Integrating Rules and Connectionism for Robust Reasoning:

A Connectionist Architecture with Dual Representation

Ron Sun

Technical Report CS-91-160
(Ph.D Dissertation)

Brandeis University
Computer Science Department
Waltham, MA 02254

July 18, 1991

ABSTRACT

Integrating Rules and Connectionism for Robust Reasoning:
A Connectionist Architecture with Dual Representation

(A Dissertation Presented to the Faculty of the Graduate School of Arts and Sciences of Brandeis University, Waltham, Massachusetts)

by Ron Sun

One of the more difficult problems for artificial intelligence is the problem of modeling commonsense reasoning and alleviating the brittleness of traditional symbolic rule-based models. This work attempts to tackle this problem by combining rules with connectionism in an integrated framework. This idea leads to the development of a connectionist architecture with dual representation that combines symbolic and subsymbolic (feature-based) processing for evidential robust reasoning: CONSYDERR.

Collins' protocols are analyzed based on the notions of *rules* and *similarity*, and are modeled by the architecture which carries out rule application and similarity matching through the interaction of the two levels. In order to understand rule encoding in the architecture, a formal analysis of the model is performed, which shows that it handles a superset of (propositional) Horn clause logic and Shoham's logic. To further improve the rule-based reasoning capability of the architecture, a solution to the connectionist variable binding problem is proposed. This work also explores the notion of causality and shows that commonsense causal knowledge can be well represented by CONSYDERR. Several other aspects of the architecture are discussed to demonstrate how connectionist models can supplement, enhance, and integrate symbolic rule-based reasoning.

An important aspect of this research is that the architecture utilizes the synergy resulting from the interaction of the two different types of representation and processing, and is thus capable of handling a large number of difficult issues in one integrated framework, such as partial and inexact information, cumulative evidential combination, lack of exact match, similarity matching, inheritance, and representational interactions, all of which are important elements of commonsense reasoning. The results suggest that connectionism coupled with rule-based symbolic processing capabilities can be effective and efficient models of reasoning for both theoretical and practical purposes.

ACKNOWLEDGEMENTS

First of all, members of my thesis committee have helped me a lot in pursuing this research: I want to thank Dave Waltz, my thesis advisor, for introducing me into the field of neural networks and connectionism, for his general guidance throughout this work, and for his careful readings of numerous rough drafts of papers, thesis proposals, and the thesis; his unwavering enthusiasm for massively parallel AI is contagious and provides tremendous support for doing connectionist research; I want to thank James Pustejovsky for spending many hours with me discussing the research that leads to this thesis, for sharing his valuable insights on AI, and for his extremely helpful comments and suggestions on the manuscript; I want to thank Tim Hickey for many very useful discussions that helped me to clarify formal aspects of my thesis and to better elucidate my roughly expressed ideas; Pattie Maes serves as the fourth member of the committee and she provided valuable feedback.

I am also grateful to some other faculty members and a number of fellow graduate students (Larry Bookman, Don Smith, and others) for helpful discussions, critical comments, and encouragements that have helped me in carrying out this research. The writing center at Brandeis helped me to polish up the final draft. I have benifited from interactions with many other people, inside and outside Brandeis, who have given me useful comments, suggestions, and helps in various ways.

The excellent computing facility provided by the computer science department makes doing this research possible. My three year stay at Brandeis is supported by fellowships, scholarships, and assistantships generously provided by the university and the department, and also in part by the Defense Advanced Research Projects Agency under contract # F 49620-88-C-0058 administered by the U.S. Air Force Office of Scientific Research.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

It is quite clear now that connectionist models are capable of implementing rule-based reasoning in a variety of ways (e.g., Hinton & Touretzky 1986, Barnden 1988, Sun 1989, Ajjanagadde & Shastri 1989, Lange & Dyer 1989, etc.), as well as other types of reasoning, such as similarity-based reasoning, categorization, or domain-specific reasoning tasks. The following questions arise:

- Can connectionist models do more in terms of accounting for robust, flexible, and multifaceted human reasoning, especially in the types of reasoning that are most often dealt with by rule-based paradigms?
- How can those data that are difficult to be accounted for by symbolic rule-based system paradigms (cf. Klahr et al 1987 and Buchanan & Shortliffe 1984 for a treatment of symbolic rule-based systems) be explained by connectionist reasoning models?

The questions can also be asked the other way around:

- Can we develop some computational mechanisms that can *better* model existing psychological data of commonsense reasoning and common patterns in the data (such as in Collins & Michalski 1989) and that can deal with some nasty brittleness problems that have been plaguing symbolic reasoning systems for so long, with the hope of enhancing our understanding of commonsense reasoning in general and connectionist reasoning in particular?

The work that will be discussed in this thesis tackles the above questions, by actually building and analyzing a connectionist model as a computational mechanism for accounting for the aforementioned data.

What this work tries to show is that connectionist models of reasoning are not necessarily mere "implementations" of their symbolic counterparts; rather, they can be better computational models of commonsense reasoning than any existing ones. They can be better because they can take into consideration the approximate, evidential and flexible nature of rule-based reasoning, and similarity based inferences (or a limited form of analogy), and they can also account for the spontaneity and parallelism in reasoning processes, which are very important but often neglected aspects of reasoning.

A number of other issues will also be addressed in this work, because they are so intrinsically related to the task of accounting for commonsense reasoning data mentioned above. These issues include how to encode rules in connectionist models, the relations between logic and connectionism, the representation of commonsense causal knowledge, and representational interactions (i.e. interactions in representation).

This chapter aims to provide an introduction, presenting the problems that motivate the work described in subsequent chapters, and also to provide a summary of what this work is all about. We will first briefly introduce the problem of common reasoning patterns in commonsense reasoning and the difficulties involved. And we will then outline a solution. The rationales for doing this research will be enumerated. Some clarifications of ideas and some guides for readers complete this chapter.

1.2 The Problem of Common Reasoning Patterns

In this section, the problem of modeling common patterns in commonsense reasoning is identified, and the difficulties involved are sketched out. A solution is then outlined, which is the main thrust of this work.

1.2.1 The Existence of Patterns

This thesis is neither about the study of a particular domain, nor about idiosyncratic reasoning in a particular domain. It deals with commonsense reasoning [1] in general, or more specifically it is about common, recurrent *patterns* in commonsense reasoning across various domains.

Do there exist any domain independent, recurrent common patterns in reasoning, especially in commonsense reasoning after all? This is a controversial question — there is no consensus regarding it.

Since the time of Aristotle, it has been widely recognized that reasoning seems to follow syntactic patterns, the simplest of which are the syllogism, *modus ponens*, and *modus tollens*. These patterns have been studied extensively and later developed into modern mathematical logic with the work of Frege, Russell, and others. There is no doubt that these patterns are extremely limited in their expressive and reasoning power (they cannot even describe formal mathematical reasoning completely). It is because of such limitations that some researchers have opposed altogether the use of formal systems (including formal logic) in studying cognition. I find that this point of view is less than fully justified: scientific progress is made by the long process of discovering new models and discarding old ones, and formal models can provide deeper understanding than informal ones. The basic belief behind this work is that, even though there has been up to now no satisfactory formal model of patterns of reasoning available, there is no reason to give up searching for such models, therefore resorting to *ad hoc* methods for dealing with the representation and use of knowledge; this is true especially in light of the new developments in connectionist models, which provide a new kind of formalisms that may prove to be of value in the study of commonsense reasoning.

Allan Collins, a well-known psychologist, collected a large number of protocols of commonsense reasoning (see Collins 1973 and Collins & Michalski 1989), in the area of elementary geography and the like. Noticing the inadequacy of traditional logic in explaining those reasoning patterns, he argues for the development of different formalisms or frameworks in the study of common reasoning patterns found in various commonsense reasoning tasks. Although they reject classic logic, Collins and Michalski (1989) believe in the existence of common patterns (vs. domain specific ad hoc mechanisms) that are widely applicable across domains.

Let us look at some data (from Collins & Michalski 1989). One protocol is as follows:

> Q: Do you think they might grow rice in Florida?
>
> R: Yeah. I guess they could, if there were an adequate fresh water supply, certainly a nice, big, warm, flat area.

[1] It is extremely difficult to define what commonsense reasoning is, just as hard as to define what intelligence is, or what knowledge is. Roughly speaking, it is taken to mean informal reasoning in everyday life regarding mundane issues, etc., where speed is more critical than accuracy.

In this example, the person answering the question deduced a conclusion based on partial knowledge with some information (concerning fresh water) absent.

Another case is as follows:

Q: Is the Chaco the cattle country?

R: It is like western Texas, so in some sense I guess it's cattle country.

Here, because there is no known knowledge, an uncertain conclusion is drawn based on similarity with known knowledge.

Yet another case is

Q: Are there roses in England?

R: There are a lot of flowers in England. So I guess there are roses.

Here the deduction is based on property inheritance (England HORTICULTURE flower; rose IS-A flower; so England HORTICULTURE rose), and the conclusion is partially certain.

Collins and his colleagues have done a good job in terms of analyzing the data and establishing a unifying framework for explaining them (for example, Collins 1975, Collins & Michalski 1989). What is needed is a computational mechanism from which various inference patterns contained in the data can emerge into existence. That mechanism ought to be analytically simple, structurally unified, and mechanistically sound. It is also important to account faithfully for the speed and spontaneity of human reasoning processes (for example, by adopting massively parallel structures).

1.2.2 How Can We Account for Those Patterns?

Although having some helpful characteristics, existing connectionist models, or any computational models for that matter, so far cannot deal very well with these above patterns in a single model.

The problem with existing connectionist systems is that they are too unstructured, too uniform, and sometimes not accurate enough (see Chapter 2). They learn the mapping from input data to output data by minimizing an overall error measure, and thus can perform the mapping only approximately most of the time. Although some of these features can help to give rise to the generalization capability in connectionist networks, they also give rise to the inaccuracy and critical errors.

The problem with existing rule-based approaches is the brittleness problem of symbolic systems in general, a problem that has been plaguing symbolic AI for a long time. Those above examples are actually instances of the problem (see Chapter 2 for more detailed discussion). We can equate the brittleness problem with the inability of a system (in a systematic way within a unified framework) to deal with reasoning involving partial information, uncertain or fuzzy information, no exact match, rule interactions, bottom-up inheritance, top-down inheritance, and learning new rules. And it is also difficult for rule-based systems to achieve the corresponding massive parallelism that is apparent in the data.

In order to account for the aforementioned patterns, we have to be able to deal with these difficult aspects of the brittleness problem. Detailed analysis of these aspects (in Chapter 2 with a lot of examples) shows that, while they look like a disparate set of problems, they can all be characterized as reasoning with rules supplemented with (feature) similarity-related inferences.

1.2.3 Precision and Approximation

One important issue that emerged from the above brief look at the problems with existing models is how we should handle the rigor and clarity in reasoning as evident in the data on one hand, and the approximate, evidential character of the same process on the other hand.

From detailed analysis of the data (see Chapter 2 and 3), we know that we need clearly-defined structures to enable effective inferences and we need precise ways of encoding knowledge possessed by a cognitive agent: the exact prerequisites for an action, the precise outcome of a given situation, etc. This rules out some types of models (for example, conventional connectionist models) as unsuitable, because of their imprecise nature.

The same data also shows that there is much flexibility in the reasoning of a cognitive agent. In order to model this, we also need corresponding flexibility in the reasoning process of our mechanism. Specifically, we need means of evidential combination, with graded information (fuzzy, uncertain, etc.) and capable of accumulating the confidence level incrementally (see Chapter 5). We should also be able to deal with analogous knowledge (as shown in the above protocols; see also Chapter 2).

To satisfy the requirements on both sides, we have to be very careful to strike a balance between them. This places a major constraint on the design of a reasoning system to account for commonsense reasoning patterns.

1.2.4 A New Approach

While the detailed specification and derivation of my solution to the above problem will be given in later chapters, some sketches are presented below to provide a general picture.

In this thesis, based on the detailed analysis as outlined above, a connectionist model for robust reasoning CONSYDERR [2] is proposed to account for some common reasoning patterns found in commonsense reasoning and to remedy the brittleness problem. In this system, a dual representational scheme is devised, which utilizes both localist representation (corresponding to the conceptual level reasoning) and distributed representation with features (corresponding to the subconceptual level reasoning). Each representation is in a subnetwork (or a "level") by itself, forming a separate feedforward network, with extra connections established between corresponding nodes across levels. We explore the synergy resulting from the interaction between these two types of representations (and between the two levels of the network), which helps to address problems such as partial information, lack of exact match, property inheritance, rule interaction, etc. Because of this synergy, the system is capable of accounting for many difficult patterns in commonsense reasoning. What can be accomplished by this system shows that connectionist models of reasoning are not just an "implementation" of their symbolic counterparts and they are actually better computational models of commonsense reasoning. And they can also be a good model for constructing practical systems.

In each level of the CONSYDERR architecture, rules are encoded as links between nodes (representing concepts, propositions, or features), and the computation at each node is so structured that it can implement the evidential combination, within rules and across rules, correctly. This computation, as will be shown later on, is meant to capture commonsense causal reasoning, and surprisingly enough does not require anything more than the weighted-sum computation with multiple sites in each node competing with each other for activation (cf. Feldman & Ballard 1982).

A number of experiments are conducted to demonstrate the utility of the architecture: it is capable of accounting perfectly for the reasoning data in Collins & Michalski (1989), it shows interesting properties in controlled experiments with systematically generated data, it is capable of storing and reasoning about a large number of data in the geography area, and it is suitable to be applied to a wide range of other domains.

[2] It stands for *a CONnectionist SYstem with Dual-representation for Evidential Robust Reasoning.*

1.3 What is the Point?

Given all the above, one may still ask the question: what is the point of doing all that? Although the answer to this question is already implied by the above discussion, to make it clearer, this section will articulate the answer to the above question directly.

First of all, CONSYDERR is an *integrated* architecture, simple but capable of dealing with a wide range of problems far beyond what the complexity level of the system will suggest, such as evidential rule-based reasoning, similarity matching (of analogous knowledge), top-down and bottom-up inheritance, and dealing with incompleteness and inconsistency of rules through interaction and generation (see Chapter 3), for which the Collins protocols provide the needed justifications for their roles in the working of a cognitive agent and their inclusions in a computational model of commonsense reasoning. Moreover, CONSYDERR is meant to be a it better computational model for accounting for these characteristics in commonsense reasoning, by using evidential rule application (i.e. cumulative evidential combination; see Chapter 2) plus similarity matching, motivated by the dissatisfaction with existing models, rule-based or connectionistic, in accounting for the robustness and evidentiality of commonsense reasoning. To say the least, a model accounting for commonsense reasoning protocols as the one developed herein can provide useful insights into the cognitive reasoning capabilities and possible computational mechanisms for modeling it.

Second of all, this work is an attempt at addressing the brittleness (fragility, rigidity, etc.) problem of symbolic AI, by combining rules with connectionist models. It will be shown that the CONSYDERR architecture is a useful framework in which several aspects of this brittleness problem (as discussed in Chapter 2) can be readily handled.

Thirdly, the capability of CONSYDERR in dealing with, in one unified framework, many difficult problems in reasoning can be credited to the dual representation (localist, or symbolic, and distributed, or subsymbolic) scheme used, and the synergy resulting from interactions between the two types of representation. It will be shown that the interaction between the two levels of CONSYDERR with different kinds of representation helps to perform intensional (meaning-oriented, feature-based) reasoning along with extensional reasoning. The key here is the *synergy* of the hybrid representation, which is one of the major points of this work and will be fully demonstrated throughout the thesis.

Fourth, an important aspect of this work is to explicate the relationship between rules and connectionist models, especially the weighted-sum model. This work shows, by formal mathematical derivations and plenty of examples, that there is little, if any, difference between rules and weighted-sum connectionist models in terms of their static descriptive power. In other words, they are somehow equivalent in their *static* behavior: each of them can express the kind of relations the other can (for details see Chapter 5). The difference however is in their *dynamic* behavior: while in symbolic systems rules are isolated (or "modular"), disembodied, context-free pieces of knowledge, in connectionist systems they are more integrated, more embodied in the overall process of reasoning, and they interact with each other (see Chapter 6) dynamically. Those "rules" in connectionist systems are represented by, or embodied in, connections (or links) between nodes (as will be discussed in details in Chapter 5). Another way of differentiating them is that, *syntactically*, connectionist models are more similar to rules than commonly thought. Differences between them are in terms of their semantic contents: one tends to be at a more abstract level, and the other more grounded in lower level processes.

Finally, we show, by theoretical construction, that connectionist models provide a framework for representing and reasoning about commonsense causal knowledge. As will be demonstrated in Chapter 5, weighted-sum connectionist models are equivalent to a fuzzy evidential logic (FEL) which in turn is a generalization of a modal logic formalism (Shoham 1987) for capturing causal relations and reasoning about them. The basic idea is that different conditions exert different weights, or play different roles, in obtaining causal outcomes, which can be expressed as modality as in Shoham (1987) or as weighted-sum computations for evidential combination as in CONSYDERR. Therefore, connectionist models are well-suited for causal reasoning and representation and, conversely, the surprising power of simple weighted-sum connectionist models can be attributed largely to this capability.

To summarize, this work provides a better model of commonsense reasoning patterns; it combines rules and connectionism, so on one hand it can cope with the brittleness problem of rule-based reasoning and on the other hand it enables connectionist networks to represent rules and causality and reason about them effectively; it also demonstrates the static/syntactical similarity and dynamic/semantical difference between rules and connectionist models; and it shows that the synergy resulting from the interaction between two types of representation is the key to its capability.

1.4 Some Clarifications and Disclaimers

This section will discuss briefly a number of issues that have to be clarified before we can proceed to the detailed study.

Rules are mentioned throughout the thesis, and therefore it should be made clear that the word is used in several different senses: it can be used to refer to *forms*, i.e., structures for encoding knowledge of various forms that consists of conditions and a conclusion so that whenever the conditions are met (in some ways) the conclusion can be reached; it can also be used to refer to *contents*, such as traffic rules, grammatical rules, or rules for doing arithmetic. These are two distinct senses, not interchangeable. In this thesis we will use the word "rule" in one sense or the other, or sometimes in both senses. Most of the time, it should be self-evident. Wherever there is a possibility of confusion, it will be indicated in which sense the word is used.

Learning is an important issue and intrinsically tied to the representation and reasoning problem. It was argued (by Hanson & Barr 1991 for example) that the biggest advantage of connectionist models is that they provide a way to study representation and learning together. Although this particular point may be debatable, I fully recognize the importance and pervasiveness of learning and the intrinsic relation and interaction between learning and representation. However, since the representation issues comprise such a huge topic, and because of the limited length of this thesis, I believe that it will be better to concentrate on one issue only and perform an in-depth and thorough study instead of providing superficial coverage of many points across a large spectrum. Therefore I will only treat the representational issues in producing the kind of reasoning we aim for, and I will not look into the learning issue in any depth.

Brittleness is one of the main issues dealt with in this thesis, and some remedies based on the CONSYDERR architecture are proposed, but nevertheless I do not at all claim that all issues related to the brittleness problem are resolved completely. As a matter of fact, it is far from that. We cannot solve all the problems at once; progress, I believe, is made by consistent, incremental advances.

Domain independence is assumed in this study, based on the recurrent, domain-independent nature of commonsense reasoning patterns. And I believe in the existence of domain independent and generic mechanisms, and their use in various cognitive processes, at least to a certain extent (otherwise how can you explain the same types of reasoning processes used for some totally different tasks?). However it is not my intention to join in any meta-level debate regarding such issues, and I fully understand and appreciate the importance of domain specific (as opposed to domain independent) study.

Formal logic is criticized in various ways throughout this thesis. However it should be made clear that I fully appreciate the usefulness of formal logic as a normative and partial analysis of cognition and in providing deep, mathematical insights into the matter. In other words, it has an important role in the study of intelligence and in the building of intelligent systems.

Natural language issues and categorization issues are not addressed in this thesis. These are major issues in cognitive science that are better dealt with in research more specifically geared toward them.

Grand unified theories of cognition are *not* what I am aiming for, and therefore this work should not be questioned along that line. For example, I did not answer questions such as: How can the system be connected to sensory-motor systems? How can it represent spatial relations? Or how does a system like that capture the ideas proposed by philosophers X, Y, and Z? although studies of such questions can be fertile grounds for further development of this theory.

1.5 The Organization of the Thesis

In order to make clear the organization of this thesis, a preview is provided in this section. This preview will take the form of a summary of each chapter in the rest of this thesis.

Chapter 2 starts by introducing Collins' protocols, and common reasoning patterns embodied in it. Then difficulties in accounting for these patterns by existing computational mechanisms are identified. This leads to the brittleness problem of symbolic AI in general. The brittleness problem is analyzed then in the context of commonsense reasoning. Based on this analysis, a rough solution is suggested, and applied to the protocol data.

Chapter 3 develops a computational architecture for implementing the solution for accounting for common reasoning patterns: first the need for multiple levels are identified, and then the appropriateness of connectionist implementation of each of these levels is argued through various evidence. Thus a generic architecture is formed. Further explications and specifications are made through analyzing the characteristics of the problems to be solved and deriving formally system parameters in accordance with these characteristics.

Chapter 4 contains a set of experiments performed on the architecture, including detailed explanations of how each of the protocols can be generated, systematic explorations of the architecture, and some applications of it.

Chapter 5 aims to answer the question of how rules can be coded by the weighted-sum computation as done in the above architecture. In so doing, the chapter looks into the question of causality, and how to capture commonsense causal knowledge in a rule-based fashion, through extending and combining existing rule-based frameworks. Based on this, a new logical formalism is defined and formally compared with existing ones, showing that it is capable of dealing with very general rule-based reasoning; it is then applied to address the problems in causal reasoning, and it fully demonstrates the usefulness of this logical framework in that regard. The formalism is shown to directly correspond to our above architecture and thus the two converge and retain all the benefits in one system.

Chapter 6 discusses further issues related to the inheritance hierarchy, the generative capability, and knowledge acquisition in the architecture, and shows that the system goes beyond symbolic rule-based reasoning.

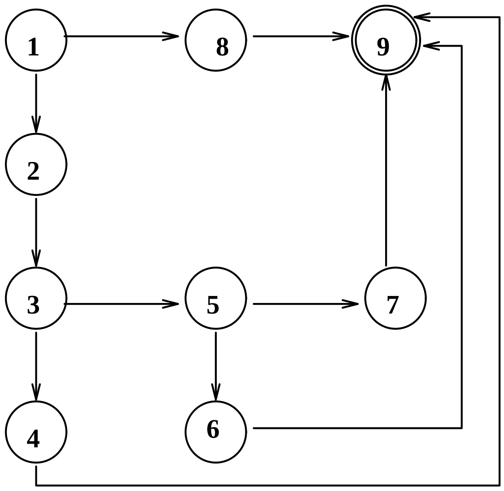


Figure 1.1: Dependency Among Chapters

Chapter 7 discusses the problem of variable binding and its solution in the architecture for extending its reasoning power, and presents some revised definitions of our systems taking this problem into consideration. The superiority of the solution is discussed briefly.

Chapter 8 reviews related work and backgrounds that lead to this work. Three basic paradigms are explored: rule-based reasoning, connectionism, and case-based reasoning. And it is shown that our system is capable of taking advantage of each of these paradigms, without being restricted by the peculiarity of any of them.

1.6 A Road Map for Readers

In order to facilitate the reading of this thesis, some suggestions regarding how to get to the relevant parts are necessary at this point.

A "dependency graph" depicting the logical relationships between the chapters is in Figure 1.1. It shows the prerequisites for reading each chapter. For example, if there is an arrow from 5 to 7, then Chapter 5 has to be read before chapter 7.

For people interested in different topics, different ways of going through the thesis are possible. For those who are interested in how to account for the Collins Protocol, and the common patterns embodied in it, Chapters 1, 2, 3, and 4 are the most relevant, as shown in Figure 1.2. For those whose primary interest is in reasoning, logic and connectionism, Figure 1.3 shows all the relevant chapters and their interdependence. For those who are concerned with the philosophical viewpoints embodied in this work but not all the details, Figure 1.4 shows the necessary chapters. Finally, for those who are only interested in connectionist models of rule-based reasoning, Chapters 1, 2, 3, 5, 7 are relevant as shown in Figure 1.5.

1.7 Summary

This chapter presents a sketch of the work contained in this thesis. Many important points or conclusions are stated without much elaboration. The rest of this thesis, starting from Chapter 2, will present detailed, step-by-step analyses of all the issues involved.

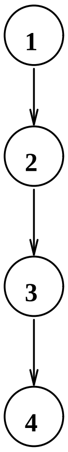


Figure 1.2: The Chapters for the Collins Protocols

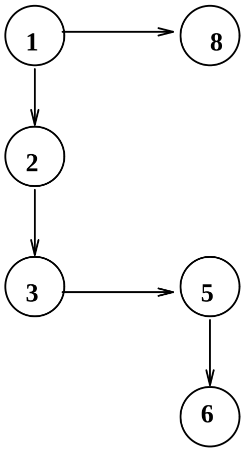


Figure 1.3: The Chapters Regarding Reasoning, Logic and Connectionism

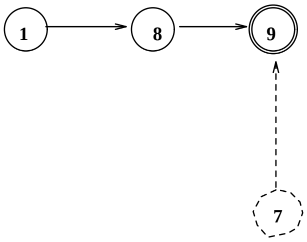


Figure 1.4: The Chapters Regarding the Philosophy Behind This Work

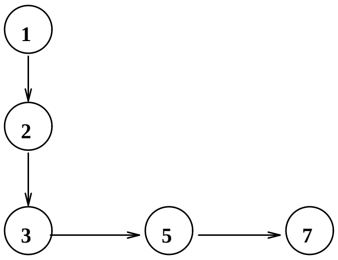


Figure 1.5: The Chapters Related to Connectionist Models of Rule-Based Reasoning

Chapter 2

Accounting for Reasoning Patterns: Rules and Similarity

2.1 Overview

The goal of this chapter is to examine protocol data of commonsense reasoning, and to make some sense out of it, so that we will be better prepared for developing computational mechanisms of commonsense reasoning. We will start by introducing the Collins Protocols, with brief examination of the reasoning process involved in each of them (the second section). We then expound on the data further to extract common characteristics and patterns from them (the third section). Then the difficulties in accounting for these patterns by existing computational mechanisms are identified. This leads to a discussion of the brittleness problem of symbolic AI in general (the fourth section). The brittleness problem is analyzed in the context of accounting for common reasoning patterns. Based on this analysis, a solution is suggested, and applied to the protocol data (the fifth section). Then some general discussion regarding rules and connectionism are made in connection with the suggested solution (the sixth section).

2.2 The Collins Protocols

We will look into a set of protocols collected and analyzed by Allan Collins and his colleagues, as in Collins (1978) and Collins & Michalski (1989) (three of these protocols were already presented in Chapter 1 and are included here for the sake of completeness). Accounting for these protocols, or more precisely accounting for common reasoning patterns in commonsense reasoning as embodied in these protocols, is the starting point, and a main goal, of this thesis. This section will present and briefly examine all the protocols to be considered, based on which the next few sections will identify patterns in them, reason about ways to explain them, and propose computational mechanisms for them.

(1) The first protocol shows uncertain, evidential reasoning:

> Q: Do you think they might grow rice in Florida?
>
> R: Yeah. I guess they could, if there were an adequate fresh water supply, certainly a nice, big, warm, flat area.

In this example, the person answering the question deduced an uncertain conclusion based on partial knowledge, with a piece of crucial information (i.e. the presence of fresh water) absent.

(2) the second protocol is as follows:

Q: Is the Chaco the cattle country?

R: It is like western Texas, so in some sense I guess it's cattle country.

Here because there is no known knowledge (in other words, no applicable rules), an uncertain conclusion is drawn based on similarity with known knowledge (or rules).

(3) the third protocol is:

Q: Are there roses in England?

R: There are a lot of flowers in England. So I guess there are roses.

Here the deduction is based on *property inheritance*. Formally, England HORTICULTURE flower; rose IS-A flower; So England HORTICULTURE rose, to use the jargon of the inheritance theory. The conclusion is only partially certain and is drawn because there is no information to the contrary (i.e. no *cancellation* of properties).

(4) The fourth protocol is:

Q: Is that [Llanos] where they grow coffee up there?

A: I don't think the savanna is used for growing coffee. The trouble is the savanna has a rainy season and you can't count on rain in general [for growing coffee].

This protocol shows a chain of reasoning: Llanos is a savanna, savanna has a rainy season, and rainy seasons do not permit coffee growing.

(5) The fifth protocol is:

Q: Is Uruguay in the Andes Mountains?

A: It's a good guess to say that it's in the Andes Mountains because a lot of the countries [of South America] are.

Here there is no rule stating whether Uruguay is in the Andes or not. However, since most South American countries are in the Andes, the *default* is therefore *in the Andes*. Uruguay just "inherits" this default value (although incorrectly).

(6) The sixth protocol is:

Q: Can a goose quack?

A: No. A goose – well, it's like a duck, but it's not a duck. It can honk, but to say it can quack. No. I think its vocal cords are built differently. They have a beak and everything. But no, it can't quack.

Two patterns are present here. One is based on similarity between geese and ducks, independent of knowledge regarding geese, yielding the conclusion that geese may be able to quack. Another pattern is a rule: since geese do not have vocal cords built for quacking, they cannot quack.

(7) The seventh protocol is:

Q: Is Florida moist?

A: The temperature is high there, so the water holding capacity of the air is high too. I think Florida is moist.

In this example the concepts involved are not all-or-nothing, but somehow graded, so the conclusion must be graded, or lower in confidence, corresponding with the confidence values of known facts and rules.

(8) The eighth protocol is:

> Q: Will high interest rates cause high inflation rates?
>
> A: No. High interest rates will cause low money supply growth, which in turn causes low inflation rates.

This example shows a chaining of rules: High interest rates will cause low money supply growth, and low money supply growth will cause low inflation rates, so high interest rates will cause low inflation rates.

(9) The ninth protocol is:

> Q: What kind of vehicles are you going to buy?
>
> A: For carrying cargo, I have to buy a utility vehicle, but for carrying passengers, I have to buy a passenger vehicle. So I will buy a vehicle that is both a utility and a passenger vehicle. For example, a van.

This example shows the additive interaction of two rules: if carrying cargo, buy a utility vehicle, and if carrying passengers, buy a passenger vehicle. The result is the combination of the two rules: something that is both a utility vehicle and a passenger vehicle. [1]

(10) The tenth protocol is:

> Q: Do women living in that [tropical] region have short life expectancy?
>
> A: Men living in tropical regions have short life expectancy, so probably women living in tropical regions have short life expectancy too.

This is another case of using similarity because of the lack of direct knowledge.

(11) The eleventh protocol is:

> Q: Are all South American countries in the tropical region?
>
> A: I think South American countries are in the tropical region, because Brazil is in the tropical region, Guyana is in the tropical region, Venezuela is in the tropical region, so on and so forth.

Although the conclusion is incorrect, this example illustrates *bottom-up inheritance* (a form of generalization). Since there is no knowledge directly associated with the superclass "South American countries" as to whether they are in the tropical or not, subclasses are looked at, and a conclusion is drawn based on the knowledge of the subclasses.

One can observe from these protocols that (cf. Collins & Michalski 1989):

- The same patterns are present in many different situations.
- In these patterns, people attribute relative weights to different pieces of evidence that bear on their conclusion.
- People are more or less certain about their conclusions depending on their certainty of information (including rules and facts).

The task we face is that of constructing a deep structure theory that is inspired by and extracted from the surface traces of the reasoning process (Chomsky 1980).

From these analyses, it is clear that, in order to model the above protocols, we need two things: Some means of encoding existing knowledge, and some means of making analogies (similarity matching) when there is no existing knowledge exactly for a particular situation; All protocols presented

[1] From this example on, we are no longer using Collins' protocols.

above indicate the existence of the two things. Only after determining the forms of these two things, can we come up with a "deep structure theory", namely a computational mechanism for the reasoning process.

We have now presented all the protocols necessary for our study [2]. In the next section we are going to study how to account for them, which will lead eventually to a connectionist architecture.

2.3 Patterns of Reasoning

In this section we will summarize what we see in the above protocols, that is, the basic, recurrent *patterns* of commonsense reasoning. Based on this, later sections will propose some computational mechanisms for accounting for these protocols.

Collins & Michalski argue for the existence of *common patterns* of reasoning, not just a large collection of unrelated instances of reasoning. I fully agree with this point of view. However, I find their analysis of the data into patterns is less than fully convincing. Moreover, the computational mechanism they propose based on this analysis (a logic formalism) lacks specificity, that is, there is no complete description of the algorithms used, no full characterization of features and properties of the logic, and no analysis of computational tractability. Therefore, we need to perform our own analysis of the reasoning patterns to find a better solution to the problem.

First of all is the question of the proper form of knowledge representation for applying existing, directly-applicable knowledge, such as in protocols 1 and 7. There are certainly many alternatives available. However, by all accounts, *Rules* seem to be the best choice as a sufficient and appropriate tool for expressing all kinds of knowledge, for the following reasons:

1. Rules are the most common form of knowledge representation, used widely in all kinds of AI systems. It has been convincingly argued that other knowledge representation schemes can be transformed into logic (rule) based schemes (cf. Hayes 1977, Nilsson 1980, and Chomsky 1980)

2. All existing knowledge applied in the above protocols can be coded in rules, one way or the other, as pointed out briefly before and as will be explored in detail later.

3. Rules ensure modularity in representation, making the representation easy to construct and manipulate. However it is worth pointing out that this modularity is somewhat limited, as will be discussed in Chapter 6 (see also Heckerman 1986).

4. Rules are formal yet flexible, allowing incorporation of confidence measures, uncertain knowledge, and plausible inference processes.

Having chosen rules as the general form of knowledge representation, we have to look deeper to determine what characteristics the rule representations must possess in order to account for commonsense reasoning. First of all, the reasoning process is *evidential*. By this is meant that the existing knowledge, or rules, are not deterministic, a priori, or transcendentally true. Rather, they are empirical, inexact, and uncertain. Rule representation schemes have to take into account these characteristics.

The concepts or propositions involved in reasoning processes are often graded, that is, not all-or-nothing but fuzzy, possibilistic, or probabilistic (cf. Dubois & Prade 1988). For example, "warm" is a fuzzy concept and there is no fixed boundary as to what is warm and what is not, similarly, the proposition "raining causes flooding" is probabilistic, not deterministic at all. We can associate with each of those concepts and propositions a generic confidence measure that can be used to facilitate the reasoning process.

[2]For issues regarding protocol analyses, see Posner 1989

The evidence combination process is cumulative, or in other words, it "adds up" various pieces of evidence to reach a conclusion, with a confidence that is determined from the "sum" of the confidences of the different pieces of evidence. Knowing two conditions in a rule results a larger confidence than knowing only one. For example, in the protocol regarding whether Florida is a rice-growing area, if we know all the four conditions, warm, flat, big and with fresh water supply, we can make the conclusion with full confidence; when we know only three of the four conditions, we reach the same conclusion with less confidence.

Different pieces of evidence are weighted, that is, each of them may have more or less impact, depending on its importance or saliency, on the reasoning process and the conclusion reached. We have to find a way of combining evidence from different sources cumulatively and with weights, without incurring too much computational overhead (such as in probabilistic reasoning or Dempster-Shafer calculus).

Another question is how we should apply analogous knowledge, in situations where we have no pre-established knowledge, i.e., making analogies or similarity matching, such as in protocols 2 and 6. What we have to do is to compare two concepts, two propositions or two situations, within the current context, and come up with an overall similarity measure. Then we can deduce conclusions based on this similarity. The confidence we have in the conclusion will be determined based on the degree of the similarity. In order to do this, we have to consider simultaneous comparisons of many pieces of existing knowledge. It is my view that the comparison process that decides similarity ought to be an intuitive and informal process, because the above protocols do not indicate anything deliberative or analytical. The process is rather informal, unstructured, and holistic. In order to model this informal process, we will have to adopt a rather informal and flexible mechanism.

From the above protocols, it is evident that applications of existing knowledge (rules) and analogy making (similarity) are intrinsically mixed together. By combining the two processes, many interesting inferences can be made with relative ease. In other words, it is the *synergy* between the two processes that creates these interesting reasoning patterns. Therefore, it is important to study this synergy and come up with a computational architecture within which this synergy can be created and utilized.

It is also important to notice that the reasoning process is parallel and spontaneous in nature. The retrieval and application of existing knowledge is done so fast that there is no room for a search or trial-and-error process, and therefore it must be done largely in parallel. This is also the case with the comparison of analogous knowledge. Multiple comparisons take place simultaneously and spontaneously, without external control and interference. Furthermore, the two processes, applications of existing knowledge and comparisons of analogous knowledge, are concurrent and interwined. For example, in the protocol about geese, the application of a rule regarding vocal cords is interwined with the similarity matching with ducks.

These protocols can also be analyzed in other ways or from different perspectives. These patterns may be divided into the following categories (as in Collins & Michalski 1989) :

- Derivation from mutual implication, in which particular values of different entities are related.
- Derivation from mutual dependence, in which functional relationships between two entities are exploited.
- GEN-based transformation, in which what is known about a particular class is generalized to its superclass.
- SPEC-based transformation, in which what is known about a class is specialized to a subclass.
- SIM-based transformation, in which what is known about a class is mapped to another class based on similarity.

- DIS-based transformation, in which what is known about a class is excluded from another class based on dissimilarity.

These categories, as will be discussed, can be combined in various ways.

In summary, some preliminary analyses regarding the protocol data are presented, and it is shown that there exist some common patterns in the reasoning processes as in the protocols.

2.4 Brittleness of Rule Based Reasoning

In this section the difficulties we encounter in accounting for the common reasoning patterns will be characterized, and it will be shown that the problem is the *brittleness* of symbolic rule-based reasoning and that these protocols manifest many aspects of this problem.

The first question one would ask, after looking at the above protocol data, is: Can conventional rule-based reasoning models account for the above reasoning? My answer is no. First of all, due to the informal, unstructured, and holistic nature of similarity matching (the lower level characteristics in the hierarchy for cognitive functionality; cf. Dreyfus & Dreyfus 1987), conventional rule-based reasoning, the characteristics of which are quite the opposite, can not possibly be used to model similarity matching. Secondly, even for the rule application part, it is difficult to capture the evidential, cumulative, and graded nature of the reasoning in a computationally efficient manner within a unified framework in conventional rule-based reasoning: fuzzy logic (as in Zadeh 1988) does not deal with cumulative evidence, the probabilistic approach does not deal with graded concepts, so on and so forth. Besides, most of these rule-based approaches are computationally too expensive.

