1983

[Hinton & Sejnowski 83] Hinton G.E., Sejnowski T. Analyzing cooperative **computation. In *Proceeding of the Fifth Annual Conference of the*** *Cognitive Science Society.* Rochester, New York. May 1983.

[Hop field 82] Hop field J.J. Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the National* ***Academy of Sciences USA,* 1982, 79 pp 2554-2558.**

[Hopfield & Tank 85] Hopfield J.J. and D.W. Tank. 'Neural' computation of decisions in optimization problems. To appear in Biological Cybernetics, 1985.

[Jackendoff 83] Jackendoff Ray. *Semantics and Cognition.* The MIT Press, Cambridge Massachusetts. 1983.

[Jaynes 57] Jaynes, E.T. Information Theory and Statistical Mechanics. Part I, *Phy. Rev.,* vol. 106, pp. 620-630, March 1957; Part II, ibid., vol. 108, pp 171-191, October 1957.

[Jaynes 79] Jaynes E.T. Where Do We Stand on Maximum Entropy. In *The maximum entropy formalism,* R.D. Levine and M. Tribus (Eds.) MIT Press, Cambridge Massachussets. 1979.

[Joshi 78] Joshi, Aravind K. Some extensions of a system for inference on partial information. In *Pattern-Directed Inference Systems,* D.A. Waterman and Fredrick Hayes-Roth (Eds.), Academic Press, 1978.

[Keil 79] Keil, F.C. *Semantic and conceptual development.* Cambridge, Mass. Harvard University Press. 1979.

[Kyburg 83] Kyburg, Henry E. Jr. The reference class. *Philosophy of Science,* 50 (1983) pp 374-397.

[Levesque 82] Levesque, Hector J. A formal treatment of incomplete knowledge bases. Ph. D. Thesis. CSRG-139. Computer Systems Research Group. University of Toronto. Toronto, Canada. February 1982.

[Levesque 84] Levesque, Hector J. A Fundamental Tradeoff in Knowledge Representation and Reasoning. In Proc. *CS-CSI-84,* London, Ontario, Canada. 1984.

[Loui et al. 85] Interval-based probabilistic reasoning. Submitted to the Workshop on probability in artificial intelligence, Los Angeles 1985.

[Lowerre & Reddy 79] Lowerre B.T., R. Reddy. The HARPY speech understanding system. In W.A. Lea (Ed.) *Trends in Speech Recognition.* Prentice Hall, Englewood California. 1979. Chapter-15.

[Lowrance 82] Lowrance John D. Dependency-Graph Models of Evidential Support, Ph. D. dissertation, Department of Computer and Information Science, University of Massachusetts, Amherst, Mass. (September 1982).

[Marr & Nishihara 78] Marr D., Nishihara H.K., Representation and recognition of the spatial organization of three-dimensional shapes. Proceedings of the Royal Society of London B 200, 1978, 269-294.

[McCarthy 84] McCarthy, John. Applications of Circumscription to Formalizing Common Sense Knowledge. In Proc. *AAAI Workshop on non-monotonic reasoning.* New Paltz, NY, 1984.

[McCarthy & Hayes 69] McCarthy, J, Hayes, P. J. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In Bernard Meltzer & Donald Michie (Eds.), *Machine Intelligence 4.* American Elsevier Publishing Company Inc., 1969.

[McClelland & Rumelhart 81] McClelland J.L, Rumelhart D.E. An Interactive Activtion Model of context effects in letter perception. Part I, An account of basic findings. *Psych. Review*, 1981, 88, 375-407.

[McDermott & Doyle 80] McDermott, D. and J. Doyle. Non-Monotonic Logic I. *Artificial Intelligence,* 13 pp 41-72, 1980.

[McDonald 83] McDonald David D. Natural language generation as a computational problem: an introduction. In M. Brady & R.C. Colby (Eds.) *Computational Models of Discourse.* Chapter-4. The MIT Press, Cambridge Massachusetts, London England. 1983

[Minsky 75] Minsky Marvin. A Framework for Representing Knowledge. In P.H. Winston *(Ed.), The Psycology of Vision.* McGraw Hill, 1975.

[Minsky & Papert 72] Minsky Marvin, Papert Samuel. *Perceptrons.* MIT Press, 1972.

[Moore 83] Moore Robert C. Semantic Considerations on Non-Monotonic Logic. In *Proc. IJCAI-83,* Karlsruhe, West Germany, 1983.

[Nilsson 84] Nilsson, Nils J. Probabilistic Logic. Technical Note 321. Artificial Intelligence Centre, Computer Science and Technology Division. SRI International. February 1984.

[Pearl 82] Pearl. Judea, Reverend Bayes on inference engines: a distributed hierarchical approach. In *Proc. AAAI-82,* pp 133-136. (1982).

[Pople 77] Pople, H.E. Jr. The formation of composite hypothesis in diagnostic problem solving: an exercise in synthetic reasoning. In *Proc. IJCAI-77,* Pittsburgh Pa. 1977.