Then the next question is: *Why* can't conventional rule-based reasoning do the job? My contention is that the inability of conventional rule-based models to account for these reasoning patterns is due to the brittleness of these models. To put it another way, these patterns exemplify various aspects of the brittleness problem, and that is why they are difficult to account for.

In fact, the term *brittleness* has been around for quite a while, and different authors have ascribed somewhat different meanings to the word But basically, "Brittleness" suggests being easily breakable: the slightest deviation in input data from what is exactly known by a system can cause a complete breakdown of the system. Specifically, it can be qualified as the inability of a system to deal with in a systematic way within a unified framework the following aspects of reasoning:

- Partial information (e.g., the first protocol),
- Uncertain or fuzzy information (e.g., the first protocol again),
- Lack of matching rules (e.g., the second protocol),
- Combinational rule interactions – lack of consistency and completeness resulting in a fragmented rule base (e.g., the ninth protocol),
- Top-down inheritance (e.g., the third protocol),
- Bottom-up inheritance (e.g., the 11th protocol),
- Changing contexts and new rules.

Conventional rule-based systems cannot solve these problems. Neither can conventional connectionist models, as they are now. We need to develop some new approaches to deal with the problems.

The brittleness problem is more pervasive than one might think. With the exception of some extremely specialized narrow domains, it shows up in all kinds of reasoning in various domains:

- The brittleness problem exists in *decision-making*: when there are no precisely specified preconditions and actions, a typical rule-based system cannot proceed without further information regarding the situation or without additional mechanisms (cf. Hayes-Roth et al 1985). So rule-based systems are *brittle* when applied to this domain. To avoid this brittleness problem, every possible scenario of combinations of conditions and decisions has to be well analyzed and structured into the system, which is not always possible, especially for large systems.

- The brittleness problem exists in *diagnosis*. Diagnosis is the interpretation and classification of data (Hayes-Roth et al 1985) and a conventional method is specifying *rules* for interpreting input data. No matter how detailed a rule set is, there are in practice always some difficult situations or some unexpected interactions that may break down a system. For large systems, it is virtually impossible to test every possible situation, and this leads to a fragile system.

- The brittleness problem exists in *planning*. The classical planning paradigm is rule-based: a set of rules are supposed to completely specify which step to take in a given situation. Since rules are often insufficient in completely specifying a situation, errors and dead-ends are inevitable, and we therefore need to perform a search, or a trial-and-error process (although some remedies have been applied to such planning systems (cf. Nilsson 1980 and Hendler 1989)). Solving the brittleness problem thus may well lead to a better, computationally more efficient planning paradigm.

The difficulty of conventional rule-based reasoning in dealing with all commonsense reasoning domains signifies the range and importance of the problem. A large number of attempts have been made to solve this problem, yet it still seems far away from having a satisfactory solution. It is paramount to be able to better deal with this problem, before we can devise truly intelligent systems performing robust reasoning. [3]

So far we have explained the brittleness problem and the importance in coping with it in order to be able to perform robust reasoning, especially in accounting for the protocols given above. Among all aspects of the brittleness problem, learning new rules is really a separate issue and should be dealt with that way, as indicated in Chapter 1. While the remaining aspects of the problem still look like a disparate set of problems, they can all be characterized as reasoning with rules supplemented with similarity-related inference, which will be shown in the next section.

2.5 Towards a Solution

In this section we will point out ways to overcome each aspect of the brittleness problem and thus to deal with the Collins Protocols. These aspects can all be characterized as rule application plus similarity matching (the same for all the protocols presented above). Detailed analysis follows.

Let us define **rule application** and **similarity matching** first. A *rule* is defined here to be a structure consisting a left-hand-side (LHS) and a right-hand-side (RHS); and numerical weights are associated with each fact in LHS. Whenever a left-hand-side is activated (to a degree commensurate with the confidence of the corresponding facts), the corresponding RHS will be activated to a degree determined by multiplying the activation value of LHS by the weight, when there is only one fact in LHS. When there are more than one facts in LHS, the activation is calculated by multiplying the vector of all activation values in LHS with the vector of all weights (i.e. calculating inner products), [4] We will denote a rule by $A \longrightarrow B$. And if A is activated, the activation of B due to A is denoted as $(A \longrightarrow B) * A$.

[3]The term *robust reasoning* is used here to mean the kind of reasoning that are free from the problem of brittleness.

[4]The justifications for this definition of rules will be given later in Chapter 5.

Similarity can be viewed as a measure of the amount of overlap between the corresponding feature sets of two concepts or two propositions. [5] We will denote similarity by $A \sim B$. So if A is activated, the activation of B due to A is denoted as $(A \sim B) * A$, that is, the activation of A times the similarity measure between A and B.

Features involved in similarity matching are mentioned in very general terms here, because many different kinds of features can be used for different domains and tasks. Sometimes we may be able to use pre-specified set of features for a reasoning task in a certain domain. Sometimes for certain domains we may want to have features that are directly linked to perception or other basic cognitive processes and are developed or formed through the interaction with the environment, so that the computational mechanism using the features may be more realistic and accurate. This research will not be concerned with the problem of how features are formed, because that is a major research area itself, but will instead take whatever features available in relevant sourcebooks and the like.

Now we will analyze each aspect of the brittleness problem in turn. These aspects encompass the reasoning in all the protocols presented earlier.

(1) When we have inexact information, we can quantify the inexactness with a confidence value, and the confidence value will be applied to rules directly:

$$A \longrightarrow B$$

and A is activated to a degree commensurate to its confidence level, so [6]

$$B = (A \longrightarrow B) * A$$

(2) When we have incomplete information, that is, when we do not have all the requisite conditions to apply a rule, we can still deduce a conclusion, although with less confidence (like a guess, more or less), because we deduce the confidence of a conclusion based on the vector multiplication computation (i.e. inner products, a simple extension of scalar multiplication, which will be justified in detail in Chapter 5). Suppose we have rules

$$ABC \longrightarrow D$$

When given A and B (and C is unknown), D is deduced with less confidence than when given A, B and C: [7]

$$D = (ABC \longrightarrow D) * (A \;\; B \;\; 0)$$

(3) The no-match-rule situation can be described as:

$$A \sim B$$

$$B \longrightarrow C$$

and A is activated ($A \neq 0$). So we have

$$C = (B \longrightarrow C) * B = (B \longrightarrow C) * (A \sim B) * A$$

(4) For top-down inheritance, suppose A is a subclass of B and B has a property value to be C, then [8]

$$A \sim B$$

[5] Again, detailed definitions and analysis will be given later.

[6] Here $(A \longrightarrow B)$ represents a number, and "*" is multiplication.

[7] Here $(ABC \longrightarrow D)$ is actually three numbers (measures), and they are applied to the conditions of the rule (A B C) to get the weighted-sum (or inner-product), as an extension to the case mentioned above with simple multiplication.

[8] The super/sub-class relationship is treated here as a special case of the similarity relationship, although it is normally much stronger than the general similarity case (see Chapter 3).

$$B \longrightarrow C$$

When A is activated, C will be activated accordingly, i.e. $C = A * (B \longrightarrow C)$.

(5) For bottom-up inheritance, suppose A is a superclass of B and B has a property value C, then

$$A \sim B$$

$$B \longrightarrow C$$

When A is activated, C will be activated accordingly, i.e. $C = A * (B \longrightarrow C)$. [9]

(6) For cancellation of inheritance, suppose A is a superclass (or subclass) of B and B has a property value C, then

$$A \sim B$$

$$B \longrightarrow C$$

$$A \longrightarrow D$$

when A is activated, D will be activated more than C.

(7) In case of rule interaction, we can describe the situation as

$$A \longrightarrow C$$

$$B \longrightarrow D$$

$$C \sim D$$

when A and B both are activated, the interaction of C and D might result in something else being strongly activated, depending on their mutual similarity.

It is useful at this point to compare this analysis with Collins & Michalski's account. Collins & Michalski divided these patterns into the following categories (as explained earlier): Derivation from mutual implication, Derivation from mutual dependence, GEN-based transformation, SPEC-based transformation, SIM-based transformation, and DIS-based transformation. According to my analysis, the first two categories have little difference and both are dealt with by rule applications. The rest of the categories are similarity related: generalization and specialization are special cases of the similarity relation; Therefore, they can be dealt with by similarity matching. [10] Overall, all the cases are described and dealt with by rules plus similarity in my analysis.

In the analysis by Collins & Michalski, the confidence of the conclusions reached depends on a number of parameters:

- Conditional likelihood
- Degree of certainty
- Degree of typicality of a subset within a set
- Degree of similarity of one set to another
- Frequency of the referent in the domain of the descriptor

[9] It is believed that the bottom-up inheritance is less reliable that the top-down case, although they are both based on similarity. In addition similarity relations are not symmetric or commutative (see Chapter 3).

[10] Dis-similarity based inference is extremely un-reliable: it is generally not the case that, just because the things are different in some particular aspects, they are different some other aspects. Therefore, this kind of inference is not considered here.

- Dominance of a subset in a set
- Multiplicity of the referent
- Multiplicity of the argument

According to my analysis, however, a smaller set of parameters can be identified: rule weights and similarity. These two parameters can subsume the above parameters used by Collins & Michalski: the first two — conditional likelihood and degrees of certainty — can be easily captured by rule weights; the rest can be accounted for by similarity measures or a combination of rule weights and similarity measures.

The combination of rules and similarity is also analogous in a way to case-based reasoning (CBR): retrieval of relevant cases, similarity matching, and adaptation. In this case the retrieval is accomplished automatically – we consider all existing knowledge at the same time. Similarity matching for finding the best case is done in a massively parallel fashion – all cases are matched against the current case at the same time. Adaptation here is done with changes in confidence values and rule interactions, which will be detailed later. However, there are some significant difference between my approach and that of CBR as will be evident along the way.

To summarize, a solution has been outlined that involves both rule application and similarity matching, and a simple framework has been developed. All cases encountered are successfully analyzed in this framework. Although the logical next step will be the development of an architecture (i.e. a computational mechanism) that implements the above proposed solution in a natural way, we will first pause to take our stock, i.e., have a brief look at what kinds of computational tools or components we have on hand for building such an architecture.

2.6 Some Reflections on Rules and Connectionism

This section will present some general reflections on rules and connectionism. This discussion is related but not necessarily limited to the problems and tasks explained above; It however will have important bearing on the further development and complete specification of the solution within connectionist frameworks. In a way, this section clears the stage for developing solutions to problems of robust reasoning.

Rules are efficient computational constructs for a compact, modular representation and direct, efficient reasoning. They can be implemented in pure symbolic forms, such as in mathematical logic or in Prolog. This form has the advantage of having well-defined semantics and well studied inference algorithms based on a clearly defined set of axioms. However this form lacks the ability to handle inexact information which leads to particularly severe brittleness (or *rigidity* as used in Barnden & Srinivas 1990). Another form is the combined symbolic and numerical representation, for example, probabilistic reasoning, fuzzy logic, and the MYCIN certainty factor model. These models deal with the brittleness problem to a certain extent, but none of them can really solve the problem since each of them only deals with one or two of the above aspects (see Chapter 8). We need a more integrated approach.

Rule-based systems can be parallelized (for the purpose of, e.g., speeding up reasoning processes to match the speed of human reasoning), but that will not help much either in terms of brittleness. Parallel systems are required to be serializable, and most of the parallel systems do nothing more than fire multiple rules at the same time, when they do not interfere with each other; thus there is no fundamental difference between serial and parallel rule-based systems. [11] We have to look elsewhere for the solution to the brittleness problem.

[11] Massive parallelism in connectionist systems, however, has a totally different flavor, as will be discussed later.

From a cognitive perspective, rule-based systems have the characteristics of being symbolic, dealing with mostly analytical concepts in a domain. It seems, therefore, in order to really solve the brittleness problem, there ought to be something else to supplement rule-based reasoning, utilizing different kinds of knowledge.

Connectionism can be characterized as models having a large number of simple processing elements with a large number of links between pairs of those elements forming extensive interconnections. Connectionist models are inherently massively parallel. Connectionist models have the advantage of being robust, exhibiting generalization, graceful degradation, fault and noise tolerance, etc. Connectionist reasoning and learning are often said to be similarity-based: It usually involves looking at previous similar cases, either individually or collectively in a statistical way (cf. McClelland & Rumelhart 1985 and Anderson & Rosenfeld 1988).

However in its simple form, connectionism lacks some elements necessary for high level cognitive capability: rule application, compositionality, explanation, etc. In other words, it does not have sufficient symbol processing capabilities. Inaccuracy results from this lack of symbolic capability: such systems usually do not have precisely specifiable preconditions and actions, only approximate mapping from input to output obtained by minimizing an overall error measure (usually a small error remains). comparing rules and connectionism, it seems that they are complementary to each other. Because of this complementary nature, it might be a good idea to combine the two.

The question then is how to combine them, so that they can complement each other and cooperate in reasoning. There is no easy answer to this. Preferably the two should be combined in a principled, not ad hoc, way. The next chapter will be concerned with this question.

It is important to point out that we are not talking about the *forms* of rules and connectionist models, but the respective perspectives, approaches, and the semantics; for, in terms of basic forms, they can be viewed as equivalent as will be shown in chapter 5. [12]

In summary, rule-based systems and connectionist models each has some unique characteristics that are indispensable to modeling cognitive agents and each complement the other nicely. Hence it is suggested that the two should be combined, in a principled way, to utilize the two respective approaches in a single integrated system.

2.7 Summary

We introduce in this chapter the problem of accounting for common patterns in commonsense reasoning. The Collins Protocols are explained and analyzed briefly. From this analysis, it is concluded that the current rule-based approach alone is not sufficient to account for the reasoning patterns embodied in these protocols, and therefore other types of reasoning, namely similarity matching, are also needed. The difficulties with rule-based systems is characterized as the brittleness problem, which includes various aspects that is shown to be able to be characterized by two common themes: rule application and similarity matching; thus a solution is outlined that involves both rule application and similarity matching. Finally, we stressed the need for combining rules and connectionism in a principled way to form an integrated architecture, which clears the stage for the later development of such an architecture.

The three goals for the rest of this thesis are

- Accounting for Collins' protocols, dealing both with rule application and similarity matching and coping with the brittleness problem.

[12] Another way to see this is: both connectionist networks and rule-based systems (e.g. Prolog) are Turing equivalent, and therefore the two are equivalent to each other.

- Devising a massively parallel architecture that carries out reasoning patterns in a natural way, utilizing the synergy between the two types of computation (rules and connectionism, or rule application and similarity matching).
- Developing and exploring various processing capabilities of the massively parallel (connectionist) architecture.

In the next chapter, we will focus on the second goal — developing a massively parallel architecture; after developing such an architecture, in Chapter 4 we will account for the Collins Protocols with the new tool. The third goal will be addressed then in Chapter 5 and 6.

Chapter 3

A Connectionist Architecture for Common Reasoning Patterns

3.1 Overview

...... Architecture,
Existing in itself, and not in seeming
A something it is not, surpasses them
As substance shadow.
— Longfellow, *Michaelangelo*

From the analysis in the previous chapter, it is clear that both explicit *rule applications* and fine-grained *similarity matching* (and the combinations/variations of the two) have to be dealt with. We need a unified computational mechanism for handling both rule application and similarity matching. This chapter will concentrate on developing such a mechanism.

In this development, we will try to answer the following questions:

- What components are needed?
- How do these components interact?
- What constraints do the problems to be solved impose on the system?
- How can we determine the system parameters to satisfy the constraints imposed?

As will be shown in detail, the answers to the above questions lead naturally to a connectionist architecture with dual representation combining symbolic and subsymbolic processing.

In the rest of this chapter, first the need for (at least) two levels in models of reasoning is identified (details are in Appendix); based on that, a generic connectionist architecture composed of two levels for symbolic and subsymbolic processing is outlined and gradually refined. In order to systematically setting up system parameters, a set of constraints obtained from the characteristics of the problems to be solved (i.e. rule application, similarity matching, inheritance, etc.) is analyzed; the correct parameter values are derived from the above constraints. Thus a computational architecture is completely developed. Some related questions, such as rule encoding, representational interaction, and chains of inheritance, will be addressed later in Chapters 5 and 6.

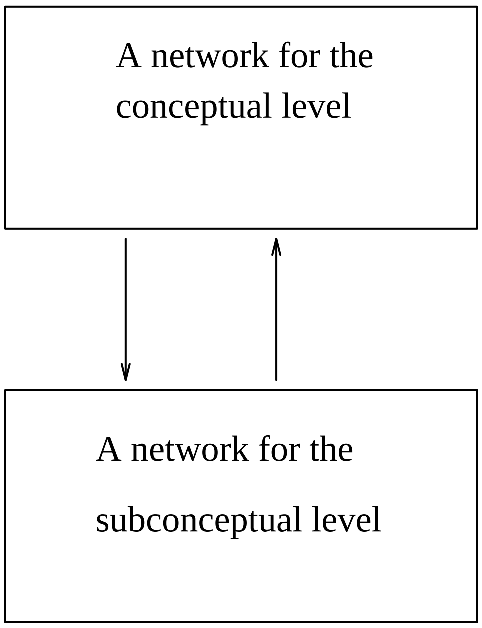


Figure 3.1: The Architecture 1

3.2 A Generic Architecture

Based on the discussion in the previous chapter on the difficult reasoning patterns, we are now ready to develop a generic architecture that are naturally integrated and can carry out all the reasoning patterns in a computationally efficient way.

Given the earlier conclusion that to account for all the patterns analyzed, it is necessary to carry out both rule applications and similarity matching, I argue that we need both a conceptual level reasoner and a subconceptual level reasoner, interwoven together, in order to deal with all these patterns. The conceptual level reasoner deals with mainly analytical concepts in a domain, based on knowledge encoded in rules. The other reasoner deals with more holistic, finer-grained, and less structured processing, namely similarity matching and associations. Moreover, I argue that both levels should be implemented in connectionist models, and should be tightly coupled in some way. Various evidence seems to support such a position; however, due to the abstract nature of the problem, it is impossible to firmly prove it one way or the other – it is more philosophical than scientific. Therefore detailed arguments will not be presented here, but they are contained in Appendix for those who are interested. For now it suffices to show that such a generic architecture for commonsense reasoning will work properly; details follow.

According to the above proposal, we can outline a very sketchy model of the architecture we are aiming for as in Figure 3.1. By dividing the architecture into two levels, we may be able to utilize the synergy between the two levels, that is, the whole system may be able to do things that neither of the components can do alone (the next two sections present a detailed analysis), and may be able to do them in a computationally efficient way (see Chapter 4 for examples and Chapter 8 for some comparisons).

In this model, first we need to implement conceptual level rule applications in a connectionist network. Since the network carries out reasoning at the conceptual level, the representation has to be explicit and exact, in order for the concepts and reasoning processes to be consciously accessible and linguistically expressible. This leads directly to the idea of local representation. Note that local representation (or its opposite, distributed representation) is not a clearly definable concept,

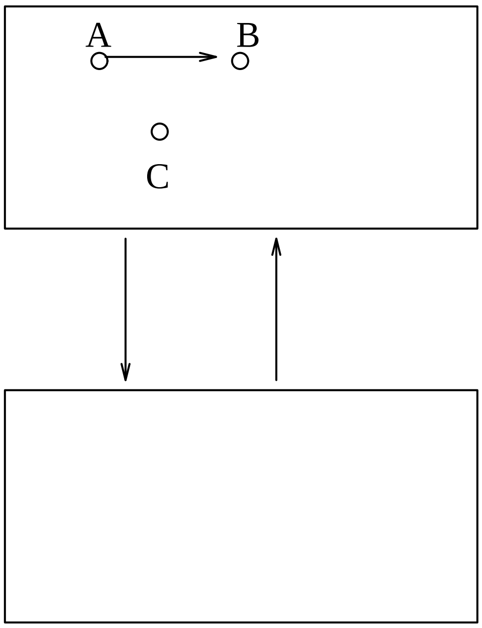


Figure 3.2: The Architecture 2

A, B and C are concepts, and the link between A and B represents a rule $A \longrightarrow B$ (for example, A = Chaco, B = Cattle-country, and C = Western-Texas).

and usually it is taken to mean that each concept or proposition in a domain is represented by a single node in a network of nodes (cf. Feldman & Ballard 1982). This is what we will do for the conceptual level. Based on that, rules are then implemented by, roughly speaking, links between nodes representing conditions and nodes representing conclusions.

Now that we have the representation (rules in this case), we need facilities for reasoning about them (based on rules). In this respect, we prefer to have some sort of *computation in place* (Waltz, personal communication), which means that the reasoning process does not need to perform indexing, retrieval, reorganization and updating of large databases of facts; rather, the computation is done *in place*, where the data is, without the overhead of selecting data and moving data around, by directly connecting together those pairs of facts that are related by rules so that only local computation for passing on activations is necessary. By this way we can achieve massive parallelism more easily, and it is cognitively more plausible (cf. also Minsky 1985). So according to these ideas, we link up a node that represents a fact in the condition of a rule with a node that represents a fact in the conclusion of the rule, so that rule applications become straightforward signal propagations. Thus we have a slightly refined working model as shown in Figure 3.2.

Commonsense reasoning patterns are, by nature, evidential, cumulative, and graded, as has been discussed before and elsewhere in great length, such as Collins and Michalski (1989), Zadeh (1965), Pearl (1989), and Sun & Waltz (1991), and references cited therein. Given the inexactness of commonsense reasoning processes, our rule encoding scheme has to deal with this issue. Hence a scheme for evidential reasoning is proposed in this work: FEL (for *Fuzzy Evidential Logic*), which uses in fact only weighted-sums as the main operation. Thus it coincides with the operation of typical connectionist models and consequently has an easy implementation in a connectionist network. FEL will be explained in great detail in Chapter 5, so that the explication of it (which is very long itself) will not interfere with our main line of presentation here.

At the subconceptual level, in order to carry out similarity matching based on features, we adopt a distributed representation with each node corresponding to a feature, a fine-grained element of the

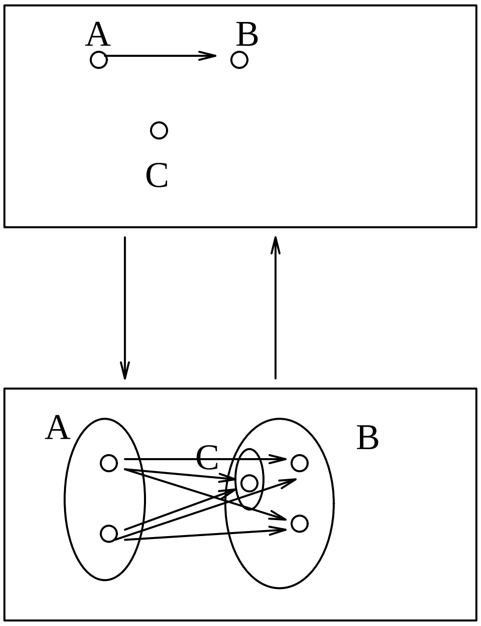


Figure 3.3: The Architecture 3
A, B and C are concepts, and the link between A and B represents a rule $A \longrightarrow B$ (for example, A = Chaco, B = Cattle-country, and C = Western-Texas).

meaning of a concept or a proposition. This feature representation is *similarity-based*, that is, the amount of overlap between two sets of nodes representing two different concepts is proportional to the degree of similarity between these two concepts or propositions. The links at the subconceptual level are the replications of the links at the conceptual level (which represent rules, by the way); that is, if there is a link between two nodes (or two concepts they represent) at the conceptual level, then we will add a link between each node in the feature set of the first concept and each node in the feature set of the second concept (the full cross-bar connection between the two feature sets), replicating diffusely the original link at the conceptual level. [1] So we have a further refined working model as in Figure 3.3.

The interaction between the two levels can have a top-down path and a bottom-up path separately (for more flexibility as discussed in Appendix), so we refine the working model as in Figure 3.4. We denote the conceptual level subnetwork CL (Connectionist network with Local representation), and the subconceptual level subnetwork CD (Connectionist network with Distributed representation).

Also, the operation of the system can be in cycles and the interaction of the two levels does not have to be ever-present (as discussed in Appendix). One cycle in the operation of the system can be divided into three phases: the top-down phase, the settling phase, and the bottom-up phase, in which top-down flows only occur during the top-down phase and bottom-up flows only occur during the bottom-up phase; in other words, the two levels have *phasic* interactions. So we now have the following working model as in Figure 3.5.

One possible set of equations for describing the computation during the three phases is:

For the top-down phase,

$$x_i(t+1) = \max_A f(A(t))$$

[1]The reason for doing this is given in Appendix, related to the idea of incorporating analytical knowledge into intuition.

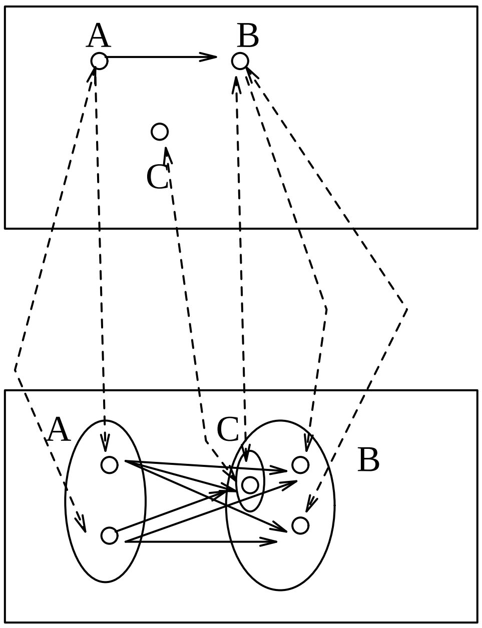


Figure 3.4: The Architecture 4
A, B and C are concepts, and the link between A and B represents a rule $A \longrightarrow B$ (for example, A = Chaco, B = Cattle-country, and C = Western-Texas).

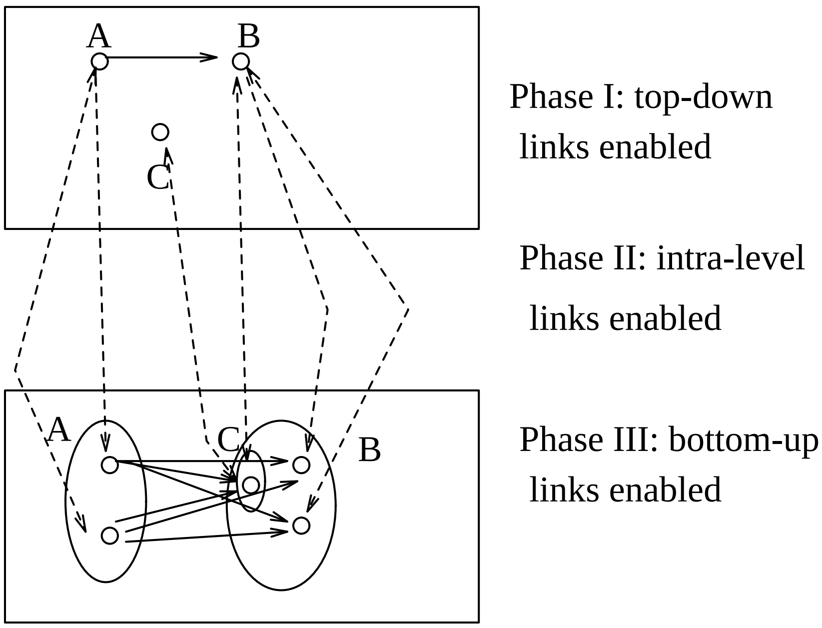


Figure 3.5: The Architecture 5
(nodes and links are the same as before)

where A is any node in CL that has $x_i \in CD_A$; that is, a node in CD receives activation from the corresponding node in CL, and chooses the largest value.

For the settling phase,

$$\Delta A = \alpha \sum W_i I_i(t) - \beta A(t)$$

and

$$\Delta x_i = \mu \sum w_i i_i(t) - \nu x_i(t)$$

where W_i, w_i are link weights (that represent strength or weights in FEL rules), and I_i, i_i are the activation of related facts (conditions or premises); that is, in this case, each node receives activation from other nodes at the same level and does a weighted-sum for incrementing its own activation.

For the bottom-up phase,

$$B(t+1) = \max(B(t), \sum_{x_i \in CD_B} g(x_i(t)))$$

where B is any node in CL, and g is a monotone increasing function; that is, a CL node receives activation from its corresponding CD node, and chooses the value as its activation if it is greater than its original activation.

Applying this cycle, one possible scenario is as follows:

> First some nodes in CL get activated by external inputs (and clamped). Then the top-down phase will activate (and clamp) the CD nodes corresponding to the active CL nodes. In the settling phase, links representing rules related to those activated nodes take effect in both CL and CD. Based on similarity, concepts may have overlapping CD representations, so some of the CD representations will be partially activated if a concept *similar* to them is activated in CD. Finally in the bottom-up phase, fully or partially activated CD representations will percolate up to activate the corresponding nodes in CL. The result can be read off from CL.

Each node in the system has one or more sites (cf. Feldman & Ballard 1982), each of which computes the weighted-sum (or any other similar functions whenever needed) of the inputs. The maximum of the values computed by all the sites is taken to be the activation value of the node. [2] One possible variation is that instead of taking that maximum to be the node activation we can treat it as an increment to the previous node activation value. This results in a gradual, incremental process of node activation.

Notice the massive parallelism and spontaneity in the above specified architecture: activations are propagated, in a massively parallel and spontaneous fashion, from all pre-link nodes to all post-link nodes; each node receives inputs as soon as it can, and therefore fires as soon as it can, ensuring

[2] This is important for combining results from different rules and for carrying out MAXs in the above equations. It can be added to the settling phase equations:

$$A = \max_j A^j$$

$$\Delta A^j = \alpha \sum W_i^j I_i^j(t) - \beta A^j(t)$$

and

$$x_i = \max_j x_i^j$$

$$\Delta x_i^j = \mu \sum w_i^j i_i^j(t) - \nu x_i^j(t)$$

where W_i, w_i are link weights (representing strength or weights in FEL rules), and I_i, i_i are the activation of related facts (conditions or premises). j denotes different sites.

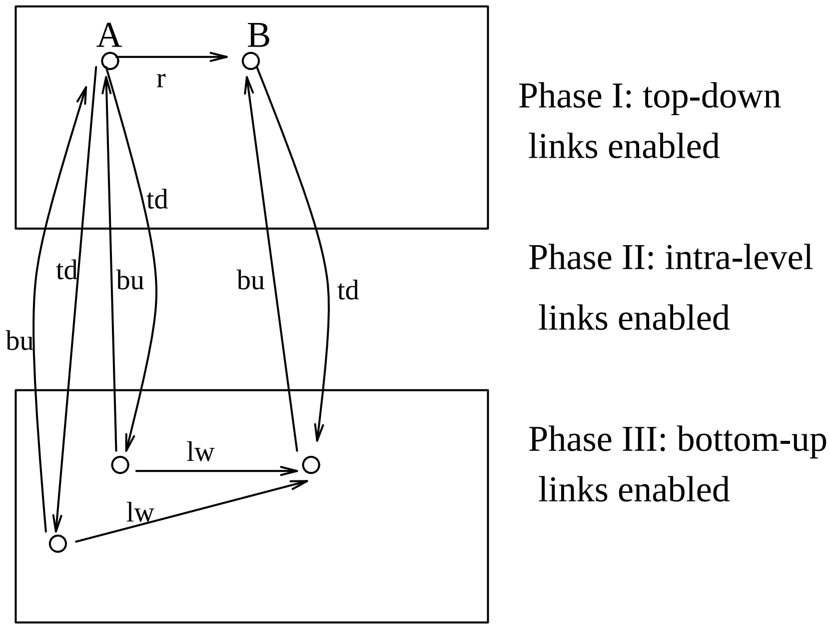


Figure 3.6: A Generic Model

a maximum degree of parallelism in terms of rule application. In terms of similarity matching, all similar concepts are activated (in their CD representations) immediately and simultaneously once an original concept is activated, and matched automatically with the original one (through top-down and bottom-up flows); thus the architecture is extremely efficient by employing the two level structure. The parallelism and spontaneity in this architecture account well for the similar parallelism and spontaneity in human reasoning processes as identified before. The massive parallelism provides a computational efficiency which no other existing connectionist models of reasoning (rule-based or case-based; see Chapter 8) can match.

We will call this architecture CONSYDERR, which stands for *a CONnectionist SYstem with Dual representation for Evidential Robust Reasoning.* A generic diagram is shown in Figure 3.5.

To conclude, this section develops a generic architecture accounting for common reasoning patterns, dealing simultaneously with rule application and similarity matching and coping with the brittleness problem in a massively parallel and spontaneous process. There are still some unspecified parameters and details left out, to be addressed later.

3.3 Fine-Tuning — A Set of Constraints

This section will consider the question of how to fine-tune (or set the values of the structural parameters in) CONSYDERR, so that the system will be able to perform the tasks ascribed to it at the beginning. In so doing, we will investigate data to be accounted for, requirements they impose, and theoretical considerations and desiderata discussed before.

In fine-tuning the architecture we first have to identify the parameters that have to be accurately specified. Examining Figure 3.6, we need to specify the following: weights for links between any node in CL and any node in CD (i.e. cross-level links, both top-down and bottom-up), and weights for links between two nodes in CD. In contrast, links between two nodes in CL, and their associated weights, are taken as given, which represent rules. [3]

[3] We will explain in detail how the rule encoding works in Chapter 6.

Now the question is: How should we delineate those parameters methodically, without being (or risking being accused of being) over restrictive on the one hand or over free-floating on the other. The approach we will take here is: (1) first determining the range of problems to be solved, (2) studying the data and problem characteristics, (3) figuring out a set of quantitative constraints, and finally (4) deriving the aforementioned parameters from those constraints. In the rest of this section, we will accomplish the first two steps, and the next section will deal with the last two steps. The reason for this separation is that this section deals only with conceptual analysis, and all the mathematics is left for the next section; readers who do not wish to go into mathematical details can skip or skim through the next section without losing track of the argument.

The range of problems to be solved can be determined as follows. Recall that a set of protocols were presented (in the second chapter) that are difficult to account for by existing conventional rule-based systems, connectionist models or other computational mechanisms, and the difficulty is identified as lying in the brittleness problem, which once solved, we would be able to account for all these protocols. Therefore we will, as our main task, attempt to deal with the aspects of the brittleness problem identified earlier.

Since according to the previous analysis, the brittleness problem can be dealt with by rule application supplemented with similarity matching, the two processes have to be dealt with in the system. **Similarity matching**, which is the first thing to be analyzed, has the following characteristics: [4]

- The degree of similarity from concept A to concept B [5] must depend on the amount of overlapping of the two corresponding feature sets, when everything else is equal. For example, suppose A is river-valley, lowland, temperate area, A′ is river-valley, lowland, tropical area, and B is temperate lowland, obviously A matches B better than A′ matches B.

- The degree of similarity from concept A to concept B must depend on the size of the feature set of B, when everything else is equal, because a larger feature set of B means there are a lot other features in B that do not match those of A. For example, suppose B is temperate plain and A is temperate area, A matches B quite well. Suppose B′ is temperate prairie plain, A matches B′ less well.

- The size of the feature set of A is not important in determining the similarity from A to B. For example, suppose B is temperate plain and A is temperate area, and further suppose A′ is temperate rainy area, the match from A to B is no less strong than the match from A′ to B, because one extra feature in A does not make B′s properties more or less true for A. What is important is what gets matched in A, not what does not get matched.

Besides the above points, an important consideration is whether similarity should be transitive, counter-transitive, or neither: that is, if A is similar to B and B is similar to C, should A be similar to C or not? There are two possibilities:

- There is something in common to all three A, B and C. In this case, A, B and C are pair-wise similar. So A *is* similar to C.

- There is something in common between A and B and something else in common between B and C. In this case, A is similar to B and B is similar to C, but A is *not* similar to C.

[4] For a general discussion of various similarity measures, see, for example, Tversky (1977), Thagard & Holyoak (1989), and Nosofsky (1984), etc.

[5] Here similarity from A to B means that, when there is no direct knowledge about A available, then we will go to B, which is similar to A, to find plausible but fallible answers.

Therefore, similarity is neither transitive nor counter-transitive, and should not be made to be either in implementation.

Similarly, since $A \sim B$ is a numerical measure, which is not commutative, i.e. $A \sim B \neq B \sim A$, similarity measures should not be made symmetric (cf. Tversky 1977).

Features used in similarity matching can be obtained from various sources, and it should not be part of the system specification to determine what features to use, because this task is highly domain specific. Using features makes potentially possible grounding reasoning processes in perceptions and other primitive processes, and therefore interacting with internal and external environments (as mentioned before). In this way, it is likely to map sensory similarity directly onto representational similarity, and to have concepts "defined" (at least in part) based on its similarity with other related concepts, so that coherence and mutual constraints in reasoning are naturally maintained.

It should be noted that there is really no context-free, universally applicable similarity measures. The above considerations, such as those regarding transitivity and symmetricity, are generalized from the examples (including the Collins Protocols) and are suitable for some large classes of problems, including all the tasks we apply the measure to (see chapter 4 for detailed examples), but are not meant to be universally correct.

A parallel issue is how to deal with **rule application**, which is very complex because it involves the questions of how to represent rules, and how to encode that representation of rules in connectionist networks, etc. To keep our focus and not to be pulled astray, we will only deal with very simple rules here, that is, rules with one condition and one conclusion. The extension of this and the full justification will be given later in Chapters 5 and 7. [6]

For a rule in the form of $A \longrightarrow B$, suppose A has a confidence value of a, and the rule has a strength (or a weight) r_{AB}, we want to be able to infer B with the confidence value $a * r_{AB}$ (that is, the confidence value of the condition modulated by the confidence value of the rule). This is intuitively appealing; the full justification will be given in Chapter 5 (cf. also Shultz 1990, Pearl 1988 etc.).

We also have to be able to deal with **mixed rule application and similarity matching** situations. For example, in one case, suppose A $\sim$ B, B $\longrightarrow$ C, we want to make sure that if A is activated, then B (the confidence value) = $s_{AB} * A$, and C =$r_{BC} * B = A * s_{AB} * r_{BC}$, where r_{AB} is the weight on the link between A and B and s_{AB} is the similarity between A and B. One instance of this is the following: Western Texas is similar to Chaco, Chaco is a cattle country, so Western Texas is probably a cattle country. Other cases include:

- A $\longrightarrow$ B, B $\sim$ C;
- A $\longrightarrow$ B, B $\sim$ C, C $\longrightarrow$ D;
- A $\longrightarrow$ B, B $\longrightarrow$ C, C $\sim$ D;

etc. In all of these cases, we want to achieve an intuitively correct result analogous to the first case.

Another thing we have to deal with is **inheritance and cancellation**. The reason for giving special consideration to this aspect, which seems to be just a mixture of rule application and similarity matching, is that inheritance and cancellation involves competition of mutually conflicting concepts (i.e. property values), which gives rise to some subtle requirements regarding system parameters. [7]

Here we will formulate the inheritance problem as inheritance of properties, that is, we express an inheritance problem in an "inheritance graph", which is a directed acyclic graph in which nodes

[6] For a general discussion of rules, see Posner (1989), Buchanan & Shortliffe (1984), Newell & Simon (1972), Klahr et al (1989), etc.

[7] For a general discussion of this topic, cf. Touretzky (1985), Shastri (1988) and Yager (1989), etc. See Chapter 6 for more details)

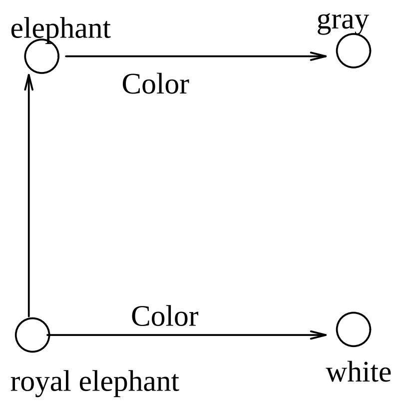


Figure 3.7: An Inheritance Graph

represent concepts and two types of conceptual links, *superclass/subclass* (or IS-A) and *have-property-value*, are used to connect pairs of nodes. See Figure 3.7 for an example. [8] This "inheritance graph" will have to be expressed as rules and similarity (and eventually have to be transformed into the CONSYDERR representation) in order to solve the problem in CONSYDERR. For example, the situation in Figure 3.7 can be described as

"Elephants are gray, but royal elephants are white. Royal elephants are elephants"

which can be expressed as rules

$$elephant \longrightarrow color\text{-}gray$$

and

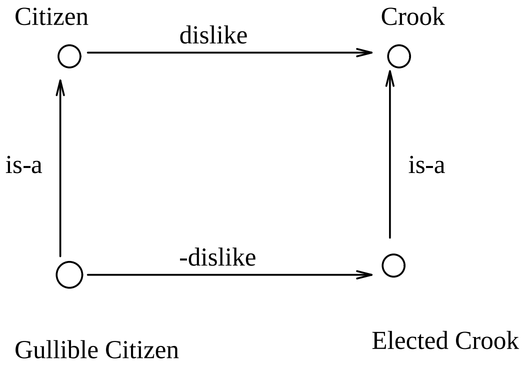
$$royal\text{-}elephant \longrightarrow color\text{-}white$$

As will be explained later, super/subclass (IS-A) links (such as in "royal elephants are elephants") are implemented implicitly through the containment relations in the corresponding feature sets of the two concepts or propositions, i.e., the feature set of "royal elephant" is a superset of the feature set of "elephant", because "elephant" is a more general concept and hence has less features (less specificity) associated with it; and "royal elephant" is a subclass of "elephant" and hence contain all the features of "elephant" plus some others that are unique to it. The other type of links (HAVE-PROPERTY-VALUE) are implemented explicitly as rules as shown above, in which LHS of a rule is a concept node and RHS is a property-value pair (for example, $elephant \longrightarrow color\text{-}gray$), or in other words, we treat a property-value pair as a concept in our system. Figure 3.8 shows an example inheritance graph, its description with rules and similarity, and its implementation in CONSYDERR (with rules in CL and feature set containment relations in CD).

Basically CONSYDERR should be able to deal with inheritance of properties (top-down), percolation of properties (bottom-up inheritance), and cancellation of inheritance. Let us look into the following cases in details (cf. Touretzky 1985).

Let A be a superset (or a superclass) of B (i.e. $A \supset B$), therefore the feature set of A is a subset of that of B (i.e. $F_A \subset F_B$). This is a well known principle of philosophical logic: the larger the extension, the smaller the intension, and the extension of a concept with an intension that is a proper subset of the intension of another concept is a superset of the extension of the other concept. Here intension simply means the "meaning" or "definition" of a concept, or more precisely a set of features that is common and jointly peculiar to a concept (cf. Leonard 1967); extension means the set of objects that are described by a concept, or by the corresponding intension.

[8] This is slightly different from the problem formulation in Touretzky (1985); for details, see Chapter 6.



(1) Inheritance graph

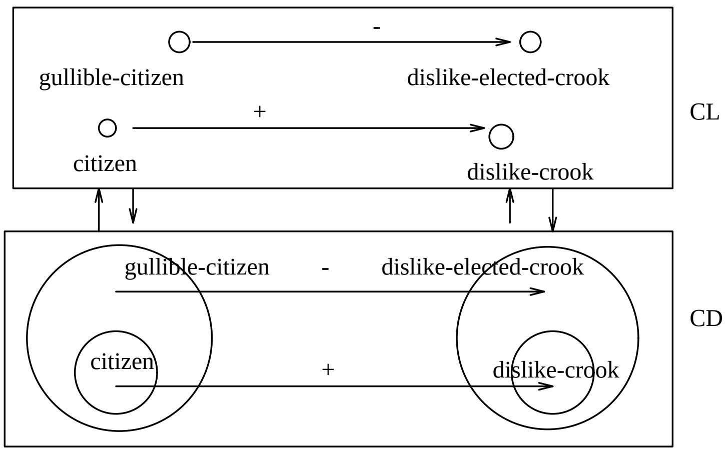
Gullible-Citizen ~ Citizen Citizen ~ Gullible-Citizen

Elected-Crook ~ Crook Crook ~ Elected-Crook

Citizen --> dislike-Crook

Gullible-Citizen --> - dislike-Elected-Crook

(2) Description



(3) Implementation

Figure 3.8: An Inheritance Graph, Its Description and Its Implementation with Two Levels
(In CD, one arrow is drawn in place of pairwise connections)

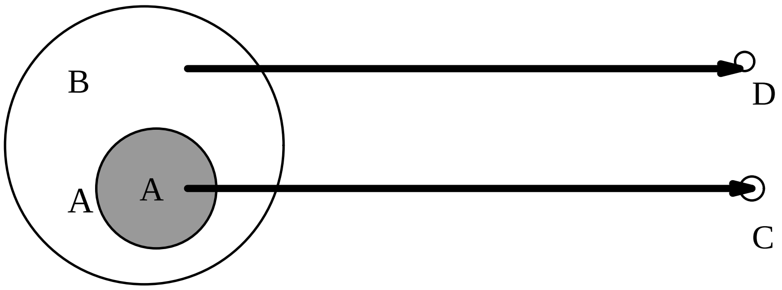


Figure 3.9: Inheritance Case I

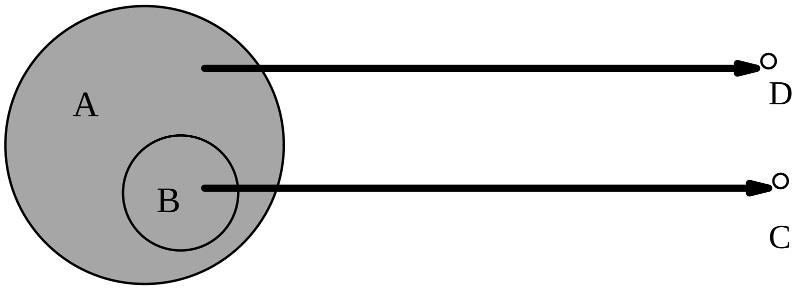


Figure 3.10: Inheritance Case II

Suppose A has a property value C, and B has no specified property value. If B is activated then C should be activated to a certain extent, from (top-down) inheritance. For example, temperate regions lie north to tropical regions in the northern hemisphere; subtropical regions constitute a subset of temperate regions; so subtropical regions also lie north to tropical regions in the northern hemisphere.

On the other hand, suppose B has a property value D, and A has no specified property value, if A is activated, D should be activated too, from the percolation of property values (bottom-up inheritance) from B, a subset of A, to A. For example, given that subtropical regions lie north to tropical regions in the northern hemisphere, and that the subtropical regions are a subset of temperate region, when there is no information to the contrary, we can infer, with certain confidence, that temperate regions lie north to tropical regions in the northern hemisphere.

What is most difficult to deal with is the cancellation of property inheritance. We will discuss the following cases:

1. As shown in Figure 3.9, A has a property value C and B has a property value $D \neq C$. if A is activated, C should win over D. For example, A = a warm, flat, fertile area with ample fresh water supply, C = rice-growing area, B = a warm, flat, fertile area with fresh water supply but rugged, mountainous, and D = non-rice-growing area. If given A, we should be able to deduce C not D.

2. As shown in Figure 3.10, A has a property value C and B has a property value $D \neq C$. if B is activated, D should win over C. For example, A = a warm, flat, fertile area with fresh water supply, C = rice-growing area, B = a warm, flat, fertile area with fresh water supply but rugged, mountainous, and D = non-rice-growing area. If given B, we should be able to deduce D instead of C.

3. As shown in Figure 3.11, A has a property value C and B has a property value D which is a subset of C. If A is activated, C should win over D. For example, A is a particular region

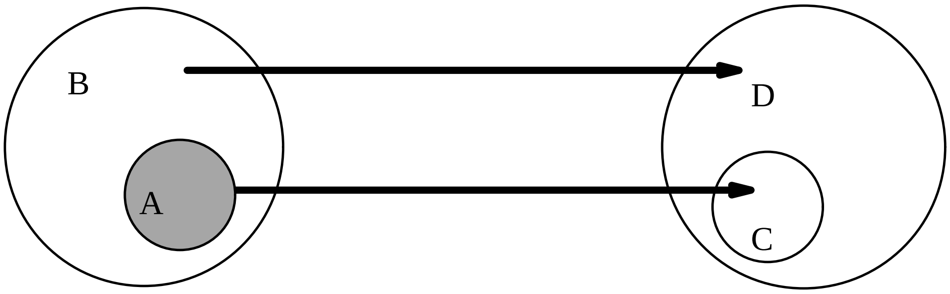


Figure 3.11: Inheritance Case III

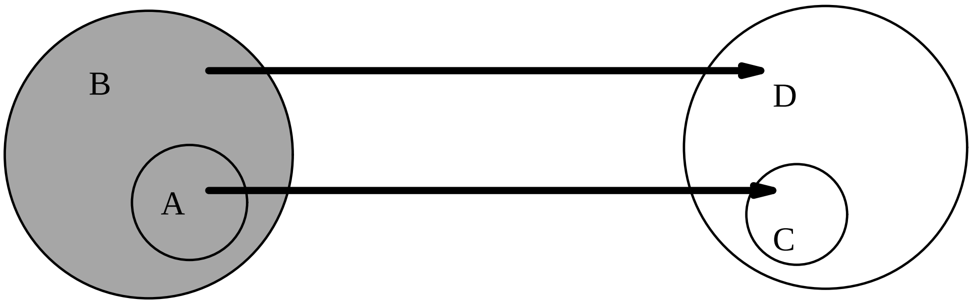


Figure 3.12: Inheritance Case IV

(say, the South), B is a subarea of A (say, the Southeast), C is fruit-growing area, and D is orange-growing area. So if A is given, C should have a higher confidence value than D.

4. As shown in Figure 3.12, A has a property value C and B has a property value D which is a subset of D. If B is activated, D should win over C. For example, A is a particular region (say, the South), B is a subarea of A (say, the Southeast), C is fruit-growing area, and D is orange-growing area. So if B is given, D should have a higher confidence value than C.

We do not deal with multiple inheritance here, because it is not essential for our purpose. It, however, will be discussed in chapter 6.

To conclude, we worked out a set of essential requirements and desiderata for several major problems to be solved. Based on these requirements, we will determine (in the next section) structural parameters of CONSYDERR to complete the development of this architecture.

3.4 From Constraints to Specifications

This section will present a formal analysis and derivation of the parameters (i.e. top-down weights, bottom-up weights, and CD link weights) of the CONSYDERR architecture. The discussion in the previous section indicates that in order to account for a certain range of reasoning patterns, the system has to be able to deal with the problems of similarity, rule application, mixed similarity matching and rule application, property inheritance, property cancellation, embedded properties, etc. These problems impose various requirements on the system, and thus we can derive a set of numerical constraints, which forms a foundation based on which we will build a system that can accomplish what we set out to do: accounting for common reasoning patterns as identified before.

The plan is as follows: The first subsection will formally present all quantitative constraints, and the second subsection will derive the parameters that satisfy the constraints.

As a preview, we summarize below in an informal way the formal derivation in the rest of this section: We first define formally the CONSYDERR architecture, crystallizing what we discussed before, so that we can later perform formal derivation and proofs (in the rest of this chapter and in Chapter 5 regarding its logical equivalences). We identify the three structural parameters to be determined: the top-down weights, the bottom-up weights, and the link weights in the CD level. To determine these parameters, we look at the requirements regarding rule application, similarity matching, and inheritance as enumerated before. Based on the desiderata associated with similarity matching, we choose a simple formula. Two cases of inheritance are analyzed to come up with the link weights in the CD level, and two other cases of inheritance are then analyzed to determine the bottom-up weights. We then determine the top-down weights in relation to the similarity measure and the other two parameters. Finally, the requirements for rule application are verified to be fully fulfilled.

The significance of this derivation and analysis lies in the fact that a connectionist architecture with a surprisingly simple two-level structure can solve a wide range of representation and reasoning problems effectively and computationally efficiently in a massively parallel manner; everything here is accomplished within a unified framework. However, it is not a complete solution to all the problems within the scope; quite to the contrary, it is a solution to a subset of the most common reasoning problems that can be solved effectively and efficiently with an extremely simple computational mechanism. This is what connectionist models are good at: a simple and efficient architecture for a large set, but not necessarily the complete set, of important problems in a certain area (cf. Shastri 1990 for similar points). In view of this result we can postulate that by integrating extensional representation (at the conceptual, symbolic level) and intensional representation (at the subconceptual, subsymbolic level), and thus forming a two-level architecture, one can solve a number of problems that cannot be dealt with easily by utilizing the synergy between the two components. The keys here are, I believe, (1) *intensional* representation (or in its simplest form feature representation) which can add tremendous power to extensional representation and reasoning systems, which prevail in AI today, [9] and (2) *synergy* between these two types of representations that results in systems that can solve easily problems that neither of the representations alone can solve satisfactorily.

3.4.1 Identifying All Constraints Quantitatively

All cases are unique, and very similar to others.
— T.S. Eliot, *The Cocktail Party*

It is clear that this model has to deal with the following problems:

- Similarity matching
- Rule application
- Inheritance of properties (top-down)
- Percolation of properties (bottom-up inheritance)
- Cancellation of inheritance

Let us examine quantitatively all constraints for each aspect based on the results of the previous section.

[9] Intensional representation enables finer-grained subsymbolic representation (cf. Dreyfus 1972 and Dreyfus & Dreyfus 1987) and meaning-oriented reasoning at a subconceptual level (vs. pure syntactical symbolic manipulation).

Similarity. For similarity matching to work correctly, as discussed before, we have to devise a *similarity measure* s_{AB} for measuring the similarity of B against A that will satisfy all of the following requirements:

> $s_{AB} \propto |F_A \cap F_B|$, that is, the similarity between two concepts or propositions is proportional to the amount of their feature overlapping. The more the two concepts or propositions have in common, the more similar they are, provided that everything else is equal.

and also

> $s_{AB} \propto \frac{1}{|F_B|}$, that is, the similarity is inversely proportional to the number of features B has, when everything else is equal, particularly the amount of feature overlapping. That is because the more feature B has, the smaller percentage of feature overlapping with A it enjoys.

This basically leads to the following possible definition:

$$s_{AB} = \frac{f_1(|F_A \cap F_B|)}{f_2(|F_B|)}$$

where f_1 and f_2 are functions.

Rules. For rules to work correctly in CD as well as in CL, the following conditions must be met, as outlined in the previous section (where $\sim$ represents similarity matching and $\longrightarrow$ represents rule links) (cf. Shultz 1990, Heckerman 1986, etc. regarding uncertain reasoning):

(1) A $\longrightarrow$ B, in this case we want to make sure that, if A is activated, then B= $r_{AB} * A$, where r_{AB} is the weight on the link between A and B (the same below);

(2) A $\sim$ B, B $\longrightarrow$ C, in this case we want to make sure that, if A is activated, then B= $s_{AB} * A$, and C $= r_{BC} * B$, where s_{AB} is the similarity between A and B (the same below);

(3) A $\longrightarrow$ B, B $\sim$ C, in this case we want to make sure that, if A is activated, then B= $r_{AB} * A$, and C $= s_{BC} * B$;

(4) A $\longrightarrow$ B, B $\longrightarrow$ C, in this case we want to make sure that, if A is activated, then B= $r_{AB} * A$, and C $= r_{BC} * B$;

(5) A $\longrightarrow$ B, B $\longrightarrow$ C, C $\longrightarrow$ D, in this case we want to make sure that, if A is activated, then B= $r_{AB} * A$, C $= r_{BC} * B$, and D $= r_{CD} * C$;

(6) A $\sim$ B, B $\longrightarrow$ C, C $\longrightarrow$ D, in this case we want to make sure that, if A is activated, then B= $s_{AB} * A$, C $= r_{BC} * B$, and D $= r_{CD} * C$;

(7) A $\longrightarrow$ B, B $\sim$ C, C $\longrightarrow$ D, in this case we want to make sure that, if A is activated, then B= $r_{AB} * A$, C $= s_{BC} * B$, and D $= r_{CD} * C$;

(8) A $\longrightarrow$ B, B $\longrightarrow$ C, C $\sim$ D, in this case we want to make sure that, if A is activated, then B= $r_{AB} * A$, C $= r_{BC} * B$, and D $= s_{CD} * C$;

(9) A $\sim$ B, B $\longrightarrow$ C, C $\sim$ D, in this case we want to make sure that, if A is activated, then B= $s_{AB} * A$, C $= r_{BC} * B$, and D $= s_{CD} * C$;

In all of the above cases, the network dynamics should result in correct node activation (ideally in both CL and CD, and in their final combined outcome) for each node involved according to an appropriate similarity measure (yet to be worked out in detail) and a rule application mechanism. [10] [11]

Inheritance. Let A be a superset of B (i.e A $\supset$ B), and therefore the feature set of A is a subset of that of B (i.e. $F_A \subset F_B$). [12] Therefore there is no need for explicit rules stating A is-superset-of B or B is-subset-of A (or B IS-A A). We have looked into the following cases (cf. also Touretzky 1985):

(1) Suppose A has a property value C, and B has no specified property value. If B is activated, then C should be activated to a certain extent.

(2) Suppose B has a property value D, and A has no specified property value, if A is activated, D should be activated too.

These two cases can be easily dealt with based on similarity, that is

$$B \sim A, \quad A \longrightarrow C$$

for the first case, and

$$A \sim B, \quad B \longrightarrow D$$

for the second case. We should also examine the following cases of cancellation:

(3) A has a property value C and B has a property value D $\neq C$. if A is activated, C should win over D.

(4) A has a property value C and B has a property value D $\neq C$. if B is activated, D should win over C.

(5) A has a property value C which has a feature set F_C, and B has a property value D which has a feature set F_D which is a subset of F_C (in other words, D is a superset of C). If A is activated, C should win over D.

(6) A has a property value C which has a feature set F_C, and B has a property value D which has a feature set F_D which is a subset of F_C. If B is activated, D should win over C.

Note that the exact magnitude of activation is immaterial here, since we can always use a *winner-take-all* network on top of the system we are developing to decide which conclusion is more credible.

3.4.2 Deriving the Parameters

With all these constraints established, we can proceed to derive the exact specifications of the parameters, and as a by-product, show that CONSYDERR can solve rule application, similarity matching, and inheritance all within one framework, when all the parameters, including top-down weights (denoted as td), bottom-up weights (denoted as bu), and weights for links between two CD nodes which diffusely replicate the rule links of CL (denoted as lw), are appropriately set (see Figure 3.6).

Let us first formally define the CONSYDERR architecture.

[10] We only deal with rules with only a single premise here; rules with multiple premises are just straightforward extensions of these cases (which we will show later on).

[11] There is no case above where there are two or more successive instances of similarity matching. This is because we do not want similarity to propagate, i.e. that A is similar to B and B is similar to C does not necessarily mean that A is similar to C (recall the earlier discussion regarding transitivity).

[12] As mentioned before, the larger the extension, the smaller the intension, and the extension of a concept with an intension that is a proper subset of the intension of another concept is a superset of the extension of the other concept.

Definition 3.1 *A* **Node Activation Function** *is a mapping between input vectors and the output value for a particular node, based on some weights and/or thresholds.*
For example, weighted-sum, thresholding, radial-basis functions, etc. or even their combinations.

Definition 3.2 *An* **Incremental Weighted-Sum Node Activation Function** *is a node activation function in the form of*

$$n = \max_a a$$

for node n, [13] *and*

$$\Delta a = \sum w_i i_i - a$$

where a is a site activation value, w_i's are weights, and i_i's are inputs to a site. [14]

Each site is used to represent a different rule, so that rules will not get mixed together.

Definition 3.3 *A* **CL** *network is a set of nodes, with incremental weighted-sum activation functions, and their connectivity: $< N, C >$, where N is a set of nodes (with incremental weighted-sum node activation functions) and C is a set of ordered pairs $\{(n_1, n_2) | n_1, n_2 \in N\}$.*

Definition 3.4 *A* **Hierarchical CL** *network is a CL network with a hierarchical structure, that is, acyclic connectivity.*

We want a network to be acyclic so that it will not have a loop in it that amounts to circular reasoning.

Definition 3.5 *A* **CD** *network is a set of nodes, with incremental weighted-sum activation functions, and their connectivity: $< N, C >$, where N is a set of nodes (with incremental weighted-sum node activation functions) and C is a set of ordered pairs $\{(n_1, n_2) | n_1, n_2 \in N\}$.*

Definition 3.6 *A* **Hierarchical CD** *network is a CD network with a hierarchical structure, that is, acyclic connectivity.*

Definition 3.7 *A* **CD Network Resulting From A One-To-Many Mapping From CL** *is a CD network satisfying the following condition: Suppose CL=$< N, C >$ and CD=$< M, D >$,and a one-to-many mapping L: $N \longrightarrow M$. $(m_1, m_2) \in D$ if and only if $(n_1, n_2) \in C$, and $m_1 \in L(n_1)$ and $m_2 \in L(n_2)$.*

This definition is meant to capture the intuition we have regarding feature representation as discussed in the previous section. The question is whether the CD network resulting from a one-to-many mapping from CL is still a hierarchical network. Generally speaking, the answer is no; but in specific applications, it can be made into a hierarchical network. [15]

[13] A thresholding mechanism can, but not necessarily, be added to the function as follows:

$$n = \max_{\mathrm{a}} a \quad if \quad |\max_{\mathrm{a}} a| > \tau; \quad 0 \quad if \quad otherwise$$

where τ is the threshold for a particular node.

[14] The above difference equation can also be expressed as a differential equation:

$$\frac{\partial a}{\partial t} = \sum_i w_i i_i - a$$

As a matter of fact, it is easily provable that at equilibrium, the above equation will reach the same value as an algebraic equation:

$$a = \sum w_i i_i$$

In the following discussion, we will no long make the distinction.

[15] This is trivial, knowing that in CL we can always divide concept nodes into layers according to the links between them (because CL is hierarchical) and then we can structure CD according to this layering so that only feature sets of the concepts that are at he same layer can share feature nodes and thus can overlap with each other. So the problem can be avoided.

Definition 3.8 Inter-Level Connectivity *is the connection pattern in which a node in CL is connected to nodes in CD.*

Definition 3.9 *The* **Separate Top-Down and Bottom-Up Inter-Level Connectivity** *is the inter-level connection pattern in which for every link from a node in CL to a node in CD (top-down) there is a link going from that node in CD to the node in CL, and vice versa.*

Definition 3.10 Inter-Level Dynamics *is the way the inter-level connectivity is utilized, or the way CL and CD interact.*

For example, the two level could be continuously interacting, or they could interact in certain fixed time intervals, or they could interact in variable time intervals, for example, when certain events occur.

Definition 3.11 *The* **Phasic Interaction Inter-Level Dynamics** *is as follows: using the separate top-down and bottom-up connectivity pattern, for each cycle, first the top-down connection is enabled (from CL to CD), then after an interval the connection is disabled and CL and CD work separately, and finally after both parts settle down the bottom-up connection is enabled (from CD to CL)*

Definition 3.12 CONSYDERR *is an architecture composed of a hierarchical CL and a hierarchical CD which is the result of one-to-many mapping from CL, both with the incremental weighted-sum node activation function, with the separate top-down and bottom-up inter-level connectivity and the phasic interaction inter-level dynamics.*

The reasons for the choices we made here for CONSYDERR are all discussed in the previous sections.

Now we can state the main results we will obtain in this section in formal terms:

Theorem 3.1 CONSYDERR *as defined above can solve the similarity matching problem as stated in the previous subsection, when all parameters are set appropriately.*

Theorem 3.2 CONSYDERR *as defined above can solve the rule application problem as stated in the previous subsection, when all parameters are set appropriately.*

Theorem 3.3 CONSYDERR *as defined above can solve the inheritance problem as stated in the previous subsection, when all parameters are set appropriately.*

These theorems state that this architecture can solve a set of problems, when these problems are considered in isolation. In fact, we can prove a stronger result —a single system based on CONSYDERR architecture can solve them all:

Theorem 3.4 CONSYDERR *can solve the similarity matching problem, the rule application problem, and the inheritance problem, with the same set of (appropriately set) parameters.*

In the rest of this section, we will analyze all the three problems and prove the strongest theorem directly.

The above definitions still leave some choices unspecified regarding details of the system. Choices available are in the following dimensions:

1. Activation values — discrete? (the number of possible values? what are these values?) or continuous? (the range of the values?)

2. Thresholding — thresholding (the value of thresholds? thresholding on actual values or absolute values? what are the resulting values of thresholding?), or not?

These choices can be made as follows for obvious reasons:

1. Activation values — we will use continuous activation values (for representing confidence values) between -1 and 1, in which 1 represents full confidence, 0 represents unknown, and -1 represents full negative confidence (we will explain this further in chapter 5).

2. Thresholding — We will not use thresholding for the simple reason that we adopted continuous activation values which represents continuous confidence values.

Let us begin our analysis by looking at similarity measures first. Given the basic desiderata of similarity, we can think of many different measures (cf. Tversky 1977, Posner 1989, and Grossberg 1987), such as

$$f1(|F_A \cap F_B|) - f2(|F_A - F_B|) - f3(|F_B - F_A|)$$

i.e. the contrast model of Tversky (1977).

Or

$$\frac{f1(|F_A \cap F_B|)}{f2(|F_A - F_B|) + f3(|F_B - F_A|)}$$

i.e. the ratio model of Tversky (1977). The idea embodied in the above two models is that what is important is the shared features and the difference in the two feature sets; the size of the shared feature set has a positive effect on the similarity, and the differences have a negative effect. There are other ways to accomplish this, for example,

$$\frac{f1(|F_A \cap F_B|)}{f2(|F_A|) + f3(|F_B|)}$$

which can be viewed as a variant of the second model, emphasizing the *sizes* of the two feature sets compared with the size of the shared feature set.

Yet others include

$$\frac{f1(|F_A \cap F_B|)}{f2(|F_A|) * f3(|F_B|)}$$

or

$$\frac{f1(|F_A \cap F_B|)}{f3(|F_B|)}$$

which are variants of the third model. Many more models can be constructed. However, when we measure them against our previous desiderata, only the last one is acceptable, because it does not involve F_A.

Looking at the matter from a different perspective, considering the implementational issues, we want as simple a formula as possible, not in terms of numbers of parameters or the time complexity of computation, but in terms of ease of implementing it in a connectionist fashion with a set of simple, autonomous, locally connected nodes (or processing elements in that regard). We want (1) all computation to be local, (2) only simple messages to be passed around, and (3) no extra nodes to be added (cf. Feldman 1986, Shastri 1988 for similar points). With these three criteria in mind, again only the last model can be selected (details are omitted).

We now turn our attention to inheritance/cancellation, including all four cases in Section 3 (also listed in Section 4). Assume in the following discussion the original rule strength in CL connecting a concept and its property values are all the same, that is, the maximal value 1, for the sake of simplifying the discussion. [16] Consider the first two cases. When A is activated (but not B), we want C to be activated more strongly than D in CD. Then during the bottom-up process, these

[16] To justify this, it suffices to notice that these values are defaults, not high probability, not the quantifier MOST, or other things of this sort (cf. Besnard 1989 and Brachman 1985).

activation values will be transmitted to the corresponding CL nodes. [17] To make sure that C is activated more strongly than D, lw (weights on the links that diffusely replicate the link in CL) should be somehow inversely related to the size of the feature set of the originating concept, Assume the original link weight is r, the weights for links in CD that replicate (diffusely) the original link is (uniformly) lw. Let A be the source node, and F_A be its feature set. Similarly, let C be the destination node, and F_B be its feature set.

$$lw_{AC} = \frac{r}{f(|F_A|)}$$

for any A and C, where f is a monotone increasing function, linear or otherwise. Similarly, when B is activated (but not A), we want D to be activated more strongly than C in CD. Because the feature set of A is a subset of that of B, the total activation transmitted to C or D should be related to the sizes of the respective feature sets. Otherwise C and D will receive the same amount of activation, and therefore it will become impossible to differentiate the two. Since the total activation is equal to the size of the feature set of the originating concept times the activation transmitted along each individual link, to make sure that D is activated more strongly than C in this case, we must make f slower than linear, so that the total activation transmitted will be related to the size of the feature set of the originating concept.

It is easy to confirm that no matter what bu and td are used, with this lw function, C and D in CD (actually F_C and F_D) will have the right activation in both cases. The details follow: suppose A is activated, and λ is the activation of the feature nodes of A in CD due to top-down activation, and ϵ is the bottom-up weight (the same for both C and D, because they both have only one feature node):

$$\begin{aligned} C &= \epsilon \sum_{F_A} lw_{AC}\lambda \\ &= \epsilon |F_A| \frac{r}{f(|F_A|)}\lambda \end{aligned}$$

$$\begin{aligned} D &= \epsilon \sum_{F_A} lw_{BD}\lambda \\ &= \epsilon |F_A| \frac{r}{f(|F_B|)}\lambda \end{aligned}$$

so C is activated more strongly than D. In the other case, if B is activated (with value λ for its feature nodes),

$$\begin{aligned} D &= \epsilon \sum_{F_B} lw_{BD}\lambda \\ &= \epsilon |F_B| \frac{r}{f(|F_B|)}\lambda \end{aligned}$$

$$\begin{aligned} C &= \epsilon \sum_{F_A} lw_{AC}\lambda \\ &= |\epsilon F_A| \frac{r}{f(|F_A|)}\lambda \end{aligned}$$

[17] As a general assumption, we take the MAX of the corresponding CL values and the bottom-up values, which represents the combination of the results of the conceptual level and the subconceptual level reasoning.

so D is activated more strongly than C.

We are now ready to examine the last two cases, which are more complicated: because F_C is embedded in F_D, it is imperative that we pick the right bu function that takes into account all effects, desirable or undesirable, of sizes of feature sets.

Look at Case Three (Figure 3.11). In order to have C activated more strongly in CL than D, [18] we have to take into account the sizes of the feature sets of C and D (i.e. F_C and F_D) in determining bu. And the bu should be inversely related to the size of the feature sets of the node in CL with which the particular bu is associated. Assume all features of C and D are activated to the same degree, and F_C is embedded in F_D, i.e. C has fewer features than D; thus if we have a uniform bu (equal to some ϵ), we will have incorrect results (D more strongly activated than C), that is,

$$\begin{aligned} C &= \sum_{F_C} bu_C * (activation\ \ of\ \ each\ \ node\ \ in\ \ F_C) \\ &= \sum_{F_C} bu_C \sum_{F_A} \lambda * lw_{AC} \\ &= |F_C| * \epsilon * |F_A| * \lambda * \frac{r_{AC}}{f(|F_A|)} \\ &= |F_C| * \epsilon * \lambda * r_{AC} \end{aligned}$$

provided f is the identity function, where λ is the activation of the feature nodes of A in CD, which are all the same;

$$\begin{aligned} D &= \sum_{F_D - F_C} bu_D(activation\ \ of\ \ each\ \ node\ \ in\ \ F_D - F_C) + \\ &\quad \sum_{F_C} bu_D(activation\ \ of\ \ each\ \ node\ \ in\ \ F_C) \\ &= \sum_{F_D - F_C} bu_D \sum_{F_A} \lambda * lw_{BD} + \sum_{F_C} bu_D \sum_{F_A} \lambda * lw_{AC} \\ &= |F_D - F_C| * \epsilon * |F_A| * \lambda * \frac{r_{BD}}{f(|F_B|)} + |F_C| * \epsilon * |F_A| * \lambda * \frac{r_A C}{f(|F_A|)} \\ &= \epsilon * \lambda * r_{AC} * |F_A|(|F_C| * \frac{1}{|f(F_A|)} + |F_D - F_C| * \frac{1}{f(|F_B|)}) \\ &= \epsilon * \lambda * r_{AC}(|F_C| + |F_A| * |F_D - F_C| * \frac{1}{|F_B|}) \end{aligned}$$

provided that f is the identity function (the same in the following derivation) and $r_{AC} = r_{BD}$. Comparing the two formulas, clearly $D > C$, which is wrong. On the other hand, if we have bu inversely proportional to the size of the feature set of the CL node with which the bu is associated, that is,

$$bu_C = \frac{1}{|F_C|}$$

and

$$bu_D = \frac{1}{|F_D|}$$

then we will wind up having, obtained the same way as before,

$$\begin{aligned} C &= \sum_{F_C} bu_C * \sum_{F_A} lw_{AC} * \lambda \\ &= r_{AC} * \lambda \end{aligned}$$

[18] Let us assume that there is nothing going on in CL, and all activations come bottom-up.

and

$$
\begin{aligned}
D &= \sum_{F_C} bu_D * \sum_{F_A} lw_{AC} * \lambda + \sum_{F_D - F_C} bu_D * \sum_{F_A} lw_{BD} * \lambda \\
&= \lambda * r_{AC} * (\frac{|F_C|}{|F_D|} + \frac{|F_D - F_C|}{|F_D|} \frac{|F_A|}{|F_B|}) \\
&< r_{AC} * \lambda
\end{aligned}
$$

or C > D, which is correct. So what we can do now is simply to have a function, g, that is faster than a constant (with coefficients properly adjusted to guarantee that the asymptotic properties also hold for small values), and to let

$$bu_C = \frac{1}{g(|F_C|)}$$

for all C. Examples of such g include $\sqrt{x}$, $\log x$, or the identity function or other linear functions, etc, with coefficients equal to 1 and constants equal to 0.

In Case Four (Figure 3.12), we want the opposite: D > C. We can perform a similar analysis. In this case, we would rather have little or no influence from the sizes of the feature sets. It is easy to see why: If we assume that $bu = \epsilon$ (i.e. g(x)=$\frac{1}{\epsilon}$), then, derived the same way as before,

$$
\begin{aligned}
C &= \sum_{F_C} bu_C \sum_{F_A} \lambda * lw_{AC} \\
&= |F_C| * \epsilon * |F_A| * \lambda * \frac{r_{AC}}{f(|F_A|)} \\
&= |F_C| * \epsilon * \lambda * r_{AC}
\end{aligned}
$$

and

$$
\begin{aligned}
D &= \sum_{F_D} bu_D \sum_{F_B} \lambda * lw_{BD} \\
&= |F_D| * \epsilon * |F_B| * \lambda * \frac{r_{BD}}{f(|F_B|)} \\
&= \epsilon * \lambda * r_{AC} * |F_B| * |F_D| * \frac{1}{f(|F_B|)} \\
&= \epsilon * \lambda * r_{AC} * |F_D|
\end{aligned}
$$

provided f is the identity function and $r_{AC} = r_{BD}$, where λ is the activation of the feature nodes of A in CD, which are all the same. Comparing the two formulas, clearly $D > C$, which is correct. But if we try to have a linear function or a function faster than linear, we will not get the correct result. For example, let $g(x) = x$, and then we have

$$bu_C = \frac{1}{|F_C|}$$

and

$$bu_D = \frac{1}{|F_D|}$$

We derive C and D the same way as before:

$$
\begin{aligned}
C &= \sum_{F_C} bu_C * \sum_{F_A} lw_{AC} * \lambda \\
&= r_{AC} * \lambda
\end{aligned}
$$

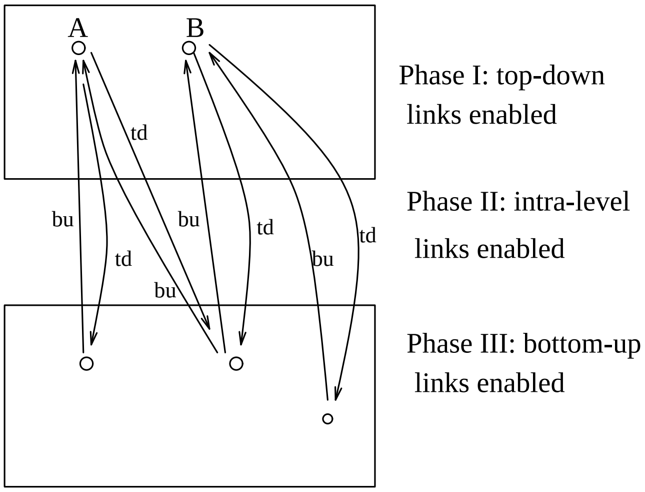


Figure 3.13: Similarity Matching

$$\begin{aligned} D &= \sum_{F_D} bu_D * \sum_{F_B} lw_{BD} * \lambda \\ &= r_{BD}\lambda \end{aligned}$$

or C = D, which is wrong. So for this case we must have a function g that is slower than the identity function (or slower that linear functions in general, but not necessarily a constant). It is easy to verify that we will have $D > C$ with such a g.

Combining results from the above two cases, we conclude that g, as part of bu, has to be a function that is slower than linear, but faster than constants.

Although the above derivation assumes that $f(x) = x$, as we have shown before, $f(x)$ has to be slower than linear. To right the situation, we just have to make $f(x)$ as close to linear as possible, so that the non-linearity of $f(x)$ will not affect the obtained relation ($C > D$ or $D > C$), given the ranges of $|A|, |B|, |C|, |D|$. For example, we can choose $f(x) = x^{999/1000}$ and $g(x) = x^{9/10}$ for our applications (see Chapter 4).

Let us look again into similarity cases. In Figure 3.13, suppose A is externally activated, then because of shared features with A, B will later be activated. The activation of A will go top-down to its feature nodes in CD through weights td. Because of overlapping, some of the feature nodes of B are activated. Then at the bottom-up phase, the activation of the feature nodes of B goes up to the node B in CL through weights bu ($= \frac{1}{g(|F_B|)}$). According to what we derived so far,

$$B = td * A * |F_A \cap F_B| * \frac{1}{g(|F_B|)}$$

To make B match exactly what is obtained from a similarity measure, specifically,

$$s_{AB} = \frac{|F_A \cap F_B|}{|F_B|}$$

we have to choose td as

$$td_A = 1$$

for any node A in CL, and we have to choose g as close to the identity function as possible, to make $\frac{B}{A} = s_{AB}$. Thus, we determine yet another parameter – the weights for top-down transmitting of activation.

Now we shall check to see if the parameters derived so far satisfy the requirements (as listed in Section 4) for correct rule applications and mixed similarity matching and rule applications. Let us verify the test cases identified in Section 4 one by one:

(1) For A $\longrightarrow$ B, if A is activated, then B= $r_{AB} * A$ in CL, where r_{AB} is the weight on the link between A and B (the same below), and the bottom-up activation from CD is $A * |F_B| \frac{|F_A| \frac{r_{AB}}{f(F_A)}}{g(F_B)} \approx r_{AB} * A$, so the overall result is $B \approx r_{AB} * A$;

(2) For A $\sim$ B, B $\longrightarrow$ C, if A is activated, then

$$\begin{aligned} B &= \sum_{F_C} \frac{\sum_{F_B \cap F_A} A \frac{r_{BC}}{g(F_B)}}{g(F_C)} \\ &= \frac{|F_C|}{g(F_C)} A r_{BC} \frac{|F_B \cap F_A|}{g(F_B)} \\ &\approx A * s_{AB} * r_{BC} \end{aligned}$$

(3) For A $\longrightarrow$ B, B $\sim$ C, if A is activated, then

$$\begin{aligned} C &= \sum_{F_B \cap F_C} \frac{\sum_{F_A} A \frac{r_{AB}}{g(F_A)}}{g(F_C)} \\ &= \frac{|F_B \cap F_C|}{g(F_C)} A r_{AB} \frac{|F_A|}{g(F_A)} \\ &\approx A * r_{AB} * s_{BC} \end{aligned}$$

(4) For A $\longrightarrow$ B, B $\longrightarrow$ C, if A is activated, then

$$\begin{aligned} C &= \sum_{F_C} r_{BC} \frac{\sum_{F_A} A \frac{r_{AB}}{f(F_A)}}{g(F_C)} \\ &\approx A * r_{AB} * r_{BC} \end{aligned}$$

(5) For A $\longrightarrow$ B, B $\longrightarrow$ C, C $\longrightarrow$ D, if A is activated, then

$$D \approx A * r_{AB} * r_{BC} * r_{CD}$$

(6) For A $\sim$ B, B $\longrightarrow$ C, C $\longrightarrow$ D, if A is activated, then

$$D \approx A * s_{AB} * r_{BC} * r_{CD}$$

(7) For A $\longrightarrow$ B, B $\sim$ C, C $\longrightarrow$ D, if A is activated, then

$$D \approx A * r_{AB} * s_{BC} * r_{CD}$$

(8) For A $\longrightarrow$ B, B $\longrightarrow$ C, C $\sim$ D,

$$D \approx A * r_{AB} * r_{BC} * s_{CD}$$

(9) For A $\sim$ B, B $\longrightarrow$ C, C $\sim$ D,

$$D \approx A * s_{AB} * r_{BC} * s_{CD}$$

There are some possible extensions to the above scheme. For example, we can allow multiple feature sets for a concept, and consequently take the maximum of all the bottom-up activations from all the feature sets during the bottom-up phase. As another example, we can add a thresholding mechanism to the inter-level interaction, the same as in the intra-level case, so that a CD node (or a CL node) is activated during the top-down (or bottom-up) phase, only if the top-down (or bottom-up) activations received exceed a threshold.

In summary, we derived the structural parameters in the CONSYDERR architecture necessary for solving the problems of rule application, similarity matching, property inheritance and cancellation. Examples of its applications to certain domains will be presented in the next chapter.

3.5 Summary

> *The complexity of philosophy is not in its matter, but in our tangled understanding.*
> — Wittgenstein, *Philosophical Remarks*

This chapter develops a computational architecture that utilizes the analysis in the previous chapter regarding ways to account for the commonsense reasoning patterns exemplified by the Collins Protocols. A two-level dual-representation connectionist architecture is devised based on the dichotomy of the conceptual and the subconceptual levels. The problem of determining justifiably and methodically the structural parameters of the architecture is tackled; in solving this problem, quantitative constraints, requirements, and desiderata are identified to form a foundation based on which we develop a system that is fully capacitated to accomplish the task of accounting for common reasoning patterns.

Specifically, we first define formally the CONSYDERR architecture to facilitate formal analyses. Based on the desiderata associated with similarity matching, we choose a formula for it. Cases of inheritance are analyzed, and we come up with the link weights in the CD level and the bottom-up weights. We then determine the top-down weights in relation to the similarity measures.

A connectionist architecture with a simple two-level structure is shown to be able to solve a quite large range of representation and reasoning problems in an effective and computationally efficient way. Integrating extensional and intensional representation (and thus forming a two-level architecture), enables the system to solve a number of problems that cannot be solved by either representation alone. Intensional representation, and the combination of the two representations, adds much power to symbolic extensional representation and reasoning systems.

3.6 Appendix: What is Needed for an Architecture? (Some High Level Arguments)

To come up with a computational architecture for accounting for common reasoning patterns, we have to determine what is needed in such a system. It should be emphasized that in this process we will try to take into consideration the structure of human cognition, although we may not be able to go very far in this regard. In the first part, arguments will be presented for the inclusion in a system of both conceptual and subconceptual levels, to account for rule application and similarity matching respectively. The second part will examine the interaction of the two levels, and argue for a two-way and phasic interaction between them.

3.6.1 The Two Levels

Concept without intuition is empty. Intuition without concept is blind.
— Kant, *A Critique of Pure Reason*

Arguments will be presented in this part for a (cognitive) reasoning system that is composed of two (or more) levels, each of which represents a qualitatively different cognitive capacity (specifically, two levels are needed that together can deal with rules and similarity in one integrated framework). We will look into issues such as what these levels are, what each should be capable of doing, and how they are cognitively related.

First of all, it is not satisfactory for me to have an ad hoc system composed of a set of randomly built subroutines bearing no resemblance to cognitive architectures. It seems that enough negative lessons have been learned in building AI systems by throwing together components or subroutines whenever a need for a particular computation arises; this tendency is aggravated by the ease with which one can build subroutine in LISP, or any other programming languages for that matter. This practice results in a collection of separate, unrelated subsystems each of which is designed ad hoc for a particular computation. [19]

On the other hand, connectionist models *are* homogeneous models with uniform representation and interaction. However, it seems very unlikely that fully homogeneous models can account for the full spectrum of cognitive capabilities, as pointed out by many leading researchers (Pinker & Prince 1989, Rumelhart & McClelland 1986, etc.). The problems include, among others, the fact that one single homogeneous model does not allow functional specialization, multiple simultaneous computations, or complex interactions of various functionalities. [20]

So, between these two extremes (i.e. ad hoc organizations and fully homogeneous structures), it is more desirable to have a system that has multiple levels, or components, but is designed in a principled way, hopefully bearing close resemblance to cognition. Towards this end, we will have to modify, shape up, or combine existing techniques. Let us start with connectionist models, identifying the main problems with them and looking into ways to amend them.

One particular criticism for connectionism is from Pinker and Prince (1988, 1990) against a connectionist model for past-tense verb learning (Rumelhart and McClelland 1986). They point out that although this model exhibits interesting behaviors and indicates new ways of understanding language and cognition, it leaves much to be desired. One particularly acute problem is the lack of rules that specify exactly the past-tense form of a verb, and the resulting lack of variables, bindings and modularity (which are essential to language capacities according them). This lack results in an

[19] Recall also Marr's objection to programs as theories (Marr 1980).

[20] Hidden units in a multilayer backpropagation network can develop some specialized processing capabilities, but this micro-level specialization is far from sufficient to deal with macro-level functional specializations.

inaccuracy in accounting for children's acquisition of past-tense forms. Specifically, the connectionist model relies on the correlational statistics between verb roots and their corresponding past-tense forms in the input (for conventional connectionist learning algorithms it is essential to rely on statistical correlations), which is not the case in human language acquisition. Therefore, some sort of rule representation and application, and some other high level cognitive capabilities as seen in symbolic systems, are very much needed in a cognitive architecture.

Another argument against homogeneous connectionist models is from the point of view of modeling conceptual and subconceptual processing capabilities. Smolensky (1989) in his famous paper "On the proper treatment of connectionism" fully illustrates the point of view that both conceptual and subconceptual processing need to be taken care of. He points out that the processor for conceptual level processing handles knowledge that possesses the following three characteristics:

- Public Access
- Reliability
- Formality

He reaffirms (as numerous philosophers from Plato to Kant have done) the appropriateness of modeling such conceptual knowledge by symbolic processes, as customarily done in symbolic AI. But contrary to this, skill, intuition, individual knowledge, and so on, are not expressible in linguistic forms and do not conform to the three criteria prescribed above. It seems to be futile to try to model this kind of knowledge in symbolic forms, as argued by Smolensky, for the following reasons:

1. The resulting systems from the symbolic approach are often too brittle and inflexible, even from the symbol processing viewpoint.
2. There are currently major difficulties unsolvable in some important domains, such as natural language understanding, planning, and learning; symbolic systems in these domains are far from comparable to human performance, despite tremendous efforts put into them, and thus the conclusion that symbolic processing is not the appropriate framework seems plausible.
3. Symbolic systems are too far removed from biological implementations in brains and nerve systems and thus they may not be able to capture some of the intrinsic properties of natural intelligence.

Hence, according to the above view, some of the capacities, such as skill, intuition, and individual knowledge, should be viewed as a different kind of capacities, i.e., the subconceptual processing, which may be better dealt with by connectionist approaches.

For connectionist models to be fully qualified as complete models of reasoning, it is not enough to model only subconceptual processes; it is also necessary to address the issues of conceptual level knowledge processing by developing symbolic capabilities in connectionist models (as argued by, for example, Pinker & Prince 1988). Along this line of thought, several pieces of research have been conducted to explore the symbolic processing capability of connectionist models. Among them, Touretzky & Hinton (1985) described a production system based on distributed representation in connectionist networks, with working memory, rule matching and selection, and rule firing mechanisms. Pollack (1989) presented a method of composing and decomposing structures with the back-propagation algorithm, and demonstrated that connectionist models can exhibit some kind of compositionality for symbolic processing. Barnden (1988) described a complex connectionist system wired up in multiple grids for performing syllogistic reasoning. Ajjanagadde & Shastri (1989) described a localist connectionist system for performing backward-chaining rule-based reasoning. Sun

(1989, 1990) and Sun & Waltz (1991) also dealt with modeling rule-based reasoning within connectionist frameworks, performing forward-chaining inference. I find very convincing the viewpoint that connectionism has to address the issue of modeling symbolic processing in a network fashion, in order to fully qualify it to be an alternative model of cognition.

Other arguments for the necessity of rules in connectionist models include Hadley (1990), in which the ease with which one can learn and manipulate rules is explored to show that there ought to be an explicit rule representation that can be consciously manipulated. Yet another argument is offered by Sun (1989 TR), in which conventional connectionist models are identified with sub/unconscious processing, due to the nature of implicit representation in conventional distributed connectionist models, and in which evidence is given for explicit, conscious rule applications in cognition, contrary to the nature of sub/unconsciousness; therefore conclusions are drawn that there should be separate components that can account for explicit conscious rule application in addition to conventional distributed connectionist models.

From a different perspective, recognizing the inherent problems and pitfalls in symbolic AI, Dreyfus (1972, 1987) repeatedly argued that symbolic AI fails to account for intuition, and situation-oriented reasoning or reaction, and that this leads AI to study only deliberative rationality (i.e. analytical knowledge). Deliberative rationality is based on symbol manipulation and therefore is suitable to be modeled by symbolic systems; however reason or rationality has to be rooted in intuition, and otherwise it will not qualify as intelligence. Dreyfus suggests that holistic, "holographic" similarity plays a large role in intuition, and due to its distributed nature, connectionist models might be able to better model intuition that symbolic AI fails to capture. I agree with this view completely.

In light of the above arguments, we can resolve our problem of modeling rules and similarity: I propose that rule applications should be modeled at the conceptual level by a symbolic system, which represents explicit, conceptual knowledge that is consciously accessible; and at the same time fine-grained similarity matching should be modeled at the subconceptual level by an informal, holistic, structureless process, which embodies intuition. So, according to this proposal, on the one hand, we have symbolic systems, deliberative rationality (or reason), conceptual knowledge, and rules; on the other, we have subsymbolic systems, intuition, subconceptual knowledge, and similarity (see Figure 3.14). One added advantage of this duality is, by dividing the reasoning process into the two levels, we are able to explore the synergy between the conceptual level and the subconceptual level, or between concepts and intuitions, as succinctly summarized in the opening quote from Kant. This two level structure and the synergy it generates help to deal with the brittleness problem of symbolic AI. Naturally we will have to consider ways to combine the two radically different aspects of cognition into one integrated system in order to utilize their synergy.

Functional considerations aside, as for the actual anatomy of the brain, there is no definitive evidence [21] why it has to be so. However this proposal seems to be a reasonable one by all accounts, in light of all the arguments given above, and moreover it seems to me to be the best one in sight.

3.6.2 The Interaction

Now the question is how these two levels make contact with one another. We have to decide first how each level should be implemented, taking into consideration the issue of facilitating interactions, and then how the interaction between the two levels should be structured, that is, the physical connections between the two and the timing of and the control over these connections.

The question of what substrate (or form) each level should be implemented in can be reduced to the question of how the conceptual level should be implemented, because the subconceptual level has to be implemented in some kind of connectionist models for lack of other alternatives (based

[21] Considering the state of the art, there is no way we can have *definitive* evidence at present.

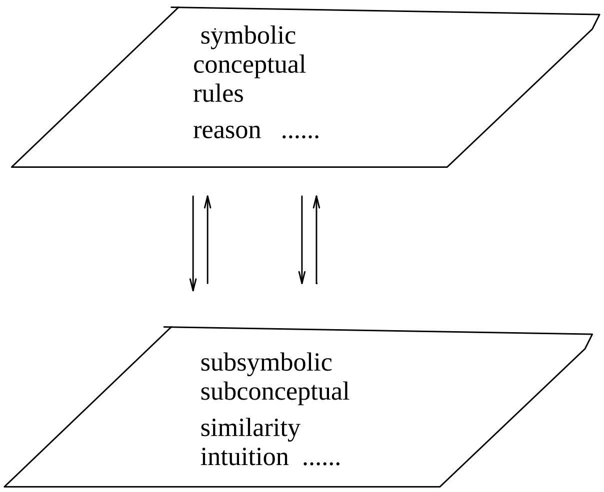


Figure 3.14: The Two Levels

on the foregoing discussion). To determine how to implement rules at the conceptual level, we can take a look at alternatives we have: for examples, there are hybrid systems, which are a mixture of a symbolic component (for example, LISP code) and a connectionist component. The problem with this approach is the difficulty of coupling, i.e., carrying out a rich, mutual interaction between the two totally different components in a massively parallel fashion. This difficulty stems from the inherent oddness of coupling a pure serial symbolic system with a parallel subsymbolic system.

A better approach is to implement both levels as connectionist networks. This way the two levels will have similar representation and comparable structures, and both will work in a massively parallel fashion, which opens the door for rich, complex, and multi-faceted interactions between the two levels, and which also corresponds well with the parallelism in the protocols of human reasoning.

The approach also has the advantage of being more biologically plausible. Any symbolic, conceptual level reasoning ultimately has to be carried out by a massively parallel and "brain-like" structure in which each element does relatively simple processing but in which as a whole, through interactions of simple processing elements, complex, high-level reasoning can take place. Thinking in terms of low level substrates, instead of knowledge level or normative analysis, can help narrow the search for plausible architectures and algorithms, [22] and reduce blind efforts in high level analyses, because of the fact that the science of intelligence is highly complex and underconstrained and that the space of plausible architectures and algorithms is too enormous to perform a brute force search. Low level considerations can play an important role in providing extra constraints with regard to what kinds of processes are possible, so we can produce highly plausible architectures, algorithms, etc. that are directly implementable in a massively parallel fashion. An added advantage is the ease with which such a system can be coupled with low level vision, perception, and motor systems. Therefore, for the sake of facilitating the interactions between the two levels and being more biologically realistic, [23] we will adopt this approach.

The interaction between the conceptual level and the subconceptual level (or feature representation as a simple form of it) is twofold. First of all, each concept at the first level should be connected to its corresponding features in the second level, and there should be both top-down influence — when concept nodes are activated, their feature nodes should also be activated to a certain extent;

[22] By no means is this a denial of the need for work at the knowledge level or otherwise at a high level. What is meant is that we need to look at lower levels too to complement the work at high levels.

[23] It also serves the purpose of further developing the symbolic processing capability of connectionist models, as noted before.

and bottom-up influence —when some feature nodes are activated, the relevant concept nodes should also be affected somehow.

Secondly, the rules established in the conceptual level should be incorporated into the subconceptual level to capture the phenomenon of conceptual, analytical knowledge being incorporated into one's intuition in a vague, inexact but flexible fashion (cf. Dreyfus & Dreyfus 1987 [24]).

The temporal dimension of the interaction is also important. It is postulated here that the interaction between the two levels is not ever-present. To see this, we notice the psychological phenomenon in which intuitions and analytical knowledge are in separate, quite independent processes, though interacting sometimes (cf. Norman 1977 p.590; cf. also Davidson & Davidson 1980 for psychobiological studies of consciousness in general). Taking this into consideration, we will make the interaction *phasic* along the time line, that is, the two levels should interact in certain time intervals in separate bottom-up and top-down phases, as a simplified model of the natural processes. [25]

To conclude, we propose a two-level architecture, explaining the need for the two levels and arguing for connectionist implementations of both levels; the interaction of the two levels mimics in some way the interaction between conceptual knowledge and intuition in human cognition.

---

[24] They propose a five-stage learning model, in which conceptual knowledge is assimilated into intuition through practice. The number of stages in their model may be questionable, but one point stands out: skills are acquired through obtaining conceptual knowledge and gradually turning it into intuition. See also Gelfand et al (1989).

[25] Although there is no evidence that the brain actually does exactly this: we simply do not know enough yet

Chapter 4

Experiments with the Architecture

4.1 Overview

It is costly wisdom that is bought by experience.
— Roger Ascham, *Schoolmaster*

This chapter presents a number of experiments that demonstrate the capability of the CONSYDERR architecture, to supplement, verify and strengthen our earlier theoretical discussions and developments of CONSYDERR.

The plan for the bulk of this chapter is as follows:

- The second section discusses experiments in reproducing the reasoning processes in the Collins protocols.
- The third section contains performance evaluations with systematically generated data.
- The fourth section discusses an important issue involved in accounting for these protocols, namely the issue of choice, focus, and context and their effects on reasoning.
- The fifth section describes a large scale system GIRO for reasoning with geographic information.
- The six section discusses applications of the architecture in a number of other domains, to demonstrate the generality and wide applicability of the architecture.

4.2 Accounting for the Collins Protocols

This section will show how the reasoning embodied in the protocols introduced in Chapter 2 can be accounted for by a system based on the CONSYDERR architecture.

The basic steps taken in accounting for each of the protocols are as follows: first, in each protocol instances of rule applications and similarity matchings are identified, and the ways in which these patterns are intermixed are made out; we describe these instances of rule application and similarity matching in the exact order as they appear in the protocol, with the formalism we developed; finally, the formulated patterns are translated in a straightforward way into representations in the CONSYDERR architecture; in so doing, concepts and features are identified and are put into systems as nodes in the proper levels, and links are added to represent rules, with proper weights set. With inputs given, we run CONSYDERR to obtain output data, which are then compared and verified with the protocol modeled. (See Posner 1989 regarding protocol analyses and modeling.)

The first subsection deals with those protocols involving only rule application in their reasoning processes. The second subsection deals with those protocols that involve both rules and similarity, in which case the interaction between the two levels are crucial. The third subsection is devoted to deal with cases involving inheritance.

4.2.1 The Protocols Involving Rules Only

We will look at the protocols 1, 4, 7, 8, and 9, all of which involve rule application but not similarity matching, and will explain how they can be produced.

CONSYDERR provides all the basic elements we need to account for these protocols: specifically, to account for those rule-based inferences, CONSYDERR provides a way of encoding rules, which "implements" FEL (see Chapter 5 for details), and perform forward-chaining reasoning in a massively parallel fashion. [1]

Let us look at Protocol 1 again:

> Q: Do you think they might grow rice in Florida?
>
> R: Yeah. I guess they could, if there were an adequate fresh water supply, certainly a nice, big, warm, flat area.

In this protocol an uncertain conclusion is deduced based on partial information. The rule used is:

> **if** an area is big, warm, flat and with fresh water supply, **then** it is a rice-growing area.

This can be handled by CONSYDERR: Each node in CL represents a concept, including among others things "big-area", "warm-area", "flat-area", "fresh-water-supply", and "rice-growing-area". The rules are represented by links between nodes:

$$\textit{big-area warm-area flat-area fresh-water-supply} \longrightarrow \textit{rice-growing-area}$$

The weights on the links reflect degrees of confidence in the respective implications (or causal connections), as well as positiveness/negativeness of the implications. In this case, the nodes representing the four conditions link to the node representing "rice-growing area" at the *same* site, because they are part of the same rule, and weights for these links sum to be no more than one (see Chapter 5 for detailed reasons). The reasoning process is as follows: First three out of the four conditions are activated to certain degrees (which reflect the corresponding confidence in these facts), then they send their activation to the node representing the conclusion ("rice-growing-area") and activate that node. Because of one missing condition, the activation, calculated with weighted-sum, will be less than one, but still larger than zero. Therefore we conclude that it *might* be a rice-growing area. Figure 4.1 depicts the whole reasoning process.

In the CD level, all the features necessary for this domain are present and are connected to the proper CL nodes. However, in this case, and in the following cases involving rule application only, the CD level is not necessary for reaching the desired conclusions and it can be safely discounted in all regards (we will not dwell on the details of this). Therefore, details in CD is not shown in the above picture.

Now let us look at Protocol 4.

[1] Because of this massively parallel forward-chaining approach, the problems of selecting from moment to moment a particular path to pursue and performing backtracking when reaching deadends are no longer existent. For more information see Chapter 6. Since this is not a study of reasoning in a particular domain, but a study of domain independent patterns in commonsense reasoning, we will not concern ourselves too much with domain details.

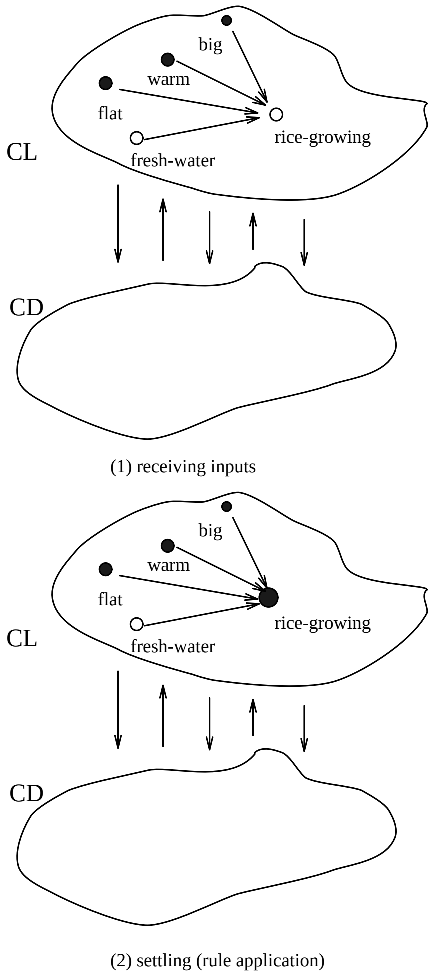


Figure 4.1: The Reasoning Process for Protocol 1

> Q: Is that [Llanos] where they grow coffee up there?
>
> A: I don't think the savanna is used for growing coffee. The trouble is the savanna has a rainy season and you can't count on rain in general [for growing coffee].

This protocol clearly shows a chain of reasoning by applying three rules in sequence: Llanos is a savanna, savanna has a rainy season, and rainy seasons prohibit coffee growing. This can be handled by CONSYDERR: Each node in CL represents a concept, including "Llanos", "savanna", "rainy-season", "coffee-growing". The rules are represented by links between nodes:

$$Llanos \longrightarrow savanna$$

$$savanna \longrightarrow rainy\text{-}season$$

and

$$rainy\text{-}season \longrightarrow \neg coffee\text{-}growing$$

etc. The weights on the links reflect degrees of confidence in the respective implications. The link between "rainy-season" and "coffee-growing" is very negative. The reasoning process is as follows: First the node for Llanos is activated (to the maximum extent 1). Then it passes on activation to "savanna", then to "rainy-season", and finally to "coffee-growing", all through the weighted-sum computation. The weight on the link from "rainy-season" to "coffee-growing" is negative, so the node representing coffee-growing will have a negative activation value when the activation is passed to it. So the conclusion is that there may not be any coffee growing in Llanos. Figure 4.2 depicts the whole reasoning process.

Now turn to Protocol 7.

> Q: Is Florida moist?
>
> A: The temperature is high there, so the water holding capacity of the air is high too. I think Florida is moist.

Here there is a rule:

> **if** the temperature is high, **then** the water holding capacity of the air is high;

and a fact:

> the temperature is high in Florida.

So the conclusion is the water holding capacity of the air is high (i.e. moist).

Again, nodes representing propositions like "temperature-high" and "water-holding-capacity-high" are linked based on the rule that relates them. The weights on the link represent the confidence we have on the implication. The reasoning is carried out by first activating the node representing "temperature-high" and then the subsequent activation of the node representing "water-holding-capacity-high" caused by the input from the first node along the link. So the conclusion is that the water holding capacity is high.

Now let us examine Protocol 8:

> Q: Will high interest rates cause high inflation rates?
>
> A: No. High interest rates will cause low money supply growth, which in turn causes low inflation rates.

This protocol again shows the chaining of two rules:

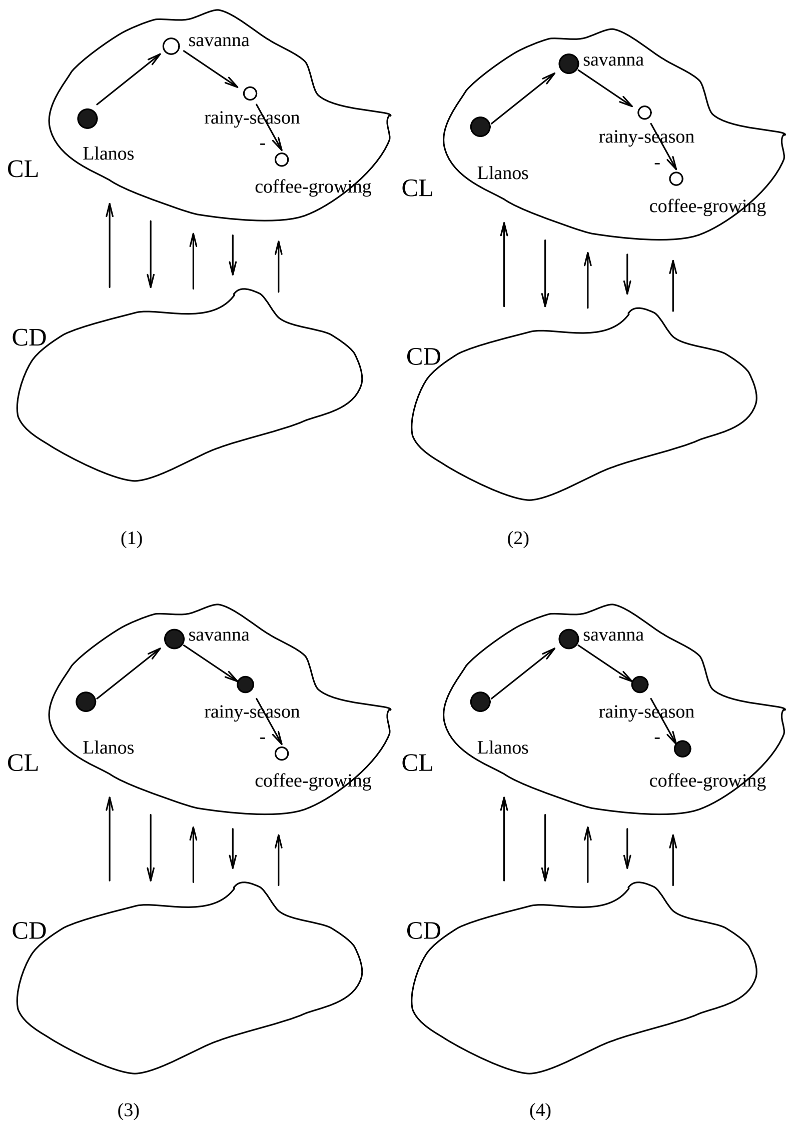


Figure 4.2: The Reasoning Process for Protocol 4
(The propagation of activation in CL.)

High interest rates will cause low money supply growth

Low money supply growth will cause low inflation rates

So, in conclusion, high interest rates will cause low inflation rates.

The nodes in CL represent "high-interest-rate", "low-money-supply-growth", and "low-inflation-rate", etc. There are links from "high-interest-rate" to "low-money-supply-growth", and from "low-money-supply-growth" to "low-inflation-rate". After "high-interest-rate" is activated, the activation will propagate to "low-inflation-rate" via "low-money-supply-growth". So the conclusion of "low-inflation-rate" is obtained.

Finally, let us look at Protocol 9.

Q: What kind of vehicles are you going to buy?

A: For carrying cargo, I have to buy a utility vehicle, but for carrying passenger, I have to buy a passenger vehicle. So I will buy a vehicle that is both a utility and a passenger vehicle. For example, a van.

This protocol uses two rules:

if carrying cargo, **then** buy a utility vehicle,

and

if carrying passengers, **then** buy a passenger vehicle.

Both rules are activated in this case, because the inputs are "carrying cargo" and "carrying passenger". So the results are both "buying a utility vehicle" and "buying a passenger vehicle". As will be explained later, the results from the two rules interact and the outcome is the "sum" of the two conclusions.

In CL, there are nodes representing "carry-cargo", "carry-passenger", "buy-utility-vehicle", and "buy-passenger-vehicle", and proper links representing rules between them. In CD, there are nodes representing all relevant features, which allow interactions between representations of various concepts. The issue of interaction will be analyzed in Chapter 6.

In the above protocols, the networks based on CONSYDERR contain a large number of relevant and irrelevant nodes and links. How do we find the right links and nodes for certain reasoning in this myriad? The answer is that the system is massively parallel and reasoning is spontaneous, so there is no need for searching a vast space to find a right answer. The right answers will pop out automatically, along with other information. As evident from the discussion and derivation in the previous chapters. when there are contradictory information around, the correct answer is certain to stand out amidst other related or unrelated answers, Because of the *massive parallelism* in the architecture, we can perform simple forward-chaining reasoning [2] efficiently, without the need for backtracking and without the need for the somewhat awkward backward-chaining as some other reasoning systems do (cf. Hayes-Roth et al 1985 and Waterman 1987). Overall, we can see that CONSYDERR is well-suited for the task of accounting for the protocols involving rule-based reasoning (see also Barnden 1988, Kosko 1988 and Blelloch 1988 for similar systems; the differences will be discussed in Chapter 8).

[2] Goal components can be added to rules, so that reasoning will be more "goal-directed". See Sun & Waltz 1991 for details. Other gadgets of this sort are also possible.

4.2.2 The Protocols Involving Rules and Similarity

We will look at the protocols 2, 6 and 10, and explain how they can be produced by the CONSYDERR architecture. These protocols involve both rule application and similarity matching. In this type of reasoning, it is crucial to have a subconceptual level with feature-based representation. When conceptual knowledge is not enough for deducing a needed conclusion, similarity in the subconceptual level is explored and a plausible conclusion is reached through the interaction between the two levels.

Protocol 2 is as follows:

> Q: Is the Chaco the cattle country?
>
> R: It is like western Texas, so in some sense I guess it's cattle country.

Here, because there is no direct knowledge regarding Chaco, an uncertain conclusion is drawn based on similarity with known knowledge. The knowledge is expressed in a rule:

$$Western\text{-}Texas \longrightarrow cattle\text{-}country$$

represented in CL by the two nodes, one for Western-Texas and the other for cattle-country, and the link between the two nodes. The similarity between the two areas

$$Chaco \sim Western\text{-}Texas$$

is implemented through feature overlapping in CD. There are many different features, including those shared by Western-Texas and Chaco such as temperate, grassland, plain, etc. And the CL links are diffusely replicated in CD the way we described before. So the reasoning is as follows: First the node for Chaco is activated; In the top-down phase the CD representation of Chaco will be activated and because of shared features, the CD representation of Western-Texas is activated partially to a degree proportional to the similarity measure; Then in the settling phase, the links representing rules take effect in CD, so the CD representation of cattle-country is partially activated; Finally in the bottom-up phase, the partially activated CD representation of cattle-country will percolate up to activate the node representing cattle-country in CL. The result, cattle-country, can be read off from CL. See Figure 4.3.

Now let us look at Protocol 6.

> Q: Can a goose quack?
>
> A: No. A goose – well, it's like a duck, but it's not a duck. It can honk, but to say it can quack. No. I think its vocal cords are built differently. They have a beak and everything. But no, it can't quack.

This example shows two patterns. One is based on the similarity between goose and duck, and it can be described by the following:

$$goose \sim duck$$

$$duck \longrightarrow quack$$

All concepts, goose, duck and quack, are represented by nodes in CL, and are connected to their corresponding features in CD. Therefore, when "goose" is activated, the activation will be passed to "quack" through similarity with "duck", the same way as before. So we conclude (wrongly, as in the protocol) that a goose can quack.

Now let us look at Protocol 10.

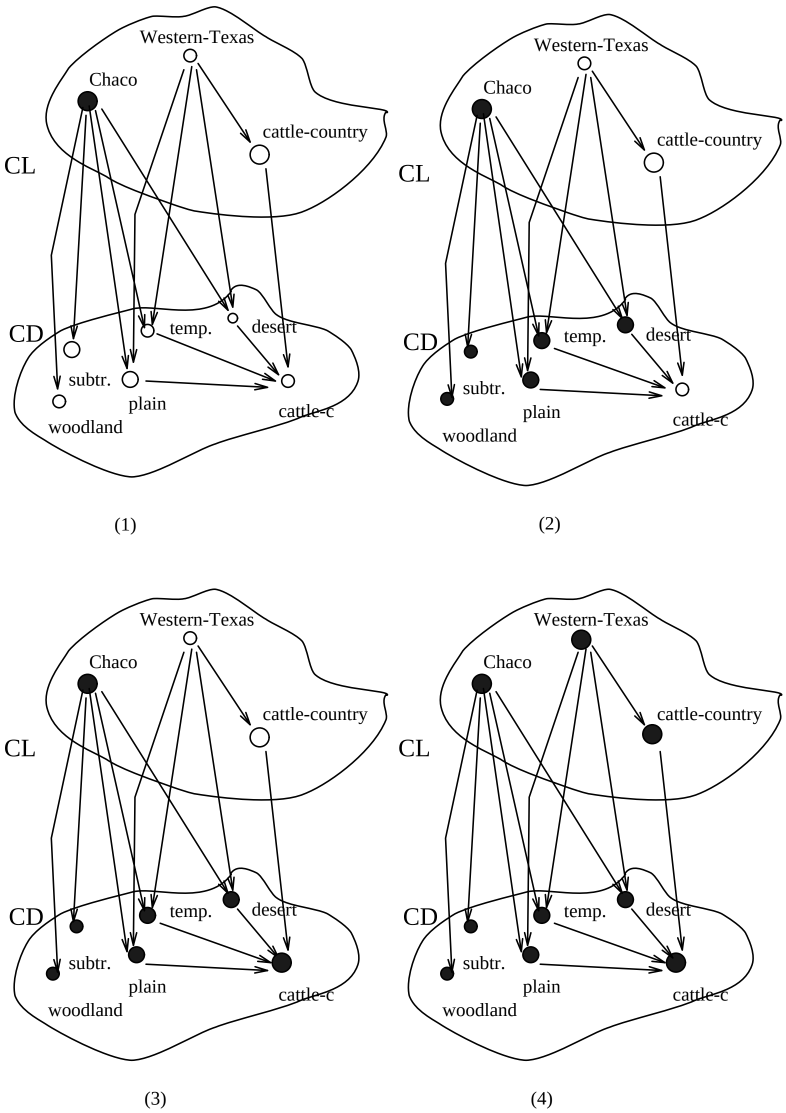


Figure 4.3: The Reasoning Process for Protocol 2
(1) Receiving inputs, (2) Top-down, (3) Settling (rule application), and (4) Bottom-up.

Q: Do women living in that region have short life expectancy?

A: Men living in tropical region have short life expectancy, so probably women living in tropical region have short life expectancy too.

This protocol is another case of using similarity, due to the lack of direct knowledge. It can be described by

$$\textit{women-in-tropical} \sim \textit{men-in-tropical}$$

$$\textit{men-in-tropical} \longrightarrow \textit{short-life}$$

Each concept involved is represented by a node in CL, and a number of (shared) features in CD. Therefore, when "women-in-tropical" is activated, "short-life" will be activated partially, through rule links and similarity matching carried out in CD.

Overall, the system handles mixed rule application and similarity matching well, through interaction and cooperation between the two levels in CONSYDERR (cf. Riesbeck & Schank 1989 for similar systems in case-based reasoning; see Chapter 8 for detailed comparisons).

4.2.3 The Protocols Involving Inheritance

We will look at the protocols 3, 5 and 11. These protocols involve inheritance, which is actually a combination of both rule application and similarity matching but is dealt with separately because it imposes some additional requirements.

Protocol 3 is as follows:

Q: Are there roses in England?

R: There are a lot of flowers in England. So I guess there are roses.

The reasoning based on property inheritance is as follows:

England HORTICULTURE [3] flower; rose IS-A flower; so England HORTICULTURE rose.

It can be described by [4]

$$England \longrightarrow \textit{flower}$$

$$flower \sim rose$$

Therefore, the same way as before, this can be implemented in CONSYDERR with the two-level dual representation and their interaction. See Figure 4.4.

Protocol 5 is as follows:

Q: Is Uruguay in the Andes Mountains?

A: It's a good guess to say that it's in the Andes Mountains because a lot of the countries (of South America) are.

[3] HORTICULTURE here is a property, but can be viewed as a link like IS-A. See Chapter 6 for detailed explanations. The same goes for other properties.

[4] Although not shown explicitly in the following formalism, in this case we actually have *flower* $\supset$ *rose* and $F_{flower} \subset F_{rose}$.

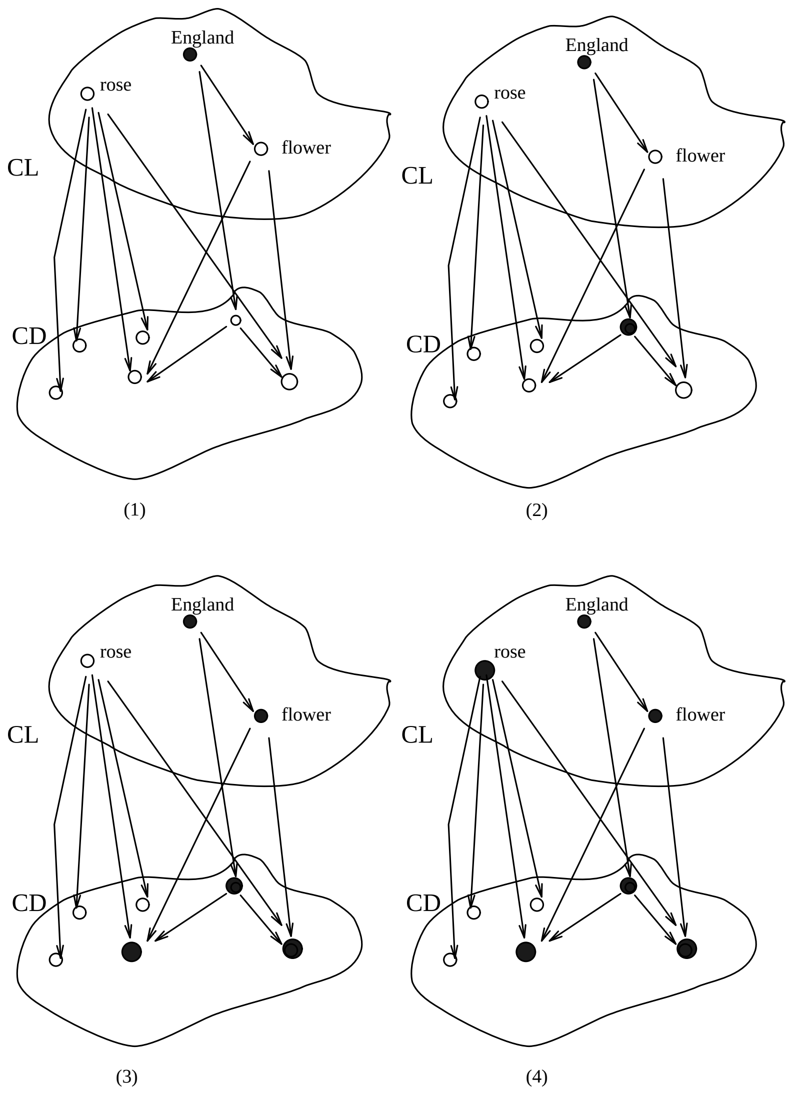


Figure 4.4: The Reasoning Process for Protocol 3
(1) Receiving inputs, (2) Top-down, (3) Settling (rule application), and (4) Bottom-up.

Formally, according to the inheritance theory, we have: South-American MOUNTAIN Andes; Uruguay IS-A South-American; so Uruguay MOUNTAIN Andes. Describing it in rules and similarity: [5]

$$Uruguay \sim \textit{South-American}$$

$$\textit{South-American} \longrightarrow \textit{in-Andes}$$

Therefore, the same way as before, this can be implemented in CONSYDERR through the interaction between the two levels of representation.

Protocol 11 is as follows:

> Q: Are all South American countries in the tropical region?
>
> A: I think South American countries are in the tropical region, because Brazil is in the tropical region, Guyana is in the tropical region, Venezuela is in the tropical region, so on and so forth.

This protocol illustrates bottom-up inheritance (a form of generalization). According to the inheritance theory, Guyana IS-A South-American; Guyana WEATHER tropical; Brazil IS-A South-American; Brazil WEATHER tropical; Venezuela IS-A South-American; Venezuela WEATHER tropical; so South-American WEATHER tropical. Describing it with rules and similarity:

$$\textit{South-American} \sim Brazil$$

$$Brazil \longrightarrow \textit{tropical}$$

$$\textit{South-American} \sim Guyana$$

$$Guyana \longrightarrow \textit{tropical}$$

$$\textit{South-American} \sim Venezuela$$

$$Venezuela \longrightarrow \textit{tropical}$$

Therefore, the same way as before, this can be implemented in CONSYDERR with the two-level dual representation and interaction.

Overall, CONSYDERR deals successfully with this type of reasoning (cf. Touretzky 1986 for discussions of inheritance; see Chapter 6 for detailed treatments of inheritance in CONSYDERR).

4.3 Systematic Performance Evaluations

We derived the system parameters and functions for CONSYDERR in Chapter 3 to make sure some important properties hold, and we applied the model to solve the problem of accounting for the Collins Protocols to demonstrate its utility. However, we do not yet know how the system and its components fare when the model is applied to a wider range of situations. In other words, we have not studied the whole spectrum of its behavior, and thus do not have a complete and concrete picture of various behaviors it can exhibit. Therefore it is both useful and necessary to explore the characteristics of the model systematically, with experiments. Specifically, we want to design test cases that can cover tests of rule application, similarity matching functions, top-down weights, bottom-up weights, CD link weights, etc., to see how they work under different circumstances.

In the following discussions, test cases and test results are presented in four different categories: similarity matching, rule applications, and inheritance.

[5] In this case we have *South-American* $\supset$ *Uruguay* and $F_{South-American} \subset F_{Uruguay}$.

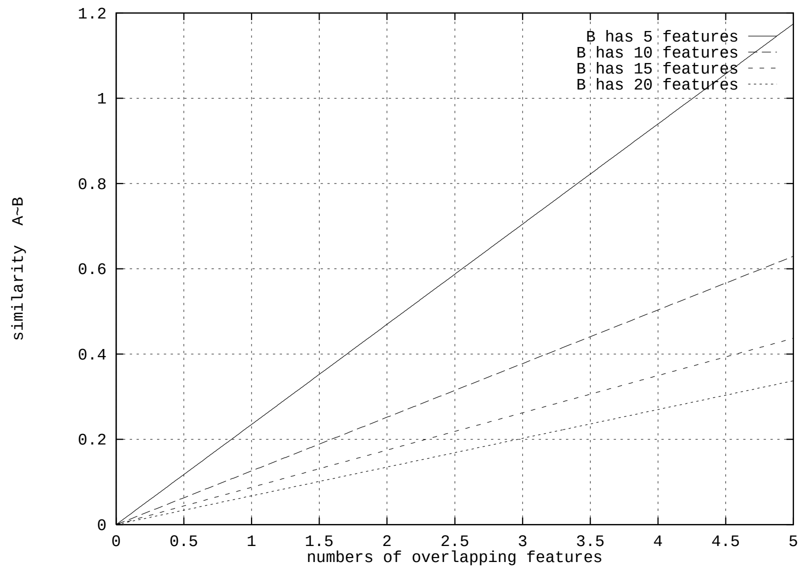


Figure 4.5: Similarity Matching in CONSYDERR
when A has 20 features and B has small variable numbers of features, where $S_{AB} = \frac{|CD_A \cap CD_B|}{|CD_B|^{\frac{9}{10}}}$

4.3.1 Similarity Matching

Let us look into the cases of similarity matching. Similarity matching is calculated with

$$s_{AB} = \frac{|F_A \cap F_B|}{g(|F_B|)}$$

This formula is implemented with the two level structure of CONSYDERR.

To test the similarity in CONSYDERR, we collected the data in the following two figures: Figure 4.5 and Figure 4.6. The two figures demonstrate the difference in the final similarity outcome when one or more of the following three parameters change: $|CD_A|$, i.e. the number of features for A, $|CD_B|$, i.e. the number of features for B, and $|CD_A \cap CD_B|$, i.e. the number of overlapping features. The results show that each curve is almost straight while in fact a slower than linear function is employed, as explained in Chapter 3. When the number of $|CD_B|$ grows big, the addition of features in B has less effect on the similarity outcome. The number of $|CD_A|$ has basically no effect on the similarity outcome.

4.3.2 Rule Application

Let us look into the cases of rule application. In CONSYDERR activation resulting from rule application are determined by

$$B = \sum W_i A_i(t)$$

at equilibrium. The actual calculation is

$$\Delta B = \alpha \sum W_i A_i(t) - \beta B(t)$$

And the same holds for CD:

$$\Delta x = \mu \sum w_i x_i(t) - \nu x(t)$$

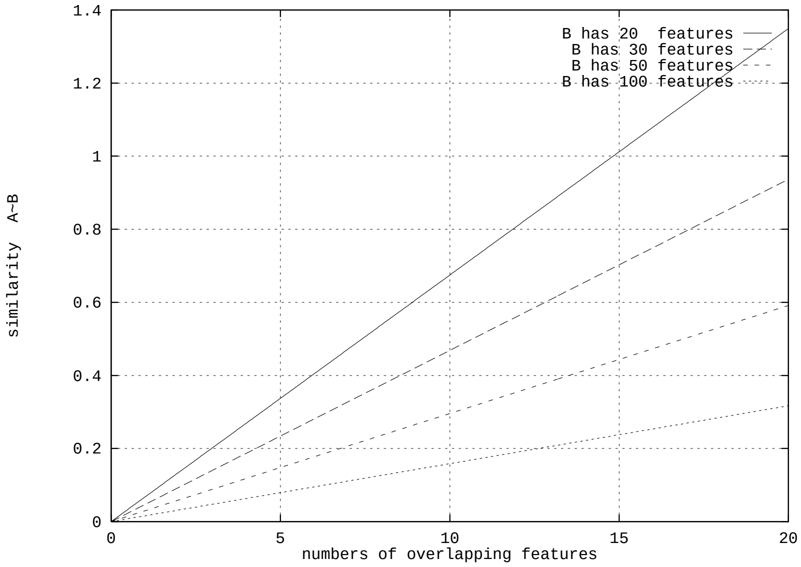


Figure 4.6: Similarity Matching in CONSYDERR
when A has 20 features and B has large variable numbers of features, where $S_{AB} = \frac{|CD_A \cap CD_B|}{|CD_B|^{\frac{9}{10}}}$

where W_i, w_i are link weights (representing strength or weights in FEL rules), and $\alpha = \beta = \mu = \nu = 1$. These formulas are implemented at the two levels of CONSYDERR.

To test this in CONSYDERR, we collected activation data as in Figure 4.7. The figure shows the different activations of the conclusion with different number of conditions activated (and to different degrees), in a fixed order but with different weight distributions.

4.3.3 Inheritance

Let us look into the cases of inheritance. Inheritance is carried out in CONSYDERR with mixed similarity matching and rule application: that is, classes are represented as nodes and feature sets, properties are represented as separate nodes and feature sets, and links between them indicates which class has what properties. These are implemented with the two level structure of CONSYDERR.

To test inheritance in CONSYDERR, we can examine the following figures: Figure 4.8, Figure 4.9, Figure 4.10, and Figure 4.11. These figures are for bottom-up inheritance, top-down inheritance, bottom-up inheritance with cancellation, and top-down inheritance with cancellation, respectively. In these figure, we assume that $f(x) = x^{\frac{9}{10}}$ (f(x) is used in bu as before). We only look at cases where each property value is represented by a distinct single node.

4.4 Choice, Focus and Context

For many are called, but few are chosen.
— Matthew XXII

A question that naturally comes to mind when looking at all the previous experiments is: How do we select the right features to focus on? because obviously similarity is not context-free. A related question is: After we reached a number of conclusions about a situation, how do we sort them out, i.e., picking up the most plausible ones and discarding contradictory ones?

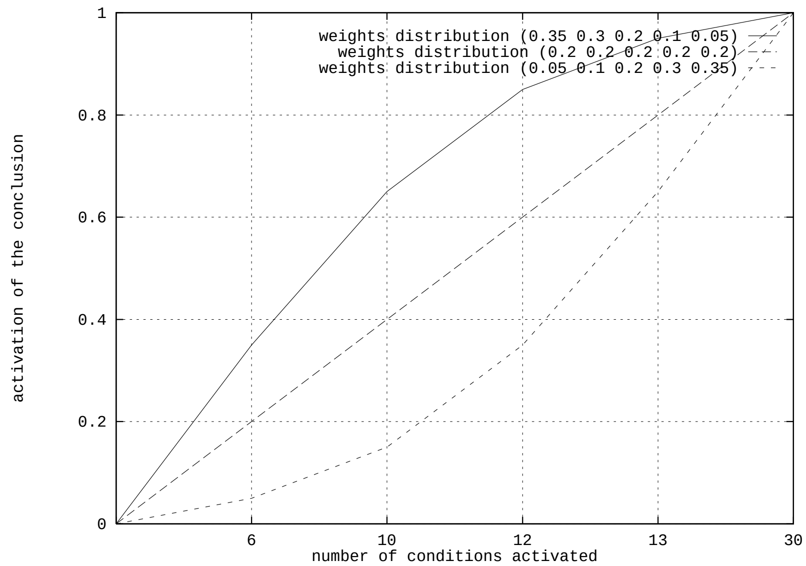


Figure 4.7: Rule Application in CONSYDERR

Activation of the conclusion relative to the number of conditions activated, with different weight distribution.

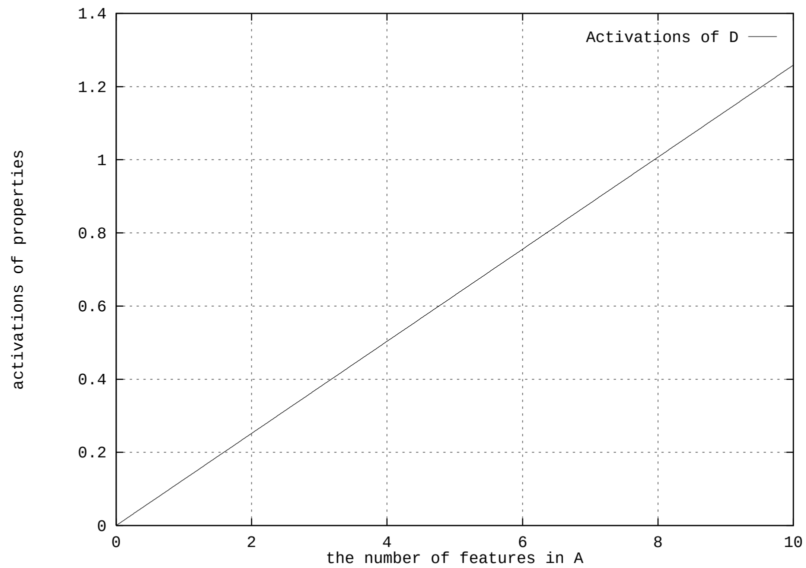


Figure 4.8: Inheritance in CONSYDERR

Activations of properties relative to the number of features in A, when A is fully activated, where $A \supset B$, $CD_A \subset CD_B$, $|CD_B| = 10$, and $B \longrightarrow D$ with the weight equal to 1.

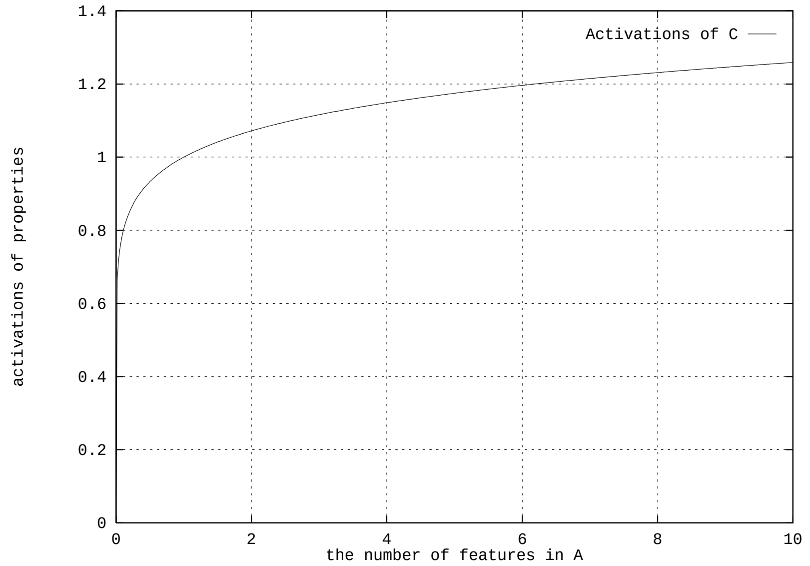


Figure 4.9: Inheritance in CONSYDERR
Activations of properties relative to the number of features in A, when B is fully activated, where $A \supset B$, $CD_A \subset CD_B$, $|CD_B| = 10$, and $A \longrightarrow C$, with its weight equal to 1.

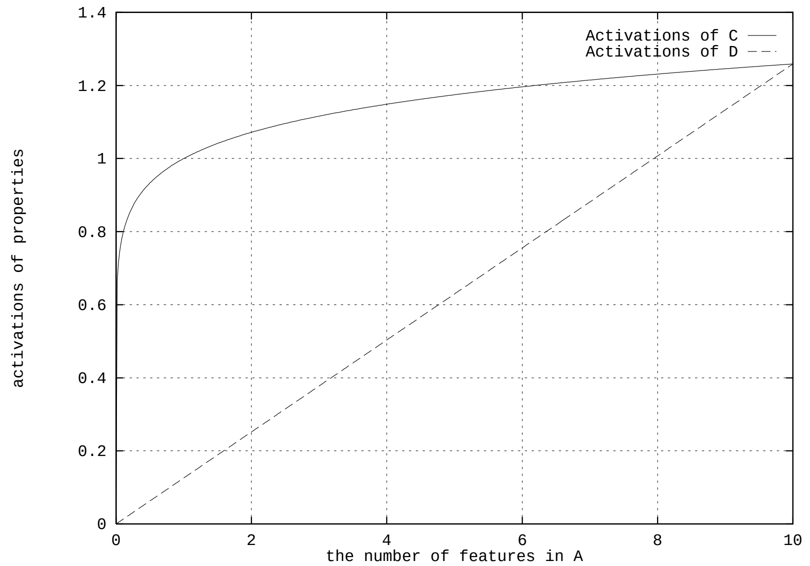


Figure 4.10: Inheritance in CONSYDERR
Activations of properties relative to the number of features in A, when A is fully activated, where $A \supset B$, $CD_A \subset CD_B$, $|CD_B| = 10$, and $A \longrightarrow C$, $B \longrightarrow D$, with the weights equal to 1.

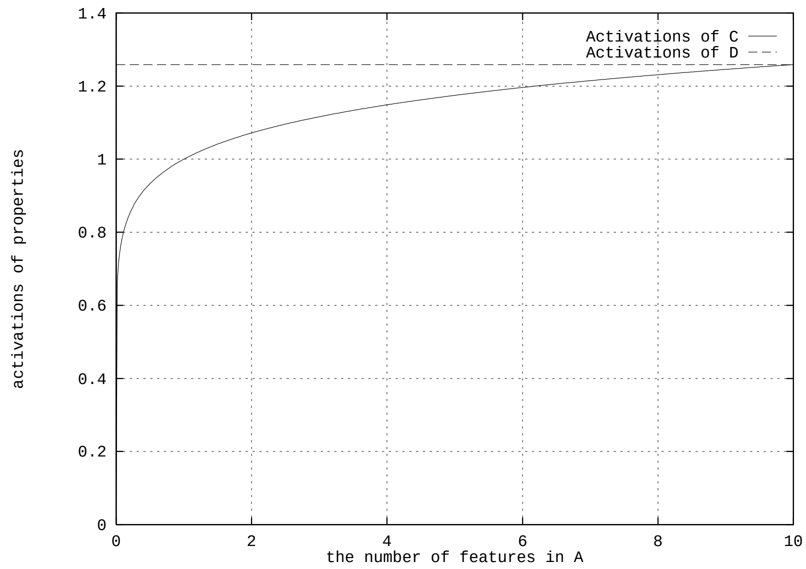


Figure 4.11: Inheritance in CONSYDERR
Activations of properties relative to the number of features in A, when B is fully activated, where $A \supset B$, $CD_A \subset CD_B$, $|CD_B| = 10$, and $A \longrightarrow C$, $B \longrightarrow D$, with the weights equal to 1.

These problems are, more or less, ubiquitous, that is, most of AI systems encounter these problems one way or the other. Dreyfus (1972) considers the inability of symbolic AI in dealing with the issues of finding similarity relative to contexts as a major obstacle towards real intelligent systems functioning in the real world. Dennett (1983) discuss a complementary issue — what he terms the "smoking gun" problem — how one can decide which of the features in a particular situation can be safely ignored. Clearly I do not have a generic solution to these problems; what I propose in this section is a specific mechanism for a specific task. In a way it is more to circumvent the problems than to solve them.

The problem of **choice** is how to choose the right answer out of many contradictory or complimentary results reached by a system, such as a connectionist network. Because of the nature of the forward-chaining reasoning as in connectionist networks like CONSYDERR, all reachable results, direct or indirect, will be reached within a fixed amount of time, some of which may be weak in activation though. Given the large number of conclusions, a system is faced with the task of sorting out useful information from this mess. One way to do this is by forming *subject sets*, each of which is a set of concepts (represented in the system by nodes) that are relevant in a given context and with regard to a particular goal. The relationships among those concepts in a particular subject set include: *contradiction* (for example, color-gray vs. color-white), *complementation* (for example, color-gray and size-large), and *no-relation* (for example, color-white and music-classic). In order to deal with these different relationships, we need to further divide a subject set into *core subject sets*, each of which contains only those concepts that are contradictory to each other. Then after reaching conclusions in a core subject set, the results can be passed on to a *Winner-Take-All* network (a separate network), which uses competition to determine the most plausible conclusion. We then examine all WTA networks for all core subject sets to gather final results.

Subject sets and core subject sets are formed dynamically, after the current context is given (or inferred). With different contexts, those sets are different – it is the context that determines the formation and the structure of those sets.

The problem of **focus** is, in similarity matching, how to attend to those features that are relevant

context	context rules
agriculture	rice-growing $\longrightarrow$ tropical subtropical temperate ...... plain highland ....
mining	mineral-deposit $\longrightarrow$ plain highland coastal rugged ....

Figure 4.12: Examples of Context Rules

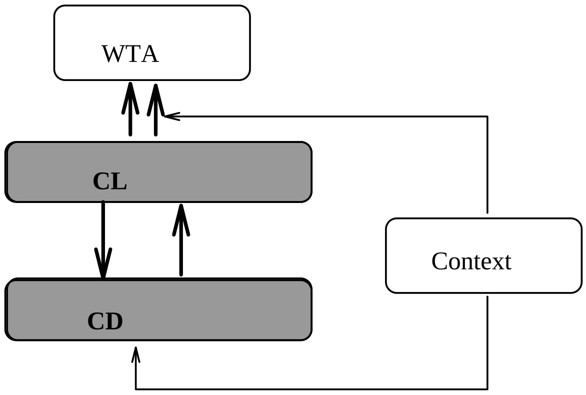


Figure 4.13: The Overall Architecture with Feature and Result Selections

in the current context with regard to current goals. We utilize contexts through "context rules", which, given a particular context (as inferred from a query as in the above protocols and as in GIRO to be detailed later), will highlight those features that are relevant to the current context and will suppress others. For example, if a query is "Is Florida a rice-growing area?", only those features that are related to agricultural products should be used, that is, climate, landform, vegetation, etc. If a query is "What is the most important mineral deposit in Texas?", features like landform but not climate will be highlighted. See Figure 4.12. Similarly, given current context information, core subject sets as explained above can be formed in the same way.

Incorporating these mechanisms, the overall system architecture for accounting for the Collins Protocols is shown in Figure 4.13. Because of the fact that these mechanisms are domain specific, they are not part of the CONSYDERR architecture, but add-on mechanisms.

4.5 Reasoning with Geographical Knowledge

In this section, we will look into a system for reasoning with geographical information: GIRO, which stands for *a system for Geographical Information Reasoning and Organization.*

In our previous discussion of how CONSYDERR accounting for the Collins Protocols, the question of sizes and scalability arises naturally: What if there are a large number of nodes in each level? What if there are a lot of similar concepts, rules and features? In order to show that systems for reasoning in a particular domain can be constructed systematically based on the CONSYDERR architecture and that it will work reasonably despite the existence of interference, distractions, and a large number of nodes, we construct and study the system GIRO, which stores large amount of knowledge extracted from encyclopedias and reasons about agricultural characterizations of regions.

The first subsection will describe the structure of GIRO: how knowledge is organized into the two level architecture of CONSYDERR. The second subsection will examine the issue of knowledge acquisition for GIRO: how knowledge is extracted from sourcebooks and put into the system. The third subsection will show how the system works by presenting several detailed examples.

4.5.1 The Structure of GIRO

GIRO is built based on CONSYDERR, a two-level, dual representation connectionist architecture, with structural parameters as specified in Chapter 3 for satisfying the constraints of the problem it purports to solve.

The knowledge representation in GIRO utilizes the two level idea in the CONSYDERR architecture by dividing the geographical knowledge represented in the system into two categories: concepts, which include basic geographic areas and regional characterizations (such as "cattle-country"), and features, which include primitive geographical descriptions of areas, such as "highland", "mountainous", and "tropical", etc. Concepts are represented in CL, and features are represented in CD. Each geographic area represented in CL is connected to its corresponding features in the CD level, and because of the fact that features are shared by similar concepts, the CD representation is similarity-based, i.e., two concepts have overlapping CD representations if and only if the two are similar and the amount of overlapping is proportional to the degree of the similarity between them, as alluded to before. Each area is also connected to concepts describing its agricultural products by links, if this knowledge is available to the system.

To give a clear idea of what is available in the GIRO system, we tabulate some of the data stored in the system. Figure 4.14 and Figure 4.15 list some of the geographic areas included in the system. Most of them are in the South America, which is chosen as an example. Figure 4.16 lists concepts for characterizing a geographical area in terms of its agricultural products, such as rice-growing-area, cattle-country, etc. Figure 4.17 lists features used in CD.

4.5.2 The Knowledge Acquisition for GIRO

Having described the structure of GIRO, we turn to the question of how to acquire all the necessary data to build up such a system.

The problem of knowledge acquisition is ubiquitous. Much research in AI has been concentrated in this area. Although this work does not contribute any general technique for knowledge acquisition to this field of research, it should be stressed that the process of knowledge acquisition for GIRO is straightforward and systematic, and nothing is tuned arbitrarily just for getting one outcome or the other.

Specifically, the knowledge in GIRO is obtained from encyclopedia, such as *Encyclopedia Britanica* or *Encyclopedia Americana*, in the form of a basic geographic region (a region with relatively uniform characteristics), its products, and its geographical features. Since those types of information is well documented and rather extensive in source books, the task is straightforward, although it is extremely tedious and time-consuming.

For example, a typical entry is like this:

```
Peru--Montana:
lowland, tropical, woodland
```

This is extracted from the following descriptions:

> Peru–Montana:
>
> From west to east the major surface features of Peru are 1) the desert coast, 2) the Andes highlands, 3) the Montana.
>
> **Montana**: The Montana is the remote region containing the lower, densely forested slopes of the eastern Andes and the flat, tropical Amazon plains called Selva. The region experience high temperatures and abundant rains throughout the year. The natural vegetation cover is a tall tropical rainforest, with a leafy canopy shading a damp, open

———————————————————————

"Chaco"
"Honduras"
"Uruguay-coastal"
"Uruguay-plateau-highland"
"Mendoza"
"Lanos"
"w-peru"
"c-peru"
"e-peru"
"Bolivia-orient-grassland"
"Bolivia-orient-rainforest"
"Bolivia-cordillera-occidental"
"e-paraguay"
"w-paraguay-forest"
"w-paraguay-savanna"
"Panama-lowland"
"w-Texas"
"guiana-pgs"
"guiana-hilly-country-forest"
"guiana-hilly-country-savanna"
"guiana-plain"
"Bolivia-SW-highlands"
"Brazil-cw"
"Brazil-s"
"Brazil-saopaulo"
"Brazil-e"
"Brazil-ne"
"Brazil-n"

———————————————————————

Figure 4.14: Geographical Areas Included in the GIRO System

"chile-n"
"chile-s"
"chile-c"
"Argentina-ne"
"Argentina-pampa"
"Argentina-patagonia"
"Argentina-andeanhighland"
"columbia-w"
"columbia-e"
"columbia-basin"
"ecuador-coast"
"ecuador-highlands"
"Venezuela-Llanos"
"Venezuela-coastalplain"
"suriname-coastalplain"
"suriname-plateau"

Figure 4.15: Geographical Areas Included in the GIRO System

"cotton-producing-area"
"coffee-growing-area"
"wine-producing-area"
"potato-growing-area"
"rubber-producing-area"
"goats-area"
"rice-growing-area"
"wheat-growing-area"
"soybean-growing-area"
"rubber-producing-area"
"sheep-country"
"producing-banana"
"producing-tropical-fruits"
"corn-growing-area"
"sugar-producing-area"
"fruit-veg-growing-area"

Figure 4.16: Regional Characterization Included in the GIRO System

temperate	arctic	woodland	plain	mediterrainian
plateau	Mts	coastal-land	lake	tropical
lowland	hill	river-valley-basin	swamp	rainforest
evergreen	deciduous	highland	upland	sparsely-populated
densely-populated	fertile	infertile	flood	prairie
dependable-rainfall	scrub	farming	rugged	subtropical
rainy	savanna	dry-arid	grassland	desert

Figure 4.17: Geographical Features Included in the GIRO System

forest floor. Grasses grow in poorly drained areas, while coarse grasses and brush grow on sandy, porous soils and rocky slopes.

In extracting information from source books, there are some subtleties that have to be taken into considerations. Each article regarding a particular region is written by a particular researcher familiar with that region, and varies in depth, presentation, amount of details and emphasis. This diversity in style, resulting from the diversity in the authorship of these articles, inevitably has adverse effects on the accuracy of the specification. The problem is the lack of details on the one hand and too much detail on the other hand. When there are not enough details from one sourcebook, we can find another sourcebook, and try to fill in what is needed. In case of too much detail, we have to be very careful in selecting the most important and relevant information out of the tangled web of irrelevant descriptions. As a rule of thumb, we usually disregard information associated with phases such as "plus", "in addition", "besides", "although", "a small portion of", "mostly..... but.....", etc. A problem is that few regions are geographically homogeneous. What we want is a description that is applicable to the largest portion of a region, expressing its *essential* characteristics, without having irrelevant information or descriptions that can only be applied to a small part of that region. There is certainly a tension between (1) capturing important characteristics of a region, and (2) excluding information applicable only to a small part of a region. The tradeoff between these two aspects helps to decide what primitive geographic regions are and what information is to be included for each such region.

After extracting the necessary information, we have to encode them into the connectionist architecture. As explained before, each geographical area is represented by a single node in the CL level, each feature is represented by a single node in the CD level, and appropriate links are established between the two levels. All agricultural characterizations are also represented in the CL level, and are replicated in the CD level. Links are established to indicate which area produces which product. All CL links have weights 1, each representing the certainty of the knowledge a link expresses. CD link weights are obtained from the specification in Chapter 3.

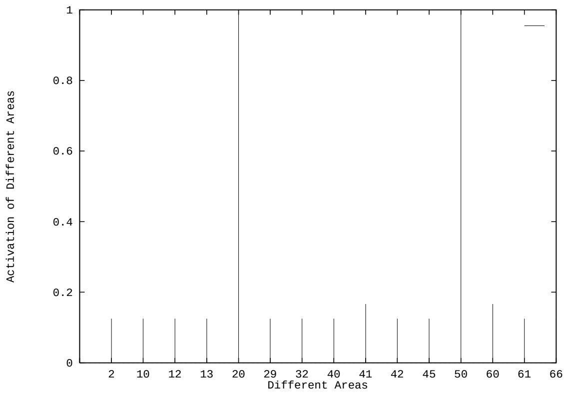
4.5.3 The Working of GIRO

Now we are ready to describe the working of the system. GIRO operates this way: once a name of a geographical area is given to GIRO, as imposing a query, GIRO will find out its agricultural characterization, such as "cattle country", "rice-growing area", or "rubber-producing area". through rule application or similarity matching, or a combination of the two.

For example, let us choose to reason about "Brazil-north", which is described as "tropical rainforest hilly plateau". We will start by giving GIRO a query: What is the main agricultural product of "Brazil-north"? That amounts to activating the node representing "Brazil-north". To answer this question, we let GIRO run to perform its reasoning.

The output from GIRO is as follows:

```
>(consyderr 0)

 TITLE:  GEOGRAPHY
 focusing on context AGRICULTURE : remove feature NIL
 setup done
 starting running
 top down
 cl propagating
 cd propagating
 bottom up
```



2	"cattle-country"	0.1249998807907104
10	"fruit-veg-growing-area"	0.1249998807907104
12	"producing-banana"	0.1249998807907104
13	"producing-tropical-fruits"	0.1249998807907104
20	"rubber-producing-area"	0.9999990463256836
29	"c-peru"	0.125
32	"Bolivia-orient-rainforest"	0.125
40	"guiana-pgs"	0.125
41	"guiana-hilly-country-forest"	0.1666666666666667
42	"guiana-hilly-country-savanna"	0.125
45	"Brazil-cw"	0.125
50	"Brazil-n"	1
60	"columbia-basin"	0.1666666666666667
61	"ecuador-coast"	0.125
66	"suriname-plateau"	0.125

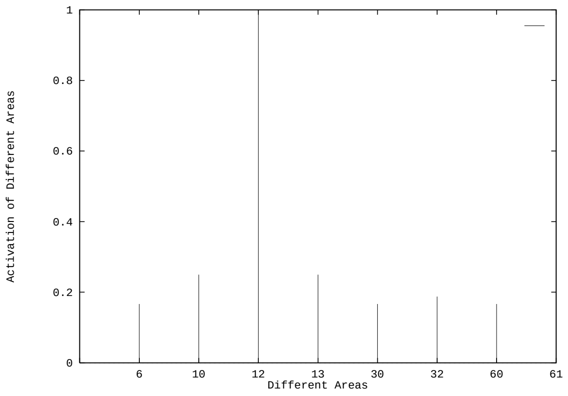
Figure 4.18: Output From GIRO: Case 1

```
 the average activation is 0.1213409896658248
 ( 2, "cattle-country", 0.1249998807907104 )
  ( 10, "fruit-veg-growing-area", 0.1249998807907104 )
  ( 12, "producing-banana", 0.1249998807907104 )
  ( 13, "producing-tropical-fruits", 0.1249998807907104 )
  ( 20, "rubber-producing-area", 0.9999990463256836 )
  ( 29, "c-peru", 0.125 )
  ( 32, "Bolivia-orient-rainforest", 0.125 )
  ( 40, "guiana-pgs", 0.125 )
  ( 41, "guiana-hilly-country-forest", 0.1666666666666667 )
  ( 42, "guiana-hilly-country-savanna", 0.125 )
  ( 45, "Brazil-cw", 0.125 )
  ( 50, "Brazil-n", 1 )
  ( 60, "columbia-basin", 0.1666666666666667 )
  ( 61, "ecuador-coast", 0.125 )
  ( 66, "suriname-plateau", 0.125 )
```

The result shows that it is a rubber-producing area for sure (with confidence value equal to 0.999999), and it is similar, to a small extent, to “Guiana hilly country” and “Bolivia orient rainforest area” etc. If we want to choose one answer out of many, we can simply use a winner-take-all network on top of this, but this is not an intrinsic part of GIRO and is not needed here. See Figure 4.18.

Another example is as follows: suppose we want to know about Ecuador coastal area, we will give GIRO a query: What is the main agricultural product of Ecuador coast? by activating the node representing “Ecuador coast”. To answer this question, we let GIRO run to perform its reasoning. The output from GIRO is as follows:

```
(consyderr 0)

 TITLE:  GEOGRAPHY
 focusing on context AGRICULTURE : remove feature NIL
  setup done
```



6	"Uruguay-coastal"	0.1666666666666667
10	"fruit-veg-growing-area"	0.2499997615814209
12	"producing-banana"	0.9999990463256836
13	"producing-tropical-fruits"	0.2499997615814209
30	"e-peru",	0.1666666666666667
32	"Bolivia-orient-rainforest",	0.1875
60	"columbia-basin"	0.1666666666666667
61	"ecuador-coast"	1

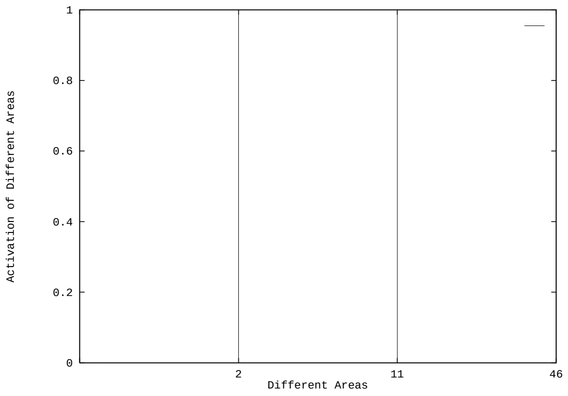
Figure 4.19: Output From GIRO: Case 2

```
 starting running
 top down
 cl propagating
 cd propagating
 bottom up

the average activation is 0.1433035089856102
( 6, "Uruguay-coastal", 0.1666666666666667 )
 ( 10, "fruit-veg-growing-area", 0.2499997615814209 )
 ( 12, "producing-banana", 0.9999990463256836 )
 ( 13, "producing-tropical-fruits", 0.2499997615814209 )
 ( 30, "e-peru", 0.1666666666666667 )
 ( 32, "Bolivia-orient-rainforest", 0.1875 )
 ( 60, "columbia-basin", 0.1666666666666667 )
 ( 61, "ecuador-coast", 1 )
```

The result indicates that the area is producing banana (with confidence value equal to 0.99999) and is very likely producing tropical fruits and other fruits/vegetables. It is similar, in some way, to "Uruguay-coastal", "eastern-Peru" and "Columbia-basin". See Figure 4.19.

As yet another example, let us reason about "Brazil south". We will start by giving GIRO a query: Does "Brazil south" produce cattle? by activating the node representing "Brazil south" and looking for "cattle" in the results. The output from GIRO is as follows:

```
(consyderr 0)

 TITLE:  GEOGRAPHY
 focusing on context AGRICULTURE : remove feature NIL
 setup done
 starting running
 top down
 cl propagating"
 cd propagating "
 bottom up
```



2 "cattle-country" 0.9999990463256836
11 "sheep-country" 0.9999990463256836
46 "Brazil-s" 1

Figure 4.20: Output From GIRO: Case 3

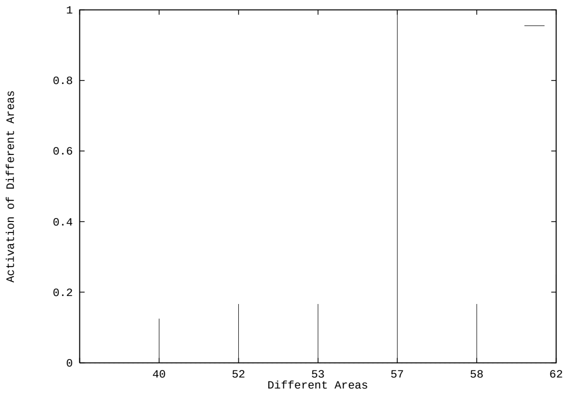
```
the average activation is 0.1492545754568917
( 2, "cattle-country", 0.9999990463256836 )
 ( 11, "sheep-country", 0.9999990463256836 )
 ( 46, "Brazil-s", 1 )

 The Result ----
( 2, "cattle-country", 0.9999990463256836 ) ---end of results
```

The result indicates that the area does produce cattle and sheep. Nothing else in the network fires strongly or distinguishably in this case. See Figure giro3.

Yet another example is about “Argentina Andean Highland”: Is it a cattle country? We activate the node representing the area, and the output from GIRO is as follows:

```
(consyderr 0)

 TITLE:  GEOGRAPHY
 focusing on context AGRICULTURE : remove feature NIL
 setup done
 starting running
 top down
 cl propagating
 cd propagating
 bottom up

 the average activation is 0.0861359060756744
 ( 40, "guiana-pgs", 0.125 )
  ( 52, "chile-s", 0.1666666666666667 )
  ( 53, "chile-c", 0.1666666666666667 )
  ( 57, "Argentina-andeanhighland", 1 )
  ( 58, "columbia-w", 0.1666666666666667 )
  ( 62, "ecuador-highlands", 1/4 )

  The Result ----
 ( 2, "cattle-country", 0.08333325386047362 ) ---end of results
```

This result shows clearly that it is not a cattle country. See Figure giro4.



40	"guiana-pgs"	0.125
52	"chile-s"	0.1666666666666667
53	"chile-c"	0.1666666666666667
57	"Argentina-andeanhighland"	1
58	"columbia-w"	0.1666666666666667
62	"ecuador-highlands"	1/4

Figure 4.21: Output From GIRO: Case 4

To conclude this section, a system (GIRO) for reasoning about geographical characteristics of certain regions is developed, which organizes relevant knowledge into concepts, features, and rules (and similarity measures that emerges from the feature representation) fitting perfectly into the CONSYDERR architecture. The knowledge in the system is extracted from source books, without arbitrarily hand-tuning. Various examples show that the system works reasonably despite the large number of data items stored in the system, and the interferences/distractions that may be resulted from having a large system.

4.6 Applications to Other Domains

The previous sections demonstrated that the CONSYDERR architecture is suitable for reasoning in geographic domains, because of the existence of the natural division of knowledge into two categories — concepts and features. Now the question is: Can this same method be applied to other domains where no such natural division seems to exist? We will look into this question, and show that the same approach as embodied in CONSYDERR does work for other domains.

Because of the space limitations, and because of the fact that this is outside of our main focus, we will not concern ourselves with all the details and will not conduct a thorough study of the domains involved. Only some brief hints as to how this architecture and approach can be applied to these other domains will be provided.

Subsection One will touch upon natural language understanding and disambiguation. Subsection Two will apply the architecture to Mundane Reasoning as in Derthick (1988). Subsection Three will address planning and temporality.

4.6.1 Applications to Natural Language Understanding

Natural language understanding is an area in which commonsense reasoning is crucial. Some discussions, through an example, will be useful in clarifying how the CONSYDERR architecture might be applied to this important domain.

In applying CONSYDERR to this area, it is important to identify all the concepts that are involved and the rule structures (i.e. the set of rules that connect related concepts); Another important aspect is identifying all the features, which are associated with the concepts identified above, with or without specific meanings attached (i.e. interpretations). We do not have ready answers for all the issues involved, for they are highly domain specific. For practical purposes, we can either perform a thorough domain analysis to identify useful features along with concepts, or use some statistical methods to determine similarity and, based on that, construct the CD representation (see Appendix at the end of the chapter for details). Yet another important issue is how to focus on relevant features around a particular objective as mentioned earlier, which implies the explication of the relationships between contexts and features.

One of the most difficult problem in natural language understanding is that of lexical disambiguation (cf. Waltz & Pollack 1985, Bookman 1989, and Pustejovsky 1991). One simple example that lends itself to our discussion is regarding "Pot", taken from Lange & Dyer (1989). We adopted the problem, but provide a different solution. The problem can be stated as follows:

> John put the pot inside the dishwasher, because the police are coming.

The point is that normally the word "pot" should be interpreted as "cooking pot", but under certain circumstances, given some pertinent clues, it should be interpreted as marijuana. Depending on what contexts or clues are given, a reasoning system with proper knowledge should be able to give correct answers.

We provide a set of additional situations to test our system in its ability to solve the problem, which are not in Lange & Dyer (1989).

(1) **The First Test**:

> John put the pot inside the dishwasher.

What does the word "pot" mean here? The solution should be "cooking pot" in this case.

(2) **The Second Test**:

> John put the pot inside the dishwasher, because the police are coming. What does the word "pot" mean here?

The word "pot" means "marijuana" in this case.

(3) **The Third Test**:

> John put the pot inside the dishwasher, because the police are coming and John wants to make the kitchen clean.

The solution should be "cooking pot" in this case.

(4) **The Fourth Test**:

> John put the pot inside the dishwasher, because John wants to make the kitchen clean.

The solution should be "cooking pot" in this case.

(5) **The Fifth Test**:

> John put the pot inside the dishwasher, when the police came for the bankrobbery across the street.

The situation is pretty ambiguous, and the system could interpret it as "marijuana".

(6) **The Sixth Test**:

> John is cleaning the kitchen, putting the pot inside the dishwasher, when the police came for the bankrobbery across the street.

The solution should be "cooking pot" in this case.

We constructed a system that can solve the original problem and pass the six tests, based on the CONSYDERR architecture, as follows:

Each of the concepts involved in the problem description is represented by one node in CL. Rules are extracted from commonsense knowledge about the concepts involved in the story. Rules that are actually put into the systems include (described informally):

1. Pot is cooking pot.
2. Pot could be marijuana.
3. Marijuana is illegal.
4. Cleaning kitchens implies cleaning cookware.
5. Using dishwashers implies cleaning cookware.
6. If one is having marijuana and the police are coming, then the police will see it.
7. Police seeing illegal substance results someone being arrested.
8. To avoid arrests, prevent the police seeing illegal substances.
9. To prevent somebody seeing something, hide it.
10. Putting something in a dishwasher is for washing it.
11. Putting something in a dishwasher could be for hiding it.
12. If there is a bankrobbery going on and the police are coming, then they are here to stop the bankrobbery.

On the other hand, CD is basically a distributed version of CL by sharing nodes among related concepts. This CD structure is constructed according to STSIS (see Appendix at the end of the chapter). Each node is vaguely interpretable. The rules in CL are duplicated diffusely in CD.

When the parameters are appropriately set the system performs the task correctly. It gives correct answers to all tests, distinguishing, often very subtle, differences, through rule applications and similarity matching. [6]

The advantage of this solution, using rule applications plus similarity matching instead of using only rules, is that it increases the expressive and reasoning power of the system by enabling analogous knowledge to be used, as discussed before in Chapter 2. Although this example is very simple and cannot possibly capture all the intrincasies involved, it does point out, to a small extent, an alternative way of dealing with lexical ambiguity and natural language understanding in general.

[6] Note that some subtle linguistic elements are not taken into account in the present implementation, for example, "*when* the police are coming" vs. "*because* the police are coming", etc. When finer distinctions are needed, more nodes will have to be added into the system, along with possibly other mechanisms, to take those elements into consideration.

4.6.2 Applications to Mundane Reasoning

Mundane reasoning is another area in which commonsense knowledge is heavily involved. Mundane reasoning is used to refer to the type of reasoning that we do daily regarding mundane matters, for example, which chair to sit in, when to eat, etc. The goal of such mundane reasoning is to come up rapidly with an interpretation of a situation or to make a quick decision, given the current context. Therefore it involves a lot of subconscious thinking or "subcognition" (cf. Hofstadter 1985).

In applying CONSYDERR to this area, we have to identify for CL all the concepts and rules involved in a particular task. We also have to identify all the features associated with the concepts, through some means.

We will look into the "Ted" example from Derthick (1989). Again, we adopted the problem, but provide a different solution: instead of doing constraint satisfaction as in Derthick (1989), we perform rule-based reasoning plus similarity matching. The problem can be stated as follows (cf. Derthick 1989):

> Ted is seen walking along a pier, dressed like a sailor. Ted launched into an excited monolog on the influence of TV programming. It seems reasonable to conclude that Ted is a professional sailor, and that he is interested in Television. But another possibility is that Ted is TV Tycoon and a millionaire playboy and has a hobby of sailing.

The point is that normally Ted should be taken to be a real sailor, but under certain circumstances, given some pertinent clues, it should be interpreted as a hobby sailor. Depending on what contexts or clues are given, a reasoning system with proper knowledge should be able to give correct answers. This task does not have to be done by constraint satisfaction through minimizing some energy functions (as in Derthick 1988), but can be done by reasoning based on rules and similarity. This way the results are less susceptible to small changes in parameter settings.

The knowledge used for performing this type of mundane reasoning is encoded in rules:

1. *dressed-like-sailor talks-like-sailor walks-like-sailor* $\longrightarrow$ *sailor*
2. *talk-about-TV* $\longrightarrow$ *interested-in-TV*
3. *talk-about-TV* $\longrightarrow$ *in-TV-business*
4. *in-TV-business* $\not\longrightarrow$ *sailor*
5. *dressed-as-sailor* $\longrightarrow$ *hobby-sailor*
6. *hobby-sailor* $\longrightarrow$ *rich-people*
7. *rich-people* $\not\longrightarrow$ *sailor*

In CD, distributed representation is used so that similar concepts have shared nodes in that level. CD is constructed according to STSIS, and each node is vaguely interpretable. The rules implemented in CL are duplicated diffusely in CD.

We provide a set of tests to verify the correctness of the system the same way as before. The system works as expected, that is, when given the input that "Ted is dressed up like a sailor" (in the form of activating nodes "dressed-like-sailor") the system will indicate that Ted is a sailor (in the form of activating nodes "sailor" strongly); when given the input that Ted is dressed up like a sailor but talks a lot about TV business (in the form of activating nodes "dressed-like-sailor" and "talk-about-TV-business"), the system will indicate hobby sailors (in the form of activating nodes "hobby-sailor" more strongly). Overall, the CONSYDERR approach works well and should prove to be of great use in this area.

4.6.3 Applications to Planning

Planning is yet another area in which commonsense reasoning is important (we are mainly concerned here with commonsensical planning activities, not formal planning based on strict mathematical models). Since we have not implemented anything in this area, we will only discuss some thoughts and speculations that are related to the usefulness of CONSYDERR for this domain.

The most important problem is that the sequential nature of the planning domain has to be adequately dealt with. The planning problem is inherently sequential: A plan is formed from a sequence of steps. Moreover, steps can interfere with each other in the form of undoing what was accomplished or disabling what should be done, etc. Some notions of temporality have to be incorporated in order to express sequences. And actions, conditions, and results have to be associated with temporal measures.

The question is how to account for temporality in CONSYDERR. CONSYDERR, by itself, is non-sequential. So we have to use means other than what is in CONSYDERR. This is where the idea of *temporal simulation* comes into play. By temporal simulation, I mean actually carrying out plan steps temporally inside CONSYDERR when forming a plan. Since the system can deduce what the next step should be within a system cycle, given the current situation, then another cycle can further deduce yet another step, with the previously derived state as the current one. This process can go on and on, until reaching desired states.

One planning paradigm that alleviates the need for temporality consideration is *universal plans*. According to Schoppers (1989):

> Universal plans classify possible situations; naturally cover the entire domain; place no restrictions on what possible situations might become actual and in what order; and use sensory information to identify the class of possible situations, hence redeciding what to do from moment to moment.

CONSYDERR could provide a framework in which this type of planning can be carried out. In studying planning, by using a network model instead of a symbolic computational program, more realistic structures, constraints and processes can be explored. On the other hand, universal planning provides the conceptual framework in which CONSYDERR might be applied to the planning problems.

Rules and similarity are used readily in planning. Rules are used to encode the relationships between the plan step taken within the current context and the new state after the step is performed and the relationships between the current state and the next step to take. Similarity is used for matching a situation not precisely specified in the rule sets, for reaching plausible conclusions. It is also useful to have interactions in representation as will be discussed in Chapter 6.

Some issues arise in such a system. For one thing, a planning system based on CONSYDERR should preferably be able to adjust itself to new situations, new types of steps, etc. Although similarity matching is one possible way of doing it, we may need more complex similarity measures, and perhaps more complex types of features. Another issue is how to focus on relevant aspects in the feature representation of a concept. Something like system goals and purposes of a step, etc. can certainly help in this respect, but a lot more work has to be done for concrete solutions. Yet another issue is how to revise plans upon receiving feedback regarding the outcome of current actions (the network equivalent of searching the state space). Some kinds of feedback links are needed, and these links have to be structured in such a way that can help to identify steps to be avoided or to be favored, along with other considerations.

4.7 Summary

This chapter covers experimental studies of the CONSYDERR architecture. These experiments show that systems that are built based on this architecture can deal very well with, in a massively parallel fashion, rule applications (including issues such as partial match, incomplete information, uncertain knowledge, etc.), similarity matching, and inheritance, etc. , the most common elements in commonsense reasoning, which are the motivation and the goal of this research. A large scale system GIRO is designed for reasoning based on rules and similarity with geographic information extracted from encyclopedia. It shows that the CONSYDERR architecture can scale up to a reasonable size and perform useful tasks. Finally, we extend the domain of application, and look into such areas as natural language understanding, mundane reasoning, and planning.

4.8 Appendix: Determining Similarity and the CD Structure

STSIS, or *a Statistical Test-Score procedure for determining Intensional Similarity*, is a procedure for determining, by empirical means, feature (intensional) similarity in CD and for constructing the CD representation. [7] This procedure can be applied to determine CD representations for systems based on CONSYDERR, without performing or without being able to perform a domain analysis to find out all relevant features (because not all areas have a set of features well analyzed like in geography).

Assume there is a set of concepts c = { c1, c2, c3, ... }, and $\overline{c}$ is the vector composed of all these concepts.

Matrix M_2 measures the pairwise similarity between elements of the vector $\overline{c}$, i.e.,

$$M_2 = S(\overline{c} \times \overline{c})$$

where $\times$ denotes outer-product, and S is the similarity measure: S([x]) = [S(x)] (or in other words, the similarity of a matrix is a matrix of the similarity of its elements).

For each matrix element, which is a pair (a, b) in the above case,

$$S(a,b) = \frac{\sum_{i=1}^{n} S_i'(a,b)}{n}$$

where $S_i'(a,b)$ is a empirical measure of similarity between a and b, ranging from 0 to 1. We can also calculate the mean squared error:

$$\delta(a,b) = \sqrt{\sum_{i=1}^{n}(S - S_i')^2}$$

Then for higher order similarity, we have

$$M_3 = S(\overline{c} \times \overline{c} \times \overline{c})$$

and

$$M_4 = S(\overline{c} \times \overline{c} \times \overline{c} \times \overline{c})$$

etc. [8]

One problem with this approach is that there are too many entries to fill in each matrix, especially in higher order ones. One way to deal with this problem is determining which entry will be zero beforehand and thus avoiding testing that entry, because a large number of entries in these matrices will be zero, which can be determined by looking at related entries in the lower order ones (if one of the related entries is zero, then the entry is zero). *Related entries* are defined to be all entries that involve a subset of concepts involved in the original entry.

Another problem is when we should stop, because obviously we do not want a large number of M matrices (it can produce matrices as many as the number of concepts). There is no theoretical result determining when to stop. However, we can set up some empirical criteria. For example, we do not want to have more matrices than half the number of the concepts involved.

Once the similarity measures are obtained, a pseudo-code description of the algorithm for constructing CD representations based on the similarity measures is as follows: Suppose we have the following matrices: M_1, M_2, ..., M_m. The total number of nodes in CD is L. The number of nodes for c_i is L_i. [9] Define S_i to be L_i.

[7] For related work, see, for example, Zadeh (1981).

[8] Ideally, we should have $S_{ab} = \frac{|F_a \cap F_b|}{|F_b|}$, $S_{abc} = \frac{|F_a \cap F_b \cap F_c|}{|F_c|}$, and $S_{abcd} = \frac{|F_a \cap F_b \cap F_c \cap F_d|}{|F_d|}$, etc.

[9] L_i is determined based on the principle of similarity based representation: the more general a concept is, the fewer feature nodes there are for representing it

Let i=2
Repeat if i $\neq$ m
.For each set of entries (e.g. M_i (a, b) = S_{ab}, and M_i (b, a) = S_{ba}), make an appropriate number of nodes shared among those concepts in the entry, and subtract the same number of nodes from each of the node pools established for the related entries of the next lower numbered matrix
.i= i+1

Here *related entries* mean entries consisting of a set of concepts that is a subset of the original concept, for example, (a b c) for (a b c d). *An appropriate number of nodes* mean the number of nodes proportional to the similarity measure in question. For example, suppose M_i (a, b) = S_{ab}, and M_i (b, a) = S_{ba}, so the appropriate number of nodes will be $S_{ab} * L_b = S_{ba} * L_a$ (assuming they are equal [10]). However, in reality this equality may not hold: because of the fact that human similarity judgments and measurements are always error-prone, inconsistency is inevitable (cf. Tversky 1977). Besides, random noises alone are enough to upset the equality. When inconsistency is encountered, we can use the average of the two instead in the above formula.

Another issue is the subtraction of nodes from the node pool of the related entries in the lower order matrices. Because of the fact that a high order similarity is part of some lower order similarity, the (feature) nodes used in the the high order similarity are part of the node pools of the lower order similarity. When we allocate nodes for a high order entry, we must subtract the same number of nodes from each of the related lower order entries. Since we establish node pools for similarity iteratively, from the lowest order up, each time we only need to subtract from the related entries of the closest lower order matrix.

The question of which nodes to remove from a pool of nodes can be answered partially by considering constraints we have, that is, the fact that we have to preserve established similarity (i.e. node sharing situations). When the constraints we have are not enough to determine an uniquely correct way of removing nodes, we can make a tentative decision and backtrack later on if necessary, i.e., performing a search over the space of all possible ways of removing nodes, until a test shows that all similarity measures are implemented correctly.

[10] According to the definition of similarity above, the equality always holds. It is because $S_{ab} * L_b = \frac{|F_a \cap F_b|}{|F_b|} * L_b = |F_a \cap F_b|$, and $S_{ba} * L_a = \frac{|F_a \cap F_b|}{|F_a|} * L_a = |F_a \cap F_b|$.

Chapter 5

More on the Architecture: Logic and Causality

5.1 Overview

Behind the coarse effect is a fine cause ....
— Emerson, *Essays, First Series: Circles*

From our previous specification and derivation of the CONSYDERR architecture, some of the questions regarding rules and FEL (*Fuzzy Evidential Logic*) [1] still linger:

- What is FEL exactly?
- How do we encode rules with FEL and implement it in CONSYDERR?
- Why is this encoding advantageous?

This chapter will investigate these issues.

In any rule-based reasoning systems, it is important, even crucial, to be able to represent and capture commonsense causal knowledge: reasoning is largely causal based, and conversely, this capability renders much power to a rule representation and reasoning scheme. The requirement of capturing commonsense causal knowledge provides useful constraints as to what kind of rule encoding is feasible, therefore helping in the understanding of rule representation in connectionist models.

We will first discuss in very general terms causation and its importance in representation and reasoning. We will then analyze a *Causal Theory* and its advantages and inadequacies in dealing with commonsense causal knowledge; this analysis will give us some hints as to possible ways of obtaining a better solution. Based on this we will formally define FEL, with a connectionist "implementation" in mind, and we will look into its logical capability and equivalence, and its realization in CONSYDERR. We will look into the relationship between FEL and causality in general, and show that FEL is suitable for dealing with commonsense causal knowledge and it requires nothing more than the weighted-sum connectionist model we defined before. Finally we will look into the issue of weights in FEL, which are yet to be specified, and will provide some interpretations. In short, this chapter shows that connectionist models using the weighted-sum computation are a natural way of "implementing" rules and a suitable framework for capturing commonsense causal knowledge.

To summarize the plan:

[1] Here the word *fuzzy* is used to refer to the gradedness or continuous inexactness of a concept or a proposition, not necessarily referring to Zadeh's definition of fuzziness based on linguistic variables.

- The second section will look into causality and causal theories in general.
- The third section will discuss a particular *Causal Theory* and problems with it.
- The fourth section will be a formal presentation of FEL, as a potential solution to the problems identified above.
- The fifth section will discuss how FEL deals with causality.
- The sixth section will explain weights involved in the formalism.

5.2 Causality in General

Every why has a wherefore.
— Shakespeare

This section will examine the issue of causality, its relationship to reasoning, and its role in representation.

Causality has been widely studied, but its nature, its role, or even its very existence are still in question. Its nature as the "cement of the universe" (Hume 1740), may entail the difficulty in understanding it. The Humean notion of a *ceteris paribus* sufficient and/or necessary condition as a definition of causes [2] has long been recognized as problematic and infeasible (cf. Mackie 1975 and Lewis 1975). Due to the problems encountered in studying causation along this line, the general trend is towards viewing causality as complex processes and as indeterminate (or probabilistic) processes (Sosa 1975 and Suppes 1970).

As a start, we can examine a theory of causality by Salmon (1984). He proposed the concept of "process" as a basic entity in his theory, rather than an event as in more traditional theories. A causal *process* transmits energy, information and causal influence. He defined the principle of mark transmission, the principle of structural transmission, and the principle of propagation of causal influence, which states: "A process that transmits its own structure (transmitting changes in structure) is capable of propagating a causal influence from one space time locale to another." One problem with this theory is that it does not address the issue of how to identify a cause out of a myriad of co-occurring events. A second problem with this theory, especially processes as the "mysterious link between cause and effect" (Hume 1740), is its lack of a concise and accurate representation. As a result the power of causal reasoning cannot be fully utilized.

Suppes (1970) advocates a probabilistic view of causality. He recognizes the *uncertainty* in the world, so according to Suppes, causes are, simply put, a set of events that proceeds the effect event and that enables one to predict the effect event with a higher probability. In case there are multiple causes, the real cause is the *earliest stronger predictor*, defined in Suppes (1970) by precise mathematical treatment. He distinguishes, based on his definition, prima facie causes, spurious causes, genuine causes, etc. Hempel (1965) proposed a statistical relevance theory of scientific explanation. According to this view, explanation is inductive-statistical. and deductive-nomological explanation is viewed as a special (limiting) case of causality. These theories have their problems, as debated in the philosophical literature, of which space does not permit a detailed examination. From an epistemological view, it seems to me that they make too strong an assumption – the axioms for the probabilistic structures, making it impossible to identify causal relations if there is no well-defined probabilistic structure.

[2] Hume defines causality as "an object followed by another, and where all the objects, similar to the first, are followed by objects similar to the second. Or in other words where, if the first object had not been, the second never had existed." (Hume 1740).

For theories of causality geared towards natural language understanding, Schank (1972) proposed his definition of a "causal chain", which is composed of six primitives, such as *enable, result*, etc., that define the relationships between different actions. [3] Although interesting, his approach is more oriented towards analyzing surface features of linguistic expressions, rather than understanding deep causal relations. From the point of view of modeling commonsense reasoning, a problem is that the "causal chain" is all-or-nothing, and there is no degree of causal relatedness that is essential if we adopt the view that the causality can be uncertain and commonsense knowledge can be vague.

Iwasaki & Simon (1986) and de Kleer & Brown (1986) show how causal relations can be identified in device behavior (as well as in economics and thermodynamics). It is argued in Iwasaki & Simon that analyses of a set of algebraic or differential equations that describes the behavior of a device can produce an ordering which entails causal relatedness and causal directions, and thus can identify causality in that device. de Kleer & Brown on the other hand insist on directly analyzing qualitative behavior of devices and thus obtaining causal relations. As in Schank (1972), both proposals are deterministic and deal with only qualitative causal relations between a set of clearly defined variables without provisions for accommodating vague concepts and indeterminate relationships.

The model proposed by Shoham & Dean (1985) is more in line with modeling commonsense causality. It provides some constructs (e.g. *clipping functions* and *generating functions*) that are more representable. They define direct and indirect causes: generally speaking, the direct cause of a fact token is the set of other fact tokens that are related to the fact token to be explained by a generating function. They further divide fact tokens into subtypes such as potential fact tokens, materialized fact tokens, ghost fact tokens, and real fact tokens. The indirect cause, in turn, is a chaining of direct causes. A generating function is what determines which potential fact tokens exist. A clipping function is what determines how each potential fact token is decomposed into materialized fact tokens and ghost fact tokens. It seems that, from a connectionist viewpoint, a generating function can be equated roughly to a set of positive links from a set of fact tokens (causes) to one fact token (effect), and moreover a clipping function can be equated to inhibitory links.

In Shoham (1987), this theory is developed into a modal logic formalism, in which some facts are necessary, and some others are also needed but less important for a given event to occur. According to this theory, we do not have to worry about relevant but non-essential facts in reasoning in the commonsense world, as long as their potential falsehood is not obvious. Thus we have two types of conditions: the necessary conditions and the possible conditions. Time is explicitly specified in this theory: a cause has to proceed its effects temporally; thus the causal influence is strictly one-directional. [4] However, the theory cannot deal with partial information, uncertain causal connectivity, and vagueness of concepts, which are all essential to commonsense reasoning, as will be shown later in the chapter.

Among the theories described, I share with Salmon the view that causality has to be studied and modeled as a complex process, and I share with Suppes the view that causality is not deterministic, especially for commonsense reasoning, and finally I share with Shoham the view that causality can be modeled by conditionals (rules) with different conditions of different importance playing different roles [5].

Let us look into now the relationship between causality and reasoning in general. Reasoning, to a large extent, can be viewed as the formation, modification, and manipulation of the internal model of the environment one is in. This internal world model captures commonsense knowledge of

[3] Actually the theory is more concerned with relations between sentences.

[4] This point of view is not universally accepted. Imagine the scenario in which the cause of a event is perceived later than the perception of the effect, so for a commonsense reasoner the effect proceeded the cause in time.

[5] However, in my view, "rules" are not isolated and they form an interacting process in this regard (see Chapter 6 for details).

the world and allows one to predict, anticipate, and meditate upon an outcome of a current event, possible consequences of an action, and future directions resulting from the interactions of various forces. Based on this internal model, people can reason about a given situation and set up a goal, and in order to achieve a certain goal, people can form a plan for the best course of action, carry out necessary steps, and handle exceptions. An internal causal model seems to be what is underlying this precise yet flexible, reliable but fallible process. Reasoning based on a world model utilizes artificial causation, changing the model without actually affecting the real world (cf. Sowa 1986).

An internal model of causality can somehow be modeled in a network fashion, capturing causal relations between elements of the world. Each element of the model corresponds to an element in the world, and each causal link represents a direct causal relation between elements in the world. The indirect cause is captured in the chaining of direct causal links (cf. Shoham and Dean 1985). Reasoning can be viewed as a search in a network modeling causal connectivity.

For the operation (i.e. search) of a world model, a rule-based approach is preferred over the idea of spreading activation (Collins & Loftus 1975). Collins & Loftus observe that the sequential activation of rules or other symbolic structures is much too slow to account for some behavioral data and cannot account for the speed, automaticity, and simultaneity exhibited by human thinking; so spreading activation over a network of prewired links representing all associations seems to be a remedy. What is wrong with this picture, I believe, is the following: (1) When talking about sequential operation of rules, they have actually in mind a von Neuman computer, which does go through a slow serial process of fetch, process and store, one data item at a time, but this is not the best model available today; (2) there are too many different things in this world associated with too many different other things, a fully automatic spreading of activation will not be able to distinguish strong causes from weak associations. (A limited model might work well when associations are fully controlled for the task, but in the real world a vast network with numerous connections will be a totally different situation.) On the other hand, a massively parallel computational device should be able to handle multiple rules at the same time. When rules (such as FEL) are designed to represent the causality of the world, with a massively parallel device (such as CONSYDERR) they behave *as if* the operation is an automatic spreading of activation. However, this "spreading of activation" is rule-governed or rule-directed (cf. Holland et al 1986 for similar points).

Causal models can serve as economical representational frameworks. Some common patterns of causality identified by philosophers include causal forks and interactive forks (cf. Salmon 1984, Pearl 1987 etc.). Causal forks, as a statistical structure for common causes, explain the non-coincidental co-occurrence of two or more events. The use of causal forks in an internal model of the world, to a certain extent, can be attributed to storage economy. Instead of multiple disparate chains of events, we only need to store one token for each distinct event. By assuming a common cause, we can easily trace the co-occurrence of events, with only local interactions (i.e. interactions occurring along the causal links of a causal fork). This results in succinct storage and easy operation of a world model. We can even have invisible (unobservable) causes in a model and such invisible causes are merely names given to storage places, which, by holding partial results, facilitate efficient manipulation of visible variables in the system (cf. Pearl 1988). The interactive fork is another common causal situation, in which the interaction of two causal processes causes two or more events which are considered effects. The interactive fork also serves as an economical construct in world models, in a way similar to the clipping function in Shoham and Dean's model.

In summary, causality not only provides a scientific framework in which the physical world is explained (such as in physics), but also provides a cognitive model of the world, indicating what to expect under a given set of circumstances. Therefore, it is imperative that a causal theory be represented explicitly and compactly, so that it serves as a data structure for representing empirical causal knowledge of the world and so that its modular organization allows efficient computation and modification with minimum central control and coordination.

5.3 Shoham's Causal Theory

In this section, based on our general discussion in the previous section, we will look specifically into the causal theory of Shoham (1987) that relates causality to rule-based reasoning (i.e. a rule-based account of causal reasoning). Its merits and limitations in representing commonsense knowledge will be discussed.

Shoham's theory is undoubtedly one of the most notable contemporary accounts of causation with rule-based formalisms. It utilizes ideas from temporal logic, modal logic and non-monotonic logic, and provides a synthesis from semantical (model-theoretical) viewpoints. The resulting temporal modal logic formalism has a close resemblance to Horn clause logic, and therefore is very suitable for use in rule-based systems. (However a syntactical theory for this logic is not provided, and all algorithms are devised based on the semantics of the logic.)

An informal description of Shoham's Causal Theory (CT) is as follows: Causes are primary conditions which, together with other conditions, will bring about the effect. These "other" conditions are somewhat secondary. In reasoning, as long as we know that the primary conditions (*causes* or *necessary conditions*) are true and that there is no information that the secondary conditions (*enabling conditions* or *possible conditions*) are false, then we can deduce that effects will follow. This definition makes the theory non-monotonic, because if later on one of these secondary conditions become known to be false, then the reasoning has to be retracted somehow. Shoham insists on using temporal notions in the logic to make sure that the condition always proceeds the effect. The theory is described in terms of modal logic, with one basic modal operator ($\Box$ or *necessity* for specifying necessary conditions). Another modal operator ($\Diamond$ or *possibility*) for specifying possible conditions is defined based on the first one (*necessity*).

The formal definition is as follows:

Definition 5.1 *A* **Causal Theory** *is a set of formulas of the following form*

$$\wedge_i \Box n_i a_i(t_{i1}, t_{i2}) \wedge_j \Diamond n_j b_j(t_{j1}, t_{j2}) \longrightarrow \Box c(t_1, t_2)$$

where n_i's are either $\neg$ or nothing, $t_2 > t_{i2}$ for all i's, $t_2 > t_{j2}$ for all j's, $n_i a_i$'s are necessary conditions (causes), and $n_j b_i$'s are possible conditions (enabling conditions). C is concluded iff all $n_i a_i$'s are true and all $n_j b_j$'s are not known to be false. [6]

With regard to its computational complexity, Shoham (1987) shows, by a constructive proof, that:

Theorem 5.1 *The problem of inferring all results in CT in accordance with above definition (i.e., computing all facts that are true in all minimal model of a CT, according to Shoham's original definition) is polynomial.*

From our standpoint of modeling commonsense knowledge, this model has the following advantages:

- It provides a simple and elegant formalism with efficient inference algorithms.
- It is easily representable (and implementable).
- It synthesizes several radically different approaches: modal logic, temporal logic, non-monotonic logic, and causal reasoning.

[6] This process is formally described by a *minimization* principle in Shoham 1987.

- It has compatibility with philosophical accounts of causality in the form of incorporating important points from Mackie, Lewis and Suppes (cf. Shoham 1987).

On the other hand, the model is intended only to be a *normative* description of causal reasoning and, as a consequence, it ignores or discounts many aspects of commonsense causal reasoning. Specifically, I have the following reservations about the theory:

1. All propositions in this theory are binary: either true or false, and there is no sense of gradedness. Commonsense knowledge is certainly not limited to true/false only. For example, "The dry weather during that period caused the loss of crops later" can be expressed in CT as

$$\Box \textit{dry(t,t+1)} \longrightarrow \Box \textit{crop-loss(t+1, t+2)}$$

   Here "dry" is a fuzzy concept, and there is no well-defined cut-off point as to what is dry and what is not.

2. Because of the lack of gradedness, the model will make projections too far along a chain of reasoning (or too far into the future). An example from Shoham (1987):

$$\Box alive(t_0, t_0)$$

$$\Box shoot(t,t) \longrightarrow \Box\neg alive(t+1, t+1)$$

$$\Box alive(t,t)\Diamond\neg shoot(t,t)\Diamond\neg \textit{otherwise-killed(t,t)} \longrightarrow \Box alive(t+1, t+1)$$

   which means that if one is alive at time t_0, one will continue to be alive as long as not being shot or otherwise killed. So if there is nothing known about "shoot" and "otherwise-killed", then according to the minimal model approach, we will predict that

$$\Box alive(t,t) \quad where\ \ t \longrightarrow \infty$$

   This is certainly not true. The problem is that, along a chain of inference (as well as in temporal projections), the confidence for the conclusions reached should weaken. We can weaken confidence along the way only when gradedness is reinstated into causal theories.

3. The clear-cut *necessity* and *possibility* is a problem, because in reality there is little, if any, qualitative difference between *causes* and *enabling conditions*. The difference is more quantitative (as will be illustrated later), and sometimes the two are interchangeable; for example, "He is shot dead" is expressed as

$$\Box shoot(t,t) \wedge \Diamond\neg \textit{wearing-bullet-proof-vest(t,t)}...... \longrightarrow \Box dead(t+1, t+1)$$

   and "His failure to wear the bullet-proof vest caused his tragic death" is expressed as

$$\Box\neg \textit{wearing-bullet-proof-vest(t,t)} \wedge \Diamond shoot(t,t)...... \longrightarrow \Box dead(t+1, t+1)$$

   So one fact can be both a cause and an enabling condition. This also suggests that enabling conditions have a more important role in causal reasoning than Shoham's model presupposes.

4. Although the model does distinguish two different types of conditions, it does not explain why some conditions are necessary, and some conditions need only to be possible.

5. According to the model, it is necessary to list all causes and all enabling conditions, in order to guarantee correct results. This could be hard to do, because the number of enabling conditions could be infinite. For example, there are too many reasons why someone is not dead after being shot at: the gun did not work and therefore did not actually fire, the person is wearing a bullet proof vest, there happens to be a stone in front of the person, another strayed bullet defrayed the bullet aimed at that person, and many others.

6. The causal connection between events in the left hand side of an implication and events in the right-hand side of the same implication may not be deterministic. It could be probabilistic, or otherwise uncertain (as discussed before; cf. Suppes 1970).

In view of these problems, we wonder if it is possible to fix this formulation by extending it somehow. The answer to this is yes. We are going to fix the problem by introducing gradedness into causal models. In the following sections, we will propose a new formalism, and evaluate it in light of what is discussed here.

5.4 Defining FEL

In this world, nothing is certain but death and taxes.
— Benjamin Franklin

We will first present definitions of FEL (including FEL_1 and FEL_2 — two versions of *Fuzzy Evidential Logic*). Then we will study their relations to traditional and not-so-traditional logics. Specifically, we will look into Horn clause logic and Shoham's Causal Theory.

FEL is aimed at resolving the problems inherent in existing logical account [7] of causality . However in this section, we will not get into the details of how FEL can do the trick. Instead we will present FEL per se for now, and later sections will deal with that issue.

Like Shoham's formalism, FEL is defined around rules (or conditionals). FEL encodes rules with the weighted-sum computation. In implementations of FEL, those computations are distributed across a set of *nodes*. Each node has one or many *sites*, where a group of incoming links representing one single rule rests. Each site of a node performs weighted-sum computation of activation values from its incoming links. Then the node will get the MAX of all activation values of different sites. Each node represents a fact, and each site with all its links represents a rule. Facts and rules are basic elements of this logic.

Formal definitions follow:

Definition 5.2 *A* **Fact** *is an atom or its negation, represented by a letter (with or without a negation symbol) and having a value between l and u. The value of an atom is related to the value of its negation by a specific method, so that knowing the value of an atom results in immediately knowing the value of its negation, or vice versa.* [8]

For example, a, ¬a, x, u, etc. are all facts.

Now we can define rules and their related weighting schemes:

Definition 5.3 *A* **Rule** *is a structure composed of two parts: a left-hand side (LHS), which consists of one or more facts, and a right-hand side (RHS), which consists of one fact. When facts in LHS get assigned values, the fact in RHS can be assigned a value according to a weighting scheme* [9]*.*

[7]FEL is not exactly a logic, at least not in the exact sense of the word. It is based on the most common connectionist formalism — the weighted sum model. So in a way it can be viewed as simply an interpretation of the connectionist model in terms of logic. In other words, FEL provides a semantics for connectionist models based on logics.

[8]We will adopt a generic confidence measure as the value of a fact. We will later further characterize this measure.

[9]When the value of a fact in LHS is unknown, assign a zero as its value.

For example, $a\ b\ c \longrightarrow d$ is a rule, and a weighting scheme associated with it will tell us the value of d once we know the values of a, b and c.

Definition 5.4 *A* **Weighting Scheme** *is a way of assigning a weight to each fact in LHS of a rule, with the total weights less than or equal to 1, and of determining the value of the fact in RHS of a rule by thresholded (if thresholds are used) weighted-sum of the values of the facts in LHS (or inner-products of weight vectors and vectors of values of LHS facts). When the range of values is continuous, then the weighted-sum is passed on if its absolute value is greater than the threshold, or 0 if otherwise. When the range of values is binary (or bipolar), then the result will be one or the other depending on whether the weighted-sum (or the absolute value of it) is greater than the threshold or not (usually the result will be 1 if the weighted-sum is greater than the threshold, 0 or -1 if otherwise).* [10]

For example, a weighting scheme for the above rule is

$$w_1 = 0.3, \quad w_2 = 0.3, \quad w_3 = 0.4.$$

and the value of d is calculated by their weighted-sum (or inner-product)

$$d = w_1 a + w_2 b + w_3 c$$

We will define the notion of a theory:

Definition 5.5 *A* **Theory** *is a 4-tuple: $< A, R, W, T >$, where A is a set of facts, R is a set of rules, W is a weighting scheme for R, and T is a set of thresholds each of which is associated with one element in A.*

One issue that needs to be clarified is how to handle multiple rules reaching the same conclusion (i.e. the same fact is at the RHS of more than one rule, and they are all activated), in which case we have to combine the results somehow. One simple way is to take the MAX of all values from various rules regarding the particular fact concluded. As a matter of fact, this is the *best* way to do it, given all the constraints and considerations we have (when a system is consistent), as will be elaborated fully in Chapter 6 when we discuss rule interactions.

Definition 5.6 *A* **Conclusion** *in FEL is a value associated with a fact, calculated from rules and facts by doing the following:*

(1) for each rule having that conclusion in its RHS, obtain conclusions of all facts in its LHS (if any fact is unobtainable, assume it to be zero); and then calculate the value of the conclusion in question using the weighting scheme;

(2) take the MAX of all these values associated with that conclusion calculated from different rules or given in initial input.

Note that *conclusion* is defined here recursively. Thus depending on the structure of a particular theory, this definition may or may not make sense (e.g. in case of recursive structures in a theory). So in order to use this definition and the procedure for calculating conclusions implied by the definition, we have to restrict the structure of theories allowed.

Definition 5.7 *A rule set is said to be* **Hierarchical**, *if the graph depicting the rule set is acyclic; the graph is constructed by drawing a unidirectional link from each fact (atom) in LHS of a rule to the fact (atom) in RHS of a rule.*

Making a rule set hierarchical avoids circular reasoning.

Now FEL can be defined as follows:

[10] This weighting scheme can be generalized, as will be discussed later on.

Definition 5.8 *A* **Fuzzy Evidential Logic** *(FEL) is a 6-tuple:* $< A, R, W, T, I, C >$*, where A is a set of facts (the values of which are assumed to be zero initially), R is a set of rules, W is a weighting scheme for R, T is a set of thresholds each of which is for one rule, I is a set of elements of the form* (f, v) *(where f is a fact, and v is a value associated with f), and C is a procedure for deriving conclusions (i.e. computing values of facts in RHS of a rule in R, based on the initial condition I).*
We want differentiate FEL into two versions: FEL_1 and FEL_2, which differ in their respective ranges for values associated with facts.
Definition 5.9 FEL_1 *is FEL when the range of values is restricted to between 0 and 1 (i.e. l=0 and u=1), and the way the value of a fact is related to the value of its negation is:*

$$a = 1 - \neg a$$

for any fact a.
Definition 5.10 FEL_2 *is FEL when the range of values is restricted to between -1 and 1 (i.e. l=-1 and u=1), and the way the value of a fact related to the value of its negation is:*

$$a = -\neg a$$

for any fact a.

The above abstract logical definitions seem far removed from any network implementation. Now in order to connect these notions with CONSYDERR networks, we will define formally *implementations* of FEL:
Definition 5.11 *An* **Implementation** *of FEL is a network of elements connected via links, where each node represents an atom and its negation (there is a one-to-one mapping between an atom and a node) and links represent rules, going from nodes representing facts in LHS of a rule to nodes representing facts in RHS of a rule.*
We define the notion of an *element* formally:
Definition 5.12 *An* **Element** *is a structure that represents one and only one fact and has multiple sites each of which receives a group of links that represents one single rule (i.e. links from facts in LHS of the same rule that has the fact which this element represents in its RHS.*

It is obvious, from the above discussion, that a FEL and one level of CONSYDERR (either one) have a one-to-one correspondence, up to isomorphism:
Theorem 5.2 *There is an isomorphic one-to-one mapping between FEL and CL (or CD).*
For proofs, see Appendix. [11]

Let us examine FEL carefully. First of all, as an illustration of its capability and correctness, we want to show that FEL can implement Horn Clause logic as a special case (we will only deal with the propositional version here, and extensions to first order cases will be dealt with in Chapter 7). Let us define Horn clause logic first.
Definition 5.13 **Horn clause logic** *is a logic in which all formulas are in the forms of*

$$p$$

or

$$p_1 p_2 ...... p_n \longrightarrow q$$

where p's and q are propositions.
We are now ready to see how FEL can simulate Horn clause logic.

[11] In the following discussion, we will not refer further to the implementation of FEL. Instead we will conduct discussions with the formal notations at an abstract level. It should nevertheless be kept in mind that FEL corresponds to one level of CONSYDERR.

Definition 5.14 *A* **Binary FEL** *is a reduced version of FEL (either FEL_1 or FEL_2), in which values associated with facts are binary (or bipolar), total weights of each rule sum to 1, and all thresholds are set to 1.*
Here is the theorem for the equivalence (see Appendix for proofs):
Theorem 5.3 *The binary FEL is sound and complete with respect to Horn clause logic.*

Having dealt with Horn clause logic, we want to show that FEL can simulate a modal logic — Shoham's Causal Theory, to further explore the logical capability of FEL. We will first redefine the Causal Theory:
Definition 5.15 *A* **Causal Theory** *is a set of formulas of the following form*

$$\wedge_i \Box n_i a_i \wedge_j \diamond n_j b_j \longrightarrow \Box c$$

where n_i's are either $\neg$ or nothing, $n_i a_i$'s are necessary conditions (causes), and $n_j b_i$'s are possible conditions (enabling conditions). C is concluded iff all $n_i a_i$'s are true and all $n_j b_j$'s are not known to be false.
This is a non-temporal version [12] of CT, that is, we strip away all temporal notations and treat the same proposition in the original definition with different temporal variables as different propositions (and represent them with different letters). In the original definition of CT, it is required to be anti-reflexive and anti-symmetric, but we will take a stronger position (to be sure that it is not over-restrictive, it suffices to point out that this stronger position is logically implied by the definition of the temporal version of CT), i.e., a CT has to be hierarchical (or, equivalently, the transitive closure of a CT is anti-reflexive and anti-symmetric).

What we need to do now is to find a weighting scheme that can equate FEL to Causal Theory. But before we can do that, we have to find a mapping between truth values of formulas in Causal Theory and values of facts in FEL. Since in CT and in FEL, there is no logical OR and there is only a (almost implicit) logical AND in the RHS of a rule, which can be taken care of by a weighting scheme as will be discussed later, we do not have to worry about these two connectives in the mapping now. Therefore, We can use a mapping as follows, which can be easily verified to be consistent with regard to logical equivalence (without AND and OR; for example, $\Box$ a = $\neg\Diamond\neg$ a, etc.):

(1) M(a= true) = 'a=1'
(2) M($\neg$ a= true) = 'a=-1'
(3) M($\Box$ a= true) = 'a=1'
(4) M($\Box\neg$ a= true) = 'a=-1'
(5) M($\Diamond$ a= true) = 'a=0'
(6) M($\Diamond\neg$ a= true) = 'a=0'
(7) M($\neg\Box$ a= true) = 'a=0'
(8) M($\neg\Box\neg$ a= true) = 'a=0'
(9) M($\neg\Diamond$ a= true) = 'a=-1'
(10) M($\neg\Diamond\neg$ a= true) = 'a=1' [13]

With the mapping in hand, we can proceed to find a weighting scheme to enable FEL to simulate Causal Theory. The problem is that in FEL we have nodes only for atoms such as a, b, m, n, etc. but not for $\Box$a or $\Diamond$b, etc. We have two ways of dealing with this:

[12] The reasons for removing the temporal notations are (1) that in commonsense reasoning it seems to me not the case that causes must precede effects in time (for example, the effect can be perceived before the cause), and (2) that we can do whatever reasoning the temporal version is capable of by introducing different atoms for propositions at different time intervals, and (3) that this version can be easily extended into the temporal one. See also Lewis (1975).

[13] Note that this mapping amounts to a S5 system (cf. Hughes & Creswell 1968), but we can easily make the mapping a K45 (or weak S5) system. See Appendix for details.

1. Extending and making more complex the weighting scheme,
2. Adding nodes that can be used to represent atoms with modal operators.

We will adopt the first approach here (the second approach will also work — the difference is insignificant). For a formula in Causal Theory

$$\wedge_i \Box n_i a_i \wedge_j \Diamond n_j b_j \longrightarrow \Box nc$$

we can assign arbitrary weights to atoms: a_i's and b_i's (if there is a negation, the corresponding weight is negative; otherwise, weights are positive), as long as their absolute values sum to 1. However, for b_i's, we will also apply the following function to the link between b_i and c:

$$f_j(b_j) = \begin{cases} 1 & \text{if } b_j = 1 \\ 1 & \text{if } b_j = 0 \text{ and } n_j \neq \neg \\ -1 & \text{if } b_j = 0 \text{ and } n_j = \neg \\ -1 & \text{if } b_j = -1 \end{cases}$$

We will call this function the *elevation* function because it turns all 0's into 1's or -1's. We have thresholds equal to 1 for all rules. We restrict the possible values of facts to -1 or 1. [14]

Now it is easy to verify that a rule in FEL with this specific weighting scheme and thresholds is equivalent to a corresponding formula in Causal Theory: Suppose we have the following formula in CT:

$$\Box a \Box b \Diamond a \Diamond b \longrightarrow \Box e$$

It can be translated into FEL as follows:

$$abc'd' \longrightarrow e \quad (w_1 w_2 w_3 w_4)$$

where $c' = f_c(c)$ and $d' = f_d(d)$ and $\sum_i w_i = 1$, and the threshold equal to 1 for the rule. Let us look into it case by case:

- If a and b are known to be true (or in FEL "a=1" and "b=1"), and c and d are unknown (or in FEL "c=0" and "d=0"), then according to CT, e can be proved. The same in FEL: a=1, b=1, c'=1, d'=1, $w_1 * 1 + w_2 * 1 + w_3 * 1 + w_4 * 1 = 1$, so we conclude "e=1".
- If a, b, c and d are known to be true (or in FEL "a=1", "b=1", "c=1" and "d=1"), then according to CT, e can be proved. The same in FEL: a=1, b=1, c'=1, d'=1, $w_1 * 1 + w_2 * 1 + w_3 * 1 + w_4 * 1 = 1$, so we conclude "e=1".
- If a and b are known to be true (or in FEL "a=1" and "b=1"), and c and d are known to be false (or in FEL "c=-1" and "d=-1"), then according to CT, e cannot be proved. The same in FEL: a=1, b=1, c'=-1, d'=-1, $w_1 * 1 + w_2 * 1 + w_3 * -1 + w_4 * -1 < 1$, so we cannot conclude "e=1".

[14] If we add nodes representing $\Diamond a$ and $\Diamond\neg a$, for all nodes a, then there is no need for this special function f on links. Instead we will have some extra nodes $\Diamond a$ and $\Diamond\neg a$. which have incoming links (with weights equal to 1) from their corresponding nodes a and whose thresholds are 0. For $\Diamond a$,

$$\Diamond a = \begin{cases} 1 & \text{if } w * i >= 0 \\ -1 & otherwise \end{cases}$$

For $\Diamond\neg a$,

$$\Diamond\neg a = \begin{cases} -1 & \text{if } w * i >= 0 \\ 1 & otherwise \end{cases}$$

It accomplishes the same as the elevation function f.

- If a and b are known to be false (or in FEL "a=-1" and "b=-1"), and c and d are unknown or known to be false, then according to CT, e cannot be proved. The same in FEL: a=-1, b=-1, c'=1, d'=1, $w_1 * -1 + w_2 * -1 + w_3 * 1 + w_4 * 1 = 1$, so we cannot conclude "e=1".

Other cases can be analyzed the same way.

To find a full correspondence between FEL and Causal Theory, however, we also need a proof theory that enables the derivation of all correct results (theorems). In Shoham's Causal Theory inferences are defined in terms of minimal models in a model theoretical approach. Here we present a proof procedure instead, in order to facilitate the correspondence with FEL:

Given a Causal Theory CT, and a set of initial conditions (true events) I:

———————————————

— For all a ∈ I, infer a, □a, and ◇a.
— Repeat:
for $\wedge_i \Box n_i a_i \wedge_j \Diamond n_j b_j \longrightarrow \Box c$ where $n_i a_i$'s are inferred, and $\neg n_j b_j$'s are non-inferable, [15]
infer c, □c, and ◇c.

———————————————

It is easy to see the correctness of this procedure:

Theorem 5.4 *The above proof procedure is sound and complete for Causal Theory as defined above.*
We can have a similar proof procedure for FEL:

Given a FEL theory, and a set of initial conditions (true facts) I:

———————————————

— For all (a,v_a) ∈ I, infer $a = v_a$.
— Repeat:
for $\wedge_i n_i a_i \longrightarrow c$ where each $n_i a_i$ is inferred with a certain value, or is non-inferable (and therefore a value zero is assumed), [16]
infer c with v_c, where v_c is calculated according to the weighting scheme used.

———————————————

It is easy to see the correctness of this procedure for FEL, and the mapping between the two proof procedures:

Theorem 5.5 *The above proof procedure is sound and complete for hierarchical FEL*

Theorem 5.6 *The proof procedure for FEL carries out exactly the proof procedure for Causal Theory when Causal Theory is implemented in FEL in the aforementioned way.*

See Appendix for the proof. Therefore,

Theorem 5.7 *For every hierarchical, non-temporal Causal Theory, there is a FEL such that CT: $w \models a$ iff FEL: $c \models$ 'a=1', where w is a possible world for Causal Theory CT, and c is the initial condition for FEL mapped over from w in CT.*

Proof

It follows directly from the above analysis.

Q.E.D

In terms of the complexity of the procedure, we have the following theorems:

Theorem 5.8 *The problem of deciding if the value of a fact is a certain given number (or deriving that number), given a hierarchical FEL and a set of initial facts and their values, is polynomial.*

[15] They are not in the RHS of any rule and not in I, or in order to infer it, we have to use a rule which has a fact as a necessary condition in its LHS that is not inferable. Since CT is hierarchical, this is easy to detect. We can pre-construct a "dependency graph" which depicts inferability relations.

[16] According to the weighting scheme used to simulate CT, if a fact is inferred, it must be inferred with a value 1; if a fact is non-inferable, then its value is 0. When other weighting schemes are used, the results will be different.

Theorem 5.9 *The satisfiability problem (the problem of whether a set of facts and their values are consistent* [17]*) for hierarchical FEL is polynomial.*

In order to keep our perspective, there are some issues to be addressed regarding the correspondence between rules and connectionist models. One is that there are also other ways to establish correspondence between rules and connectionist models. For example, in Oden (1990) a fuzzy logic formalism is transformed into a connectionist model with logistic activation functions. This work adds another way of linking rules (logics) with connectionist models, especially the most commonly used one — the weighted-sum model.

Another issue is that the correspondence between FEL and CONSYDERR is not perfect, when the proof procedure is taken into consideration. For one thing, in the CD level which has a distributed, overlapping representation, the above proof procedure cannot be carried out as it is, because of the parallel nature of the network and the interaction that the parallelism creates (see Chapter 6 for a detailed discussion of this topic). On the other hand, since in CL there is no overlapping in representation, parallelism will not cause any problem as long as the FEL theory in question is consistent. To see this, notice that the above proof procedure is actually performing a search (with the rules as "operators"), but the procedure itself does not specify a particular order (such as breadth-first, depth-first, etc.) used in search. What happens in CL (according to the equations specified in Chapter 3) amounts to a parallel [18] breadth-first search. [19] For a consistent theory, a parallel search works perfectly.

To better understand the advantage of FEL and its place among various logics, we also compare FEL with Halpern's Logic of Probability AX_{MEAS} (see Halpern 1990). In that formalism, there is a fixed set of *primitive propositions* ϕ_i, and there are $\wedge$, $\vee$, and $\neg$ just as in ordinary propositional logic; what is different is *weight formulas* in the form of

$$\sum \alpha_i \beta(\phi_i) >= c$$

The logic is meant to be used for reasoning about probability, and is defined by complete axiomatization (see Appendix for a list). It was shown that:

Theorem 5.10 *AX_{MEAS} is sound and complete with respect to probabilistic structures.*

Furthermore,

Theorem 5.11 *The problem of deciding whether a weight formula is satisfiable in a probability structure is NP-complete.*

Although aimed for solving different problems and following different approaches, FEL can be viewed somehow as a subset of AX_{MEAS}, by mapping the corresponding definitions of the two different formalisms (see Appendix for details). Comparing the two logics, FEL has the following advantages:

- The formulas that can be proven in FEL are a subset of AX_{MEAS}, a subset that is most useful in terms of uncertain, evidential reasoning in AI.

- FEL has a polynomial time algorithm for its satisfiability problem, while AX_{MEAS} is NP-complete.

To summarize, in this section a new logical formalism FEL is proposed to account for commonsense causal knowledge. We show that it corresponds to the CONSYDERR architecture. The

[17] Consistency is here defined as satisfying the following condition: no fact can be deduced in FEL more than once with values of opposite signs. Note that it is a natural extension of the definition in binary logic for consistency. As we will see in the next chapter, most of the time, reasoning is inconsistent (or based on inconsistent rules, facts, etc.).

[18] It is parallel in the sense that all steps that can be explored at a given time are explored together at the same time.

[19] The equivalence is approximate; see Chapter 6 for more explanation

relationships between this formalism and some existing ones — Horn clause logic, Shoham's modal logic formalism, and others — are also explored to demonstrate the representational power of FEL. The reasoning efficiency of FEL is also shown.

5.5 Accounting for Commonsense Causal Knowledge

> *Cause and effect, means and ends, seed and fruit, cannot be severed; for the effect already blooms in the cause, the end pre-exists in the means, the fruit in the seed.*
> — Emerson, *Essays, First Series: Compensation*

Based on the review of previous causal theories, analyses of problems associated with them, especially with Shoham's theory, and the introduction of FEL, we are now ready to explain how FEL can extend Shoham's model to remedy the problems we mentioned previously. CONSYDERR will be shown to be a proper framework for representing commonsense causal knowledge.

We identified earlier the following problems with Shoham's formalism:

- Binary valued concepts.
- Constant confidence along a chain of reasoning.
- Distinguishing necessary and possible conditions.
- Clear-cut necessary and possible conditions.
- Enumeration of all conditions.
- Determinism in causal connections.

Now we are ready to extend Shoham's Causal Theory to solve these problems. Since in the previous section we showed that FEL can implement Shoham's Causal Theory, we can now extend that implementation, instead of the original version, because FEL offers a more general framework which allows greater flexibility.

To extend the FEL version of Shoham's Causal Theory, we first notice that the causes need not be known with absolute certainty, i.e. we should allow a confidence measure being associated with each *necessary* fact (i.e. the one with $\Box$), because of the gradedness, uncertainty and fuzziness of our knowledge. By the same token, the conclusions need not be binary either, so that uncertain causes can generate uncertain effects. Moreover, even facts (causes) of absolutely certainty may not guarantee the expected effects (i.e. the idea of uncertain causality; cf. Suppes 1970). Therefore, we will associate a confidence measure with each of the causes (i.e. the facts in LHS of a rule, to use the FEL terminology) between -1 and 1, and a confidence measure also with the effect (i.e. the fact in RHS of a rule, to use the FEL terminology). We can use weights to create a mapping between confidence measures of causes (i.e. values of the corresponding facts) and confidence measures of effects (i.e. values of the corresponding facts), so from a set of causes and their confidence measures (i.e. a set of facts and their values) we can deduce a confidence measure for an effect (or a value for a fact in RHS). Moreover, the set of weights associated with facts in LHS of a rule should reflect their relative importance: more important causes should have a larger weight associated with them, and since the total weights sum to 1, the value of a weight for a particular fact (condition) reflects its relative importance against a background of all other conditions.

Another issue to consider is how to handle *possible condition* facts (i.e. those with $\Diamond$). As explained before, there is a special function associated with them, which elevate 0 to 1 or -1, according

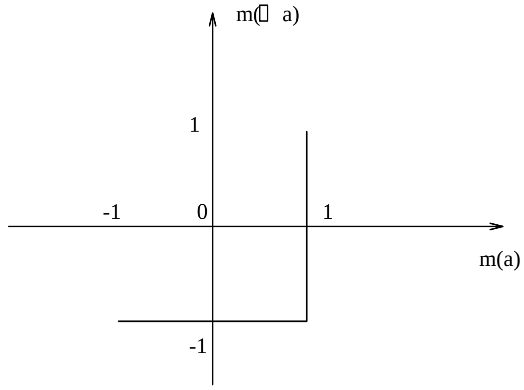


Figure 5.1: A Necessity Function

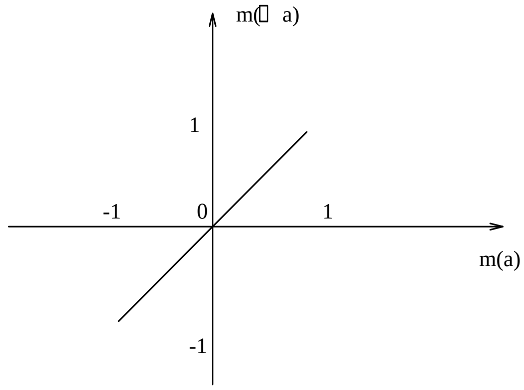


Figure 5.2: A Fuzzified Necessity Function

to whether positive or negative forms appear in the causal rule. Since we now extend the binary (or bipolar) space for truth values into a graded, continuous space, there is no more need for that elevation function. It follows from the fact that when a *possible condition* fact is unknown (i.e. its value is 0), the conclusion can still be reached, albeit with a smaller value (in confidence level). Now that we no longer require a binary (or bipolar) outcome, it is fine to have a smaller value for a conclusion when some enabling conditions are unknown. When one of these enabling conditions become known, the value will become higher, that is, we will have more confidence is the conclusions. Normally the weights associated with those enabling conditions will be relatively small anyway, because they are non-essential and close to "*don't care*" conditions. So it is advantageous to remove the elevation functions in the FEL version of Shoham's Causal Theory and assign weights instead.

An alternative perspective of viewing the extension is that of "fuzzifying" the necessity function and the possibility function. Instead of having a necessity function as depicted in Figure 5.1, we can have a fuzzified necessity function as in Figure 5.2, for the purpose of accounting for the inexact nature of real world situations and knowledge. Similarly, instead of having a possibility function as depicted in Figure 5.3, we can have a fuzzified possibility function as in Figure 5.4. Once fuzzified, these new functions wind up to be identity functions (as if no function at all). Therefore, combining the above two perspectives, *causes* are those conditions that have *high* weights, and *enabling conditions* are those conditions that have *low* weights.

We can now easily map the FEL terminology into the causal terminology as follows:

Events *are facts in FEL.*

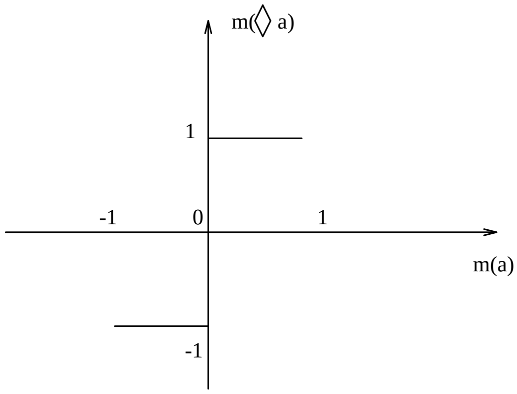


Figure 5.3: A Possibility Function

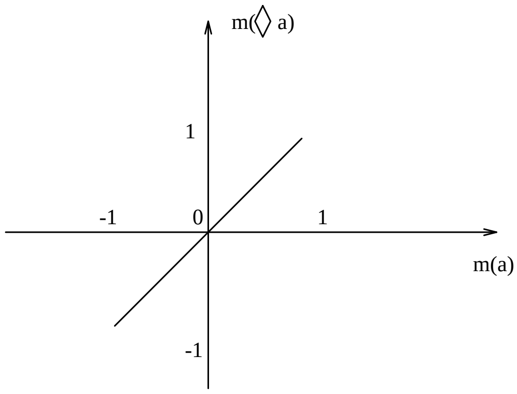


Figure 5.4: A Fuzzified Possibility Function

Causal Statements *are rules in FEL.*

Causes *are those conditions of a rule that have high weights associated with them according to some particular weighting scheme.*

Enabling Conditions *are those conditions of a rule that have low weights associated with them according to some particular weighting scheme.*

Effects *are facts in the RHS of a rule.*

Now that we have finished developing the new formalism FEL, we can go back to the issues we raised in Section 3, and demonstrate that this logic and this definition of causality can better handle these issues. The details follow:

- The gradedness is readily taken care of in our system by the confidence values we have associated with each fact.
- Because of the introduction of the gradedness and uncertain rules (i.e. total weights sum to less than 1), the confidence we have in the conclusions will weaken along the way in a chaining. For example, here is a FEL rule stating that if one is alive at time t, one will be alive at time $t + 1$:
$$alive(t) \longrightarrow alive(t+1)$$
Suppose the weight is equal to 0.99, then if given *alive(0)=1*, we will have *alive(1)=0.99, alive(2)=0.98, alive(3)=0.97*, and so on.
- There is no more need to tell exactly which condition is necessary and which condition is possible: they are all graded and the difference is only quantitative (see the next section on weights).
- There is no more need to list all conditions (the total number of which might be infinite), as long as we leave room in the weight distribution (i.e., keeping total weights less than 1). For example, as discussed before, there are too many reasons why someone is not dead after being shot at: the gun did not work, he wore a bullet proof vest, there is a stone in front of him, another bullet defrayed the bullet aimed at him, etc. We can list only those conditions that we care about, and by doing so, the sum of weights will then be less than 1, accommodating possible roles of other unlisted conditions in determining the causal outcome.
- The indeterminate or probabilistic nature of causality is readily captured in the weighting scheme: the weights do not have to sum to 1, and not all conditions have to be known for certain in order to deduce a plausible conclusion.

Let us further explicate what is implicit in the definitions of FEL regarding how causality is treated in various situations. In the FEL framework, causality is handled uniformly based on the following two basic principles:

The Minimality Principle: *Nothing should be assumed unless given or explicitly derived (i.e., making minimal assumptions about an agent's or a system's belief). If something is not known or cannot be derived, assume neutrality regarding its truth (i.e. values to be zero).*

The Weighting Principle: *In deriving a conclusion based on a causal rule, calculate the highest possible value of the inner product between the vector of the weights associated with the facts in the causal rule and the vector of the values of the facts in the condition, without violating the Minimality Principle.*

Let us look back to the shooting example. Instead of having two separate causal statements in CT:

$$\Box shoot(t,t) \wedge \Diamond\neg \textit{wearing-bullet-proof-vest(t,t)}...... \longrightarrow \Box dead(t+1,t+1)$$

and

$$\Box\neg \textit{wearing-bullet-proof-vest(t,t)} \wedge \Diamond shoot(t,t)...... \longrightarrow \Box dead(t+1,t+1)$$

we will have in FEL one single causal statement for all the situations:

$$\neg \textit{wearing-bullet-proof-vest} \wedge \textit{shoot}...... \longrightarrow \textit{dead} \quad \textit{weights} = (w_1, w_2, ...)$$

We apply the minimality principle and the weighting principle (or in other words, apply the previously defined FEL inference procedures), that is, assume the values of the unknown facts are zero and calculate the value of the conclusion by inner-products of the weights and the values of the facts in the LHS of the rule. So each fact in a LHS can be a cause, when the rest of the LHS is obvious in a certain context or is unknown for lack of information. When the rest of the LHS is known for certain in a given context, such as in "His failure of wearing the bullet proof vest caused his death", the value of the conclusion is high; when the rest of the LHS is simply unknown, the value of the conclusion is lower, representing a weaker causal connection.

In the CONSYDERR architecture, events are represented by nodes (including both causes and effects), and causal statements are represented by the links between a node representing causes and a node representing effects (recall also the mapping between causal terminology and FEL). The minimality principle is carried out by setting node activations to zero initially and making them stay at zero when there is no input. The weighting principle is carried out by the weighted-sum computation (with MAX) in each node that represents effects.

In GIRO, there are many examples of causal rules. For instance, in the Florida protocol, we have the following rule:

$$\textit{big-area warm-area flat-area fresh-water} \longrightarrow \textit{rice-growing}$$

which specifies the causal connection between being a warm, big and flat area with fresh water supply and being a rice-growing area. In another instance, we have the knowledge that the Brazil rainforest area is a rubber producing area (i.e., $Brazil \longrightarrow$ *rubber-producing*), and the features of that area include "tropical", "high-precipitation", "rainforest", "rugged", etc. The causal knowledge is captured by the links in CD (for replicating diffusely the CL link) that connect these features to the nodes representing *rubber-producing area*, which amounts to a rule (although it is not explicitly known):

$$\textit{tropical high-precipitation rainforest rugged} \longrightarrow \textit{rubber-producing}$$

So overall the causal knowledge is captured in GIRO in two different forms: one form is explicit causal rules in CL, such as the first instance, and the other form is links (diffusely replicating original rules in CL) between features of different concepts in CD.

Reasoning could either follow strict causal connections (rules), to find out the corresponding causal consequences, or it could utilize similar, related but not directly applicable causal rules, for example in case of analogy. In CONSYDERR, analogy is achieved through similarity in the CD representation, which is the lower level in the representational hierarchy. Through this low level similarity, one node (representing a proposition or a concept) finds another related node that can help in continuing the reasoning with its causal connections (which are not available to the original node). This capability extends the meaning of causality, and and can produce parsimonious representation of causality, by using prototypical causal connection. [20]

[20] This "analogy based causality" might also be explained by probabilistic theories, though maybe not very elegantly.

To summarize, FEL provides an elegant and justified way of extending Shoham's Causal Theory, and constitutes a step towards capturing the graded and inexact nature of real world commonsense causal knowledge. The CONSYDERR architecture with dual representation implements FEL and provides a set of primitives for representing and reasoning about various cases [21] of causality.

5.6 Determining Weights

One last question that arises in accounting for causality is how we interpret (and/or determine) weights in FEL rules. We will look briefly into this question.

First of all, there is the question of how to interpret values associated with facts, which has to be answered before we can possibly determine the nature of the weights used in calculating those values. Those values can be viewed, alternatively, as grades of fuzzy membership (Zadeh 1965), as probabilities, or as possibilities (Zadeh 1975), and so on. As a matter of fact, most of the time, fuzziness, probability, and other inexactness are tangled together, and cannot really be separated and dealt with in isolation. For example, in the sentence "John is probably very angry by now", there is uncertainty (probability), as represented by the word "probably", and fuzziness, since "very angry" is not something with clear-cut boundaries. Therefore in order to be able to handle all these in one framework, we must adopt a more generic measure. This new generic measure can be called a *generalized characteristic function*, which is over the domain of all possible objects, to borrow the term *characteristic function* from fuzzy logic. In the classical logic theory, for any predicate, there is a set of objects (from the universe of discourse) that make the predicate true; similarly, for a predicate in fuzzy logic (Zadeh 1988), there can be a set of objects that do not necessarily make it true to its full extent but make it true to a certain extent; the degree of the truth (fuzziness) of a particular object with regard to that predicate is determined by a characteristic function defined based on the particular predicate. Since in FEL we are not dealing only with fuzziness we adopt a more generic measure — a measure of the confidence that we have in that predicate considering fuzziness, probability and other factors with respect to the object and the context, which is thus a generalized characteristic function and can take a value between -1 and 1 (or between 0 and 1) [22] for each object in the universe.

Now let us discuss several different interpretations of weights for FEL_1 [23] that relate FEL to existing frameworks. First is **the fuzzy interpretation**. Suppose we have fuzzy sets A_1, A_2, ..., A_n, and B; and a fuzzy rule $A_1 A_2 .... A_n \longrightarrow B$ with a measure of its confidence

$$w(x_1, x_2, ..., x_n, y) =$$

$$\min(1, (1 - \min(m_{A_1}(x_1), m_{A_2}(x_2), ..., m_{A_n}(x_n)) + m_B(y)))$$

where x_i's are elements from the fuzzy set A_i's respectively, y is an element from B, and m's are grades of membership for respective fuzzy sets. (This measure is not a single value; cf. Zadeh 1988;

[21] Clearly we do not deal with *all* aspects of causality as discussed in numerous philosophical and psychological literatures, for example, the principle of temporal contiguity, the principle of space contiguity, or the principle of anticedence, etc. (cf. Anderson 1990 and Salmon 1984).

[22] Different ranges of values entail different interpretations of the meaning of a particular value. For example, if the range is between -1 and 1, 0 is viewed as "unknown". But if the range is between 0 and 1, 0 is viewed as "not".

[23] As a matter of fact, the two different formalisms FEL_1 and FEL_2 can be transformed into one another. Let m_1 stand for a value of a fact in FEL_1, and let m_2 stand for a value of the same fact in FEL_2, then

$$m_1 = \frac{m_2 + 1}{2}$$

See also Chapter 8). Then the computation of the confidence of the conclusion reached will be, given as input an element from each fuzzy subset:

$$\max_{y \in B} \min(\min(m_{A_1}(x_1), m_{A_2}(x_2), ..., m_{A_n}(x_n)), w(x_1, x_2, ..., x_n, y))$$

Instead of using one measure for the confidence of a rule, we can use a vector of measures, with its dimension equal to the number of conditions in a rule, corresponding to the relative importance of each of the conditions of the rule; and instead of using MIN/MAX, which does not accumulate evidence (see Chapter 2), we use + to accumulate evidence and * for the weighting of a piece of evidence, that is,

$$w_1 * m_{A_1}(x_1) + w_2 * m_{A_2}(x_2) + ... + w_n * m_{A_n}(x_n)$$

where w_i's are elements of the vector of confidence measures associated with rules, and x_i's are elements from the fuzzy set A_i's respectively, and m's are grades of membership for respective fuzzy sets. This interpretation relates FEL to fuzzy logic, but leaves out the question of how to determine the weights.

The second interpretation is **the probabilistic interpretation**: Suppose we have probabilistic events A_1, A_2, ..., A_n, and B. A rule like $A_1A_2....A_n \longrightarrow B$ can be readily interpreted as computation of p(B) from p(A_1), p(A_2), ..., p(A_n), i.e.

$$p(B) = p(B|A_1A_2....A_n)p(A_1A_2....A_n)$$

When A_i's are pair-wise independent and exhaustive, the above formula can be rewritten as

$$p(B) = p(B|A_1)p(A_1) + p(B|A_2)p(A_2) + ... + p(B|A_n)p(A_n)$$

Comparing it with our weighted-sum formula, we immediately notice the similarity: $p(A_i)$ are values of the conditions, $p(B)$ are the value for the conclusion, and $p(B|A_i) = w_i$. So in this case, we have an interpretation of the weights in FEL rules as conditional probabilities.

Probability and fuzziness can be, and in most cases are, mixed together, forming a single confidence measure. A way to obtain weights is to use some learning algorithms (connectionist or non-connectionist), so that weights can develop from the data presented. A learning algorithm adopted serves as an **operational definition** for weights in FEL. From the vast array of available learning algorithms, those that guarantee that each of the weights for a group of links reflects the relative importance of the corresponding link and the total weights for a group of links sum to be equal to or less than 1, or those that can be modified to fit this criterion, are suitable for our purpose. Many different algorithms seem to be applicable in this regard, such as the competitive learning algorithm (McClelland et al 1986) or ART (Grossberg 1987).

5.7 Summary

In order to explain and justify the way rules are encoded in CONSYDERR, this chapter turns to theories of causality and some modal logic formalism that purports to explain and express causality.

To succinctly summarize the conclusions we draw from the discussions in this chapter:

- FEL is a suitable way for representing rules and performing commonsense reasoning.
- Connectionist models are, therefore, fully capable of representing rules — syntactically there is little difference between connectionist models and rules in general.

- Causality plays an important role in commonsense reasoning and representation, and the power of FEL, and that of CONSYDERR, derives (in part at least) from its ability to represent commonsense causal knowledge of various sorts.

To a large extent, this chapter is devoted to showing that there is a direct correspondence between weighted-sum connectionist models and rules (or logics), notably without complicated transformations and intermediate steps (such as in the proof of Turing equivalence of connectionist models). The next chapter will discuss another set of questions regarding CONSYDERR that complements the discussion in this chapter, showing the unique characteristics of this model, different both from conventional rule-based systems and from conventional connectionist systems.

5.8 Appendix: Proofs For Theorems

Theorem There is a isomorphic one-to-one (static) mapping between FEL and CL (or CD).

Proof FEL is defined to be a 6-tuple $< A, R, W, T, I, C >$. A particular FEL theory is a 4-tuple $< A, R, W, T >$. CL is defined to be $< N, C >$. We will show a mapping from CL to FEL, while the inverse mapping can be done the same way. Let A=N. For each n ∈ N, do the following: find all links that have the form: (m, n); and among them find all links that rest on the same site (forming a group) i.e. $(m_1, n), (m_2, n), ..., (m_l, n)$, and make a corresponding rule in R, i.e. $m_1, m_2, ..., m_l \longrightarrow n$; do the same for all sites. The weights in all the node activation functions make up W in FEL. T is the same set of thresholds as in the node activation functions of CL.

Q.E.D.

Theorem The binary FEL is sound and complete with respect to Horn Clause logic.

Proof The inference rule for FEL is a variant of forward chaining, defined as follows: Let K be a set of FEL propositions, i.e. pairs (f,v) where f is a proposition and v is a real number between 0 and 1 (or between -1 and 1) representing the confidence that f is true. We assume that all propositions are uniquely represented in K (though their confidence values may be zero). The inference rule will simply add FEL propositions to K until no new propositions can be added:

Given the FEL rule: (p_1, w_1),...$(p_r,w_r)- >$ q

If (p_1, d_1),....(p_r, d_r) are in K, then

let d' = $w_1 * d_1 + ... + w_r * d_r$ and assume (q,d) be in K.

If d'>d, then replace (q,d) by (q,d').

Given FEL fact: q

then we simply replace (q,d) in K by (q,1).

Given a Horn clause theory H, one can produce a corresponding FEL theory F as follows: Each Horn clause $p_1, ...p_r- >$ q is transformed into a FEL clause by associating a weight w_i with each atom p_i, in such a way that the $w_i > 0$ and they sum to 1.

Suppose that we are given a Horn clause theory H. Let F be a FEL theory corresponding to T (as above), and assume that all propositions in K initially have certainty values 0, then (f,1) is a FEL logical consequence of F iff f is a logical consequence of H.

To see this, observe that a FEL proposition (f,1) is added to K if and only if all of the propositions in the body of the clause have certainty values 1. Thus, (f,1) is only introduced to K if there exists propositions $(p_1$,1),....,$(p_r$,1) in K. It follows that the FEL inference rule behaves exactly like the forward chaining operator for Horn clause logic when we restrict our attention to propositions of weight 1. Forward chaining is a sound and complete inference rule for Horn clause theories. f is a logical consequence of H, iff f can be inferred by forward chaining in H, iff (f,1) can be inferred by FEL inference.

Therefore, because of the fact that Horn clause forward-chaining inference rules is sound and complete, it is clear that Binary FEL is sound and complete with respect to Horn Clause Logic.

Q.E.D

A Weighting Scheme that can equate FEL to Causal Theory:
(1) M(a= true) = 'a=1'
(2) M(¬ a= true) = 'a=-1'
(3) M(□ a= true) = 'a=1'
(4) M(□¬ a= true) = 'a=-1'
(5) M(◇ a= true) = 'a=0'

(6) M($\Diamond\neg$ a= true) = 'a=0'
(7) M($\neg\Box$ a= true) = 'a=0'
(8) M($\neg\Box\neg$ a= true) = 'a=0'
(9) M($\neg\Diamond$ a= true) = 'a=-1'
(10) M($\neg\Diamond\neg$ a= true) = 'a=1'

This mapping amounts to a S5 system (cf. Hughes & Creswell 1968). To see this, notice that the axioms for S5 are

$$\Box(P \longrightarrow Q) \longrightarrow (\Box P \longrightarrow \Box Q)$$

$$\Box P \longrightarrow \Box\Box P$$

$$\neg\Box P \longrightarrow \Box\neg\Box P$$

$$\Box P \longrightarrow P$$

Therefore it is easy to verify that the above mapping satisfies all the axioms, and thus it is a S5 system.

We can also make the mapping a K45 (or weak S5) system by mapping $\Box a$ into 1-ϵ instead of 1 (and other related changes). Note that a K45 system has all the axioms as a S5 system except the last one, and it is easy to verify that this new mapping now no longer satisfies the last axiom in the S5 system.

Since in Causal Theory, there is no formula in which an atom appears without a modal operator proceeding it, the two systems are equivalent in this case.

Theorem The problem of deciding if the value of a fact is a certain given number (or deriving that number), given a hierarchical FEL, a set of initial facts and their values, is polynomial.

Proof We will use the proof procedure defined in Section 4. Obviously, Step 1 takes polynomial time. In step 2, the REPEAT loop takes a number of iterations up to the number of rules, by using a dependency graph as a data structure. In each iteration, time taken is proportional to the number of facts in LHS, which is bounded to be less than or equal to the number of total facts.

So the total amount of time taken is polynomial with respect to the number of facts and the number of rules.

Q.E.D

Theorem The satisfiability problem (the problem whether a set of facts and their values are consistent) for hierarchical FEL is polynomial.

Proof We can use the result of the previous theorem.

We have a set of facts and their values, and have to decide if they are consistent. Rules are always consistent among themselves, by definition, due to the adoption of the MAX operation in the rule definition. We can keep doing the following until we exhaust all facts: we pick up a fact out of the set, apply rules to them, and check if the results are consistent with the rest of the set, i.e., to make sure it is not the case that if one is positive and the other is negative. If they are consistent, pick up another fact out of the remaining facts in the set and repeat the process. If they are not, the set is not satisfiable.

The number of times the process is repeated is equal to the size of the fact set, which is less than or equal to the total number of facts. The time complexity of the above process being repeated is polynomial according to the previous theorem. So the total amount of time taken is polynomial with respect to the number of facts present, the number of total facts, and the number of rules.

Q.E.D

Theorem The proof procedure for FEL carries out exactly the proof procedure for Causal Theory when Causal Theory is implemented in FEL in the aforementioned way (the two system are sound and complete for each other).

Proof

Let us look at each step of the procedure.

Step 1: For implementing CT, the initial conditions are given with value 1. So inferring a=1 in FEL is equivalent to inferring a, $\Box a$ and $\Diamond a$ in CT.

Step 2: Inferability is dealt with in the same way in FEL and CT. When a rule is chosen in FEL, we infer the conclusion c=v_c according to the particular weighting scheme which maps FEL to CT. With this particular weighting scheme for implementing CT, v_c has to be 1, which means in turn we can infer, in CT, c, $\Box c$ and $\Diamond c$.

Q.E.D

The Logic AX_{MEAS}: In this logic (see Halpern 1990), there is a fixed set of *primitive propositions* ϕ_i, $\wedge$, $\vee$, and $\neg$ just as in ordinary propositional logic, and *weight formulas* in the form of

$$\sum \alpha_i \beta(\phi_i) >= c$$

The axioms for AX_{MEAS} are

1. Tautologies. All instances of propositional tautologies.

2. MP. From f and $f \longrightarrow g$ infer g.

3. Inequality. All instances of valid linear inequalities.

4. W1. $\beta(\phi) \geq 0$.

5. W2. $\beta(true) = 1$.

6. W3. $\beta(\phi \wedge \psi) + \beta(\phi \wedge \neg\psi) = \beta(\phi)$.

7. W4. $\beta(\phi) = \beta(\psi)$, if $\phi = \psi$ is a propositional tautology.

The relationship Between FEL and AX: As mentioned before, FEL can be viewed somehow as a subset of AX_{MEAS}, by mapping the corresponding definitions in the two different formalisms. FEL is not exactly a subset of AX_{MEAS}. We have to add to FEL the restrictions that there are no multiple rules having the same fact in their RHS and that the range of values is [0,1]. To see their relationship with the above added restrictions, we have to turn FEL rules into logical relations, that is, for rules $\wedge a_i \longrightarrow c$ with weights w_i's, making $c = \wedge_i(a_i \wedge b_i)$, where $\beta(b_i) = w_i$, and $\beta(a_i) = value\ \ of\ \ a_i$. Instead of a particular value v for a conclusion as in FEL, we can use a range $[v, v+\epsilon)$ in AX. Therefore the weight formulas used in AX for determining the value of a fact in corresponding FEL (i.e. 'c=v') are in the form of $\beta(c) \geq v$ and $not(\beta(c) \geq v+\epsilon)$.

Chapter 6

More on the Architecture: Beyond Logic

6.1 Overview

Logic is one thing and commonsense another.
— Elbert Hubbard, *The Note Book*

A few other topics regarding CONSYDERR will be discussed in this chapter. We will concentrate on three different issues: inheritance, representational interactions, and knowledge acquisitions for CONSYDERR. The discussion of these issues together shows the *difference* between rules in symbolic systems and rules in connectionist systems. Mainly, the difference is the difference between isolated, context-free inferences and complex, interacting processes.

The first section will further explore the issue of inheritance. We will analyze how a complete inheritance hierarchy can be implemented, and the issues of cancellation, multiple inheritance, and new categories. The second section will discuss the problem and utility of interaction in representation: the generative capability in CONSYDERR. The third section will briefly touch upon the problem of learning and knowledge acquisition. Since these topics per se are beyond the main focus of this work, we will not talk much about the basics. For those uninterested in details, some summary is provided in the end of each section.

6.2 Further Analysis of Inheritance

We discussed in Chapter 3 several isolated cases of inheritance as requirements for CONSYDERR. In order to further our understanding of this topic, in this section, a complete analysis of inheritance reasoning in CONSYDERR will be presented. To put this analysis into perspective, some background regarding the inheritance problem will be discussed first. Then the solution to the inheritance problem in CONSYDERR will be analyzed.

Since there are various symbolic algorithms already available for dealing with these issues, a question that may arise is: why do we need a different solution, other than to be able to deal with commonsense reasoning involving inheritance with CONSYDERR? The answer to this question is the following:

- It provides a computationally efficient (i.e. massively parallel) solution towards a theoretically and practically important problem.

- More importantly it points out ways for a semantical (vs. syntactical) approach to the problem (or a feature-based approach).

- The inheritance problem is reduced to the more general problem of rules and similarity in this solution, which is provided as a part of the solution to problems of much broader scopes and thus is not an ad hoc tricks, not even a specialized mechanism.

Different solutions for the same problem, to say the least, can shed more light on the problem and lead to better understanding.

6.2.1 Background

Inheritance systems are knowledge representation systems based on hierarchical structuring of knowledge which enables certain useful and customarily made inferences (i.e. inheritance) to be made. This kind of hierarchical structuring allows economical representation and efficient reasoning. It serves as a foundation for building intelligent systems.

There are many different versions of the inheritance problem and its solutions, and most knowledge representation systems embody some inheritance mechanisms. Various implementations exist, ranging from simple ones to highly complex parallel spreading activation systems. It is beyond the scope of this section to review them all here; so we will only briefly look at some of the solutions most directly related to our approach presented here.

Among all the work in this area, Touretzky (1986) is by far the most comprehensive study of inheritance, In this work, inheritance networks are construed to be *lattice* structures, the full mathematical analysis of them is developed thereby, and a clear semantics thus follows for the inheritance problem. This analysis leads to the concept of *the shortest inferential distances* for deducing correct answers, and various kinds of inheritance scenarios are presented to see how each of them can be handled correctly.

Shastri (1988) presents an evidential framework for the inheritance problem and a connectionist implementation of the evidential formulation. His treatment of inheritance is *extensional*, that is, based on the numbers of objects with different property values, and the conclusion is reached with information maximization. However, in my view inheritance is better dealt with *intensionally* as will be explained later. The connectionist implementation is computationally efficient but makes use of several specially designed node types for evidential combinations.

Yager (1989) presents a *possibilistic* treatment of the inheritance problem based on the fuzzy set theory. Basically, results of inheritance are rated with a possibilistic measure, and the highest rated result is adopted. However other apparatuses are also needed, one of which is the ordering of facts from the most specific to the most general. Although the particular solution seems somewhat ad hoc, the general idea of a fuzzy solution to the inheritance problem stands as solid. Mili & Rada (1990) represent a different approach in applying the fuzzy set theory to inheritance and deals with inheritance in the context of generalized fuzzy regularity, that is, hierarchical relationships between concepts reflect relationships between the concepts' properties and vice versa. It treats both top-down and bottom-up inheritance in one framework. However, it tends to be a mathematically motivated treatment, instead of a cognitively motivated one.

Below a treatment of inheritance in CONSYDERR will be presented. It shares some ideas with the fuzzy set theory, such as graded concepts. But it is mainly based on the idea of the two-level dual representation in the CONSYDERR architecture.

6.2.2 Analysis of Inheritance in CONSYDERR

IS-A and HAS-PROPERTY-VALUE

First of all some clarifications of the basic notations involved shall be made.

In our discussion of the property inheritance problem, we will use a slightly different formulation of the problem: instead of all IS-A links, we will have two types of links: IS-A and HAS-PROPERTY-VALUE. IS-A is used here to denote relations between superclass and subclass; HAS-PROPERTY-VALUE is used to denote property values of a class. These two types of relations are fundamentally different. Although mathematically through some transformation one type can be turned into another, the resulting representation is unnatural. [1]

For example, the elephant example can be expressed in Touretzky's formalism as:

elephant IS-A gray-thing
royal-elephant IS-A NOT gray-thing
royal-elephant IS-A elephant

Here "gray things" or "white things" are not natural categories, although one can always make up such categories if so desired. In our formalism, the example can be expressed, more naturally, as

elephant HAS-PROPERTY-VALUE color-gray [2]
royal-elephant HAS-PROPERTY-VALUE NOT color-gray
royal-elephant IS-A elephant

Or equivalently,

elephant COLOR gray
royal-elephant COLOR NOT gray
royal-elephant IS-A elephant

In CONSYDERR, the above example as in the second formulation can be expressed as rules:

$$elephant \longrightarrow color\text{-}gray$$

$$royal\text{-}elephant \not\longrightarrow color\text{-}gray$$

and the relationship between royal elephants and elephants (i.e. the IS-A link or the superset/subset relation) is automatically taken care of in the feature representation in CONSYDERR (the feature set of the superclass is a subset of the feature set of the subclass).

In general, as in the above example, it is a rule of thumb that in CONSYDERR IS-A links are implemented implicitly through CD representations and HAS-PROPERTY-VALUE links are implemented with rules in both CD and CL.

---

[1] Let p represent a particular property, v its value, and c a class. We can make a category out of all categories that has a certain value for a certain property, i.e.

$$c = c_1 \cup c_2 \cup ...... \cup c_n$$

such that

$$p(c_i) = v \quad i = 1, 2, ..., n$$

and

$$p(x) \neq v \quad if \;\; x \neq c_i$$

[2] It means that elephant has a property *color* whose value is *gray*. The same below.

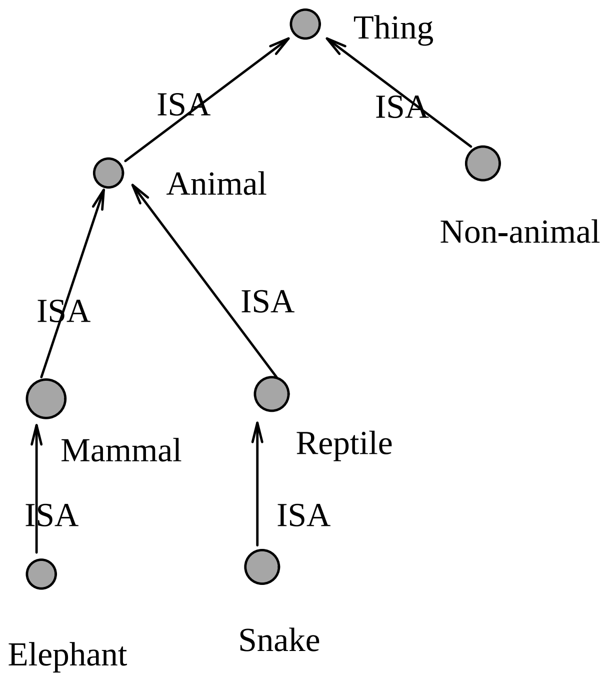


Figure 6.1: An Inheritance Graph with Long Chains of IS-A Links

Chaining of IS-A Links

Now we will analyze how inheritance hierarchies can be formed in CONSYDERR. An inheritance hierarchy is an acyclic graph of nodes representing either a class of objects or a particular property value pair (e.g. color-gray), connected by two types of links (i.e. IS-A and HAS-PROPERTY-VALUE). One important characteristic of inheritance is the chaining of IS-A links, which forms the backbone of an inheritance hierarchy. For example, Figure 6.1 shows an inheritance graph with long chains of IS-A links.

As mentioned before, IS-A links are implemented in CONSYDERR through the set containment relations between the corresponding feature sets for two nodes associated with a IS-A link (see Figure 6.2). Long chains of IS-A links are handled exactly the same way by utilizing feature representation, which speeds up inferences by avoiding tracing through chains of links as in symbolic inheritance systems.

The question now is how property values of a concept can be inherited along this chain upwards or downwards correctly. Below a formal analysis will be presented; those uninterested in mathematical details can skip to the end of this section for the summary.

Let us look at top-down inheritance first. By top-down is meant that the property values of a class are inherited by its subclass (usually directly below the original class in an inheritance graph). It seems easy to accomplish this task: based on similarity between the superclass and the subclass in their corresponding feature representation (one is contained in the other), the subclass can easily inherit the property values. So can the subclass of the subclass, the subclass of the subclass of the subclass, ... and so on. Troubles arise when there are one or more cancellations of the property somewhere along the chain. In such cases, we have to make sure that correct values are inherited down the chain.

Suppose, in an inheritance graph, a is a class and b is its closest ancestor that has a value for a property we want (i.e. $a < b$ and $p(b) \neq \emptyset$, and there is no c such that $a < c$, $c < b$ and $p(c) \neq \emptyset$; where p(x) stands for the value of the property p for the class x, and $<$ denotes IS-A relations). See

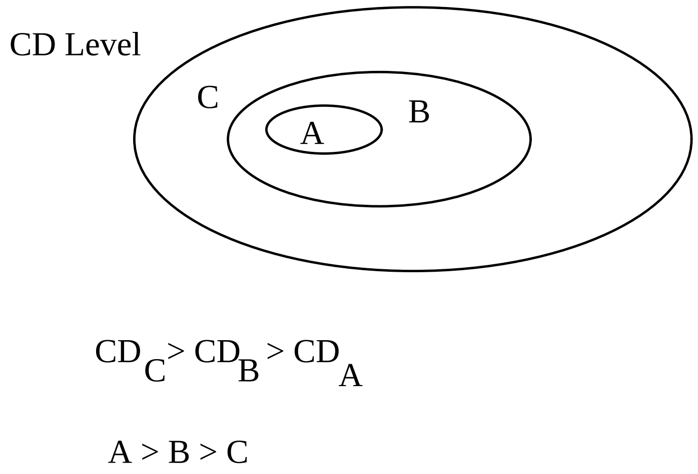


Figure 6.2: Implementing Chains of IS-A Links: $A \supset B \supset C$ and $CD_A \subset CD_B \subset CD_C$

Figure 6.3. To simplify the discussion, we assume there is no bottom-up inheritance. We want to have, regardless of other ancestors of a,

$$p(a) = p(b)$$

Then the question is how we can accomplish this. The only situation we have to look into is that there is d such that $b < d$ and $p(d) \neq \emptyset$. If p(d) =p(b), there is no difficulty at all accomplishing the above goal. If p(d) $\neq$ p(b), the class a has to inherit p(b) not p(d), or in other words, in CONSYDERR the node representing p(b) has to fire more strongly than the node representing p(d) [3]. This is exactly what CONSYDERR will do, given the feature representation in CD. Intuitively, since we have $CD_a \supset CD_b \supset CD_d$ (i.e. the feature set of a is closer to the feature set of b than that of d), class a will inherit b's property values instead of c's. Mathematically, suppose p(b) and p(d) are represented in CD by one node each, according to the formulas specified before (for details, see Chapter 3), [4]

$$p(b) = \sum_{CD_b} \frac{ACT_a}{f(|CD_b|)}$$

and

$$p(d) = \sum_{CD_d} \frac{ACT_a}{f(|CD_d|)}$$

where ACT_a represents the initial activation of a. Since $CD_d < CD_b$, and f is an monotone increasing function (slower than linear), it follows that $p(b) > p(d)$.

Let us now look into bottom-up inheritance. By bottom-up is meant that the property values of a class are "inherited" by its superclass (usually directly above the original class in an inheritance graph). Based on similarity between the superclass and the subclass in their corresponding feature

---

[3] In order to deal with inheritance correctly, the activation of nodes will be allowed to be slightly more than one, so that we will be able to distinguish two different inherited properties in terms of which one is more plausible in a given situation (accomplished by, say, a winner-take-all network). For final output from a system, any activation greater than one will be treated as exactly one.

[4] It is not difficult to come up with the results, when we have the following in mind:

$td_a = 1$
$bu_b = \frac{1}{g(|CD_b|)}$
$lw_{ab} = \frac{1}{f(|CD_a|)}$

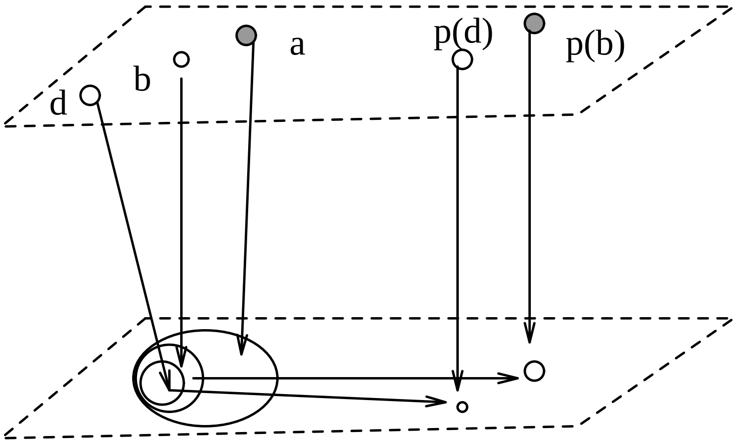


Figure 6.3: Inheritance Case 1
(Arrows in CD represent pairwise links across two sets)

representation (one is contained in the other), the superclass can easily activate the same property values. This inheritance can occur further and further along the IS-A chain just as in the case of top-down inheritance. When there is a cancellation along the chain (or in other words, there are different inheritable property values), we have to make sure that correct values are inherited. Since this case is symmetrical to the top-down case, it is easy to see why CONSYDERR will work. Suppose, in a inheritance graph, a is a class and b is its closest descendant that has a value for a property we want (i.e. $a > b$ and $p(b) \neq \emptyset$, and there is no c such that $a > c$, $c > b$ and $p(c) \neq \emptyset$, where p(x) stands for the value of the property p for the class x, and $<$ denotes IS-A relations). See Figure 6.4. To simplify the discussion, we assume there is no top-down inheritance for the time being. We want to have, regardless of other descendant of a,

$$p(a) = p(b)$$

Suppose there is d such that $b > d$ and $p(d) \neq \emptyset$. If p(d) =p(b), there is no difficulty at all. If p(d) $\neq$ p(b), the class a has to inherit p(b) not p(d), or in other words, in CONSYDERR the node representing p(b) has to fire more strongly than the node representing p(d). Mathematically, suppose p(b) and p(d) are represented in CD by one node each, according to the formulas specified before (for details, see Chapter 3),

$$p(b) = \sum_{CD_a} \frac{ACT_a}{f(|CD_b|)}$$

and

$$p(d) = \sum_{CD_a} \frac{ACT_a}{f(|CD_d|)}$$

where ACT_a represents the initial activation of a. Since $CD_d < CD_b$, and f is an monotone increasing function (slower than linear), it follows that $p(b) > p(d)$.

When there are both top-down inheritance and bottom-up inheritance, there are some complications that have to be analyzed. Suppose a is a class and b is its closest ancestor that has a value for a property we want and c is the closest descendant that has a value for the property, i.e., on one hand $a < b$ and $p(b) \neq \emptyset$, and there is no d such that $a < d$, $d < b$ and $p(d) \neq \emptyset$; and on the other hand $c < a$ and $p(c) \neq \emptyset$, and there is no d such that $d < a$, $c < d$ and $p(d) \neq \emptyset$, where p(x) stands for the value of the property p for the class x, and $<$ denotes IS-A relations as above). If p(c) =p(b), there is no difficulty at all in having the correct property values inherited. If p(c) $\neq$ p(b), a problem

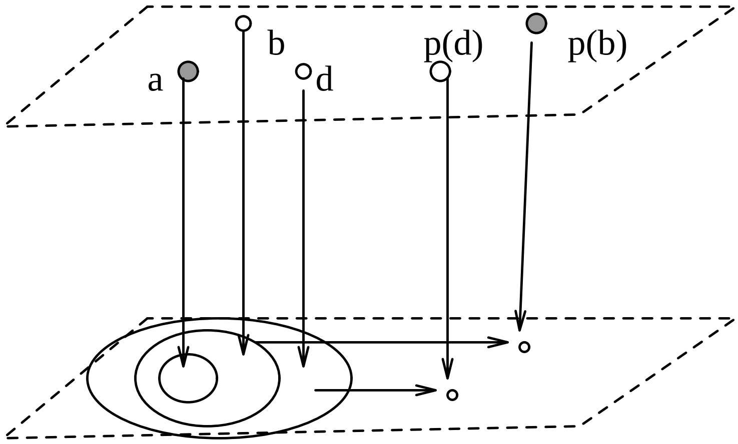


Figure 6.4: Inheritance Case 2

arises. Mathematically, suppose p(c) and p(d) are represented in CD by one node each, according to the formulas specified before (for details, see Chapter 3),

$$p(c) = \sum_{CD_a} \frac{ACT_a}{f(|CD_c|)}$$

and

$$p(d) = \sum_{CD_d} \frac{ACT_a}{f(|CD_d|)}$$

where ACT_a represents the initial activation of a. Since $CD_c > CD_a > CD_d$, and f is an monotone increasing but slower-than-linear function, it follows that the result is indeterminate. This indeterminacy is the direct result of the ambiguity in the knowledge given originally in the problem specification. We will have $p(d) > p(c)$ if and only if $\frac{|CD_d|}{f(CD_d)} > \frac{|CD_a|}{f(CD_c)}$. Intuitively, to inherit from top-down, the feature set of the superclass should be as close to the feature set of a as possible; and to inherit from bottom-up, the feature set of the subclass should be as close to the feature set of a as possible.

In the above discussion we assume that each property value is represented by one simple node in CD. What if those property values form a hierarchy too? In case of top-down inheritance, suppose $a < b < d$ and p(b) and p(d) are represented in CD by feature sets, and $CD_{p(b)} > CD_{p(d)}$ (i.e. p(b) is a subset of p(d)) [5], when a is activated, according to the formulas specified before [6],

$$p(b) = \sum_{CD_{p(b)}} \frac{\sum_{CD_b} \frac{ACT_a}{f(|CD_b|)}}{g(|CD_{p(b)}|)}$$

and

$$p(d) = \sum_{CD_{p(d)}} \frac{\sum_{CD_b} \frac{ACT_a}{f(|CD_b|)}}{g(|CD_{p(d)}|)}$$

where ACT_a represents the initial activation of a. Since $CD_d \subset CD_b$ and $CD_{p(d)} \subset CD_{p(b)}$, and g is an monotone increasing function (slower than linear), it follows that $p(b) > p(d)$. So the result is correct. See Figure 6.5.

[5]It cannot be otherwise. cf. Touretzky (1986).

[6]In this case, those nodes in $CD_{p(d)}$ receive inputs based on two different rules: $b \longrightarrow p(b)$ and $d \longrightarrow p(d)$, and the former is stronger than the latter. We use MAX in combining results from different rules.

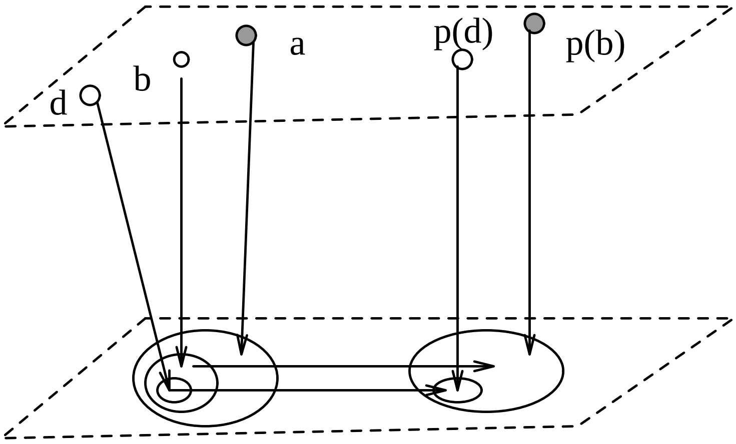


Figure 6.5: Inheritance Case 3

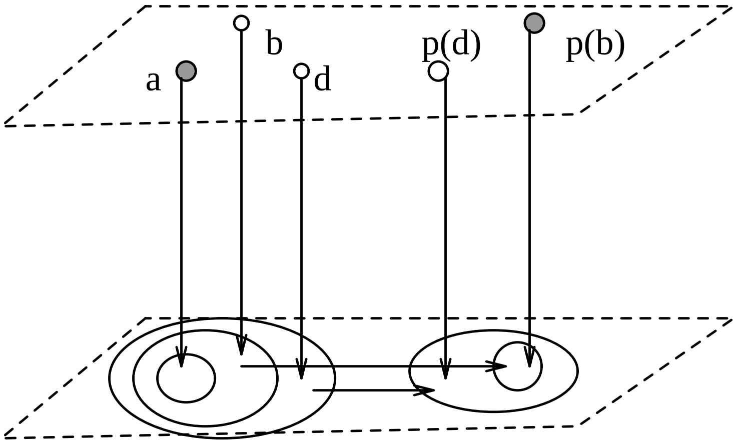


Figure 6.6: Inheritance Case 4

In case of bottom-up inheritance, suppose $a > b > d$ and p(b) and p(d) are represented in CD by feature sets, and $CD_{p(b)} \subset CD_{p(d)}$ (i.e. p(b) is a superset of p(d)) [7], when a is activated, we have [8]

$$p(b) = \sum_{CD_{p(b)}} \frac{\sum_{CD_a} \frac{ACT_a}{f(|CD_b|)}}{g(|CD_{p(b)}|)}$$

and

$$p(d) = \sum_{CD_{p(b)}} \frac{\sum_{CD_a} \frac{ACT_a}{f(|CD_b|)}}{g(|CD_{p(d)}|)} + \sum_{CD_{p(d)}-CD_{p(b)}} \frac{\sum_{CD_a} \frac{ACT_a}{f(|CD_d|)}}{g(|CD_{p(d)}|)}$$

where ACT_a represents the initial activation of a. Since $CD_d < CD_b$ and $CD_{p(d)} < CD_{p(b)}$, and f and g are monotone increasing functions (slower than linear), when proper functions are used for f and g (as discussed in chapter 3), we have $p(b) > p(d)$. So we obtain the correct result. See Figure 6.6.

Having analyzed all the essential cases that guarantee the correct working of inheritance, we can be certain now that CONSYDERR provides the right mechanism for performing inheritance. The

[7] It cannot be otherwise, the same as before.

[8] In this case, those nodes in $CD_{p(d)}$ receive inputs based on two different rules: $b \longrightarrow p(b)$ and $d \longrightarrow p(d)$, and the former is stronger than the latter. We use MAX in combining results from different rules.

CONSYDERR architecture provides an "implementation" of inheritance which is not explicitly path-based inheritance, but merely rules and feature similarity. The synergy between the two different types of representation (and the use of intensional representation that makes this possible) in a way "eliminates" the inheritance problem outright. And this solution, or "elimination", is provided as a part of the solution to problems of much broader scopes. In fact, the problem of inheritance is reduced to the more general problem of rules and similarity.

This analysis also shows that CONSYDERR provides a computationally efficient solution to inheritance, because all inferences are done in parallel and there is no long chain to be traced.

We might also want to examine the expressive power of this solution compared with other systems. We can look at examples from Touretzky (1986), and it is relatively easy to see that all of these examples can be handled by CONSYDERR. (Because of the space limitation, the discussion will not be included.)

Emergent Concepts

For most reasoning systems, an inheritance hierarchy is fixed in its structure, that is, it has nodes, links, etc. as given initially, and it can only infer what is put in there. However for many intelligent systems, some sort of a generative capability is very useful or even crucial (cf. Pustejovsky 1991). And new concepts and categories should be able to be formed and generated on the fly, when the need for them arises.

For example,

> If carrying cargo, buy utility vehicles. If carrying passengers, buy passenger vehicles. If carrying both cargo and passengers, what shall one buy?

What we want is for a system to come up with a solution such as "van", which is both a passenger vehicle and a utility vehicle, without the need for a rule to take care of the particular situation.

In CONSYDERR, certain new concepts can be generated dynamically (though in a limited way) through the interaction in representation. From this perspective, it is not just a straight "implementation" of the same good old inheritance theory, but seems to represent a new approach. Analyses are presented in the next section where representational interactions are studied.

Multiple Inheritance

Multiple inheritance is a difficult problem for all inheritance reasoning systems, which intensional approaches might help to deal with. When different answers are obtained from inheritance through different paths, an inheritance system is in a ambiguous situation. CONSYDERR can provide a way of resolving ambiguity in multiple inheritance based on feature similarity, that is, deciding which inheritance path has the strongest strength, taking into consideration both similarity of feature sets and rule weights.

In general, the property values of a class are always inherited from a class that is the closest to it in terms of features, when there is no direct information available regarding its property values (assume all HAS-PROPERTY-VALUE links have the same weights). Suppose that $A > C$ and $B > C$, and $A \longrightarrow E$ and $B \longrightarrow F$ (A and B are both superclasses of C, and each has a different property value E or F). In CD, $CD_A \subset CD_C$ and $CD_B \subset CD_C$. Assume E and F are each represented by one single node in CD. Let us further assume that $CD_A > CD_B$, i.e. A is closer to C in feature representation. Now if C is activated, after a cycle we have

$$E = \sum_{CD_A} \frac{1}{f(CD_A)}$$

and

$$F = \sum_{CD_B} \frac{1}{f(CD_B)}$$

where f is a monotone increasing function slower than linear. So E is activated more strongly than F. Further work is needed to explore this approach.

In summary, this section analyzes how inheritance can be dealt with in CONSYDERR, in terms of implementing a complete hierarchy (for either top-down inheritance or bottom-up inheritance, or both together), dealing with multiple inheritance, and forming new concepts. The results so far indicate that CONSYDERR is well-equipped to handle this aspect of reasoning.

6.3 Analysis of Interactions in Representation

In this section, we will study the problem of *representational interaction* [9] in CONSYDERR. First some background is presented, and then we will analyze various kinds of representational interactions in CONSYDERR.

The analysis shows that in CONSYDERR rules are no longer context-free pieces of knowledge: instead rules form complex, interacting processes. Interaction in representation during reasoning processes can result in the generation of new concepts and categories, which can be useful in filling in the knowledge gaps a system has.

6.3.1 Background

Many researchers point out that there are severe problems associated with pure rule (or logic) based models of reasoning: [10]

- Israel (1977) in his discussion of what is wrong with default logic argues that real inferences are done by belief revision and fixation, not by pure syntactical rules of inferences. As he puts it:

  > "[It is not] the case that if one is justified in believing that p and justified in believing that if p then q, is then one justified in believing that q. Unless, of course, one has no other relevant beliefs. But one always does."

- Perlis (1986) continues the same theme. He distinguishes the two different approaches: one is to seek a specification of formal rules, and the other is to seek to identify the process of drawing conclusions. (In his original discussion, he only refers to default reasoning, but the same idea applies to all reasoning.)

- Johannessen (1990) points out in his discussion of rule-following behavior:

  > "There must be a non-interpretive way of grasping a rule. On the basic level, rules are simply followed. That is the point of saying that practices, the established ways of believing, giving words their meaning. Hence rules get their identity from the practices in which they are embedded. To understand a rule is to master the corresponding practices in which it is 'inscribed'. The established practices are not isolated monads; they are essentially interrelated."

[9] Here it is defined to be the interaction between different concepts or rules due to their overlapping representations.

[10] Here the word *rule* is used to refer to *forms* rather than *contents*.

There are some attempts at dealing with this problem of context-freedom of rules and resulting inconsistency/incompleteness. A cursory survey shows a number of efforts; for example, Wilkins & Tan (1988) deal with inconsistent, incomplete, and incorrect rules in a rule base through the use of learning methods to fix the rule base; Duval & Kodratoff (1986) deal with inconsistent databases through introducing *modulation*, a kind of quantifiers, in rules.

In this section we will try to deal with the inconsistency and incompleteness through representational interactions, i.e., treating reasoning as a process, not applications of isolated rules. We will analyze the issue of interaction, and try to answer the following questions:

- How is interaction made possible?
- How should we combine results from interaction?
- How may the interaction help to deal with inconsistency and incompleteness in rules?

Cases of interaction include those between conditions in a rule, between conditions and conclusions in a rule, between different rules regarding the same conclusion, and between conclusions reached by different rules. A lot more work beyond what is presented here is needed to fully understand this issue.

6.3.2 How is Interaction Possible?

Two Sources of Representational Interactions

Representational interaction is made possible, first of all, by the massively parallel characteristics of the CONSYDERR architecture, which differs from its typical symbolic counterparts, and secondly, also by the overlapping representation in CD.

In typical symbolic rule-based systems (or production systems; cf. Klahr et al 1986, Posner 1989, and Anderson 1983), a *control structure* is devised so that a system can decide which rules are eligible to fire, how rules are picked for actual firing, and in which order they are supposed to fire. Usually, conditional parts of rules are matched against data in a *working memory*; those rules whose conditional parts match the working memory data are eligible to fire. Conflicts are resolved by ordering rules and firing the rule that comes first in the order. Other policies are also used in various systems, for example, the policy of refractoriness for avoiding a high-priority rule being executed repeatedly with the same data, or the policy of giving preference to rules that match the data most recently created, or the policy of giving preference to rules with stronger conditions, etc. In more recent systems, *goal symbols* [11] are used to control the order and to focus on relevant rules.

The main problem with this symbolic rule-based approach is that, by restricting a system to fire one rule at a time (even in parallel systems, only rules that are not in conflict with each other are allowed to fire at the same time), we miss out on the opportunity for combining results from different rules to fill in knowledge gaps that might exist in a system, or in other words, the