[Quillian 68] Quillian Ross M. . Semantic Memory. In M. Minsky (Ed.), *Semantic Information Processing.* MIT Press, 1968.

[Quinlan 83] Quinlan J.R. Consistency and plausible reasoning. In *Proc. IJCAI-83,* Karlsruhe, West Germany. 1983. pp 137-144.

[Reiter 80] Reiter, Raymond. A Logic for Default reasoning. *Artificial Intelligence,* 13 pp 81-132.

[Reiter & Criscuolo 81] Reiter, R. and G. Criscuolo. On interacting defaults. In *Proc. IJCAI-81,* Vancouver B.C. 1981.

[Rich 84] Rich, Elaine. Default Reasoning as Likelihood reasoning. In *Proc. AAAI-84* Washington D.C., 1984 pp 348-351.

[Roberts & Goldstein 77] Roberts Bruce, Goldstein Ira. The FRL Manual MIT AI memo 409, 1977.

[Rollinger 83] Rollinger Claus-Rainer. How to represent Evidence - Aspects of Uncertain Reasoning. In *Proc. IJCAI-83.* Karlsruhe, W. Germany, 1983, pp 358-361.

[Rosch 75] Rosch E. Cognitive Representations of Semantic Categories. In *Journal of Experimental Psychology*: General 104, 192-233, 1975.

[Rumelhart & McClelland 82] Rumelhart D.E, McClelland J.L. An Interactive Activtion Model of context effects in letter perception. Part 2, The contextual enhancement effect and some tests and extension of the model. *Psych. Review,* 1982, 89, 60-94.

[Sabbah 85] Sabbah D. "Computing with connections in visual recognition of Origami objects, *Cognitive Science,* Vol. 9, No. 1, Jan.-March, 1985.

[Shafer 76] Shafer G. *A mathematical theory of evidence.* Princeton, N.J.: Princeton University Press, 1976.

[Schank 73] Schank Roger C. Identification of conceptualization underlying natural language. In R.C. Schank and K. Colby (Eds.). *Computer Models* *of Thought and Language.* Freeman, San Francisco 1973,

[Schank 82] Schank Roger C. *Dynamic Memory: a theory of reminding and* *learning in computers and people.* New York Cambridge University Press. 1982.

[Schubert 75] Schubert LK. Extending a Expressive Power of Semantic Networks. In *Proc. IJCAI-75,* 1975.

[Schubert et al. 83] Schubert L.K., Papalaskaris, MA, and Taugher, J. Determining Type, Part, Color, and Time Relationships. IEEE Computer 16(10): 55-60, October 1983.

[Shastri & Feldman 84] Shastri, Lokendra and J.A. Feldman. Semantic Networks and Neural Nets. *TR 131.* Computer Science Department, University of Rochester, January 1984.

[Shortliffe 76] Shortliffe, E.H. *Computer-based Medical Consultation:* *MYCIN.* American Elsevier Inc., New York, 1976.

[Simmons & Slochum 72] Simmons R.F, Slochum, J. Generating English discourse from Semantic Networks, Comm. of the ACM 15, 10 891-905. 1972.

[Small, Shastri, Brucks, Kaufman , Cottrell, & Addanki 82] Small S.L. Shastri L, Brucks M.L., Kaufman S.G., Cottrell G.W., Addanki S. ISCON: An interactive Simulator for Connectionist Networks. TR 109. Dept of Computer Science, University of Rochester, December 1982.

[Smith & Medin 81] Smith Edward E., Medin Douglas L. *Categories and Concepts.* Harvard University Press, Cambridge Massuchusetts. 1981.

[Sommers 65] Sommers, F. *Predicability.* In Philosophy in America (ed.) M. Black. Ithaca, N.Y.: Cornell University Press.

[Touretzky 84] Touretzky, David S. The Mathematics of Inheritance Systems. Ph. D. Thesis, Carnegie Mellon University, CMU-CS-84-136.

[Walker 78] Walker D.E. (Ed.) *Understanding spoken language.* American Elsevier. New York. 1978.

[Wickelgren 79] Wickelgren Wayne A. Chunking and Consolidation: A Theoretical Synthesis of Semantic Networks, Configuring in Conditioning, S-R Versus Cognitive Learning, Normal Forgetting, the Amnesic Syndrome, and the Hippocampal Arousal System. In *Psychological Reviews.* Vol. 86, No. 1, 44-60., 1979.

[Wittgenstein 53] *Philosophical investigations,* G.E.M. Anscombe, Oxford: Blackwell. 1953.

[Woods 75] Woods William A. What's in a Link: Foundations for Semantic Networks. In D.G. Bobrow, and A. M. Collins (Eds), *Representation and Understanding: Studies in Cognitive Science.* Academic Press, 1975.

[WUPAI 85] Proceedings of the Workshop on Uncertainty and Probability in Artificial Intelligence. Sponsored by AAAI, University of California, Los Angeles, August 1985.

[Zadeh 84] Zadeh L.A. Test score semantics as a basis for a computational approach to the representation of meaning. UCB/ERL M84/8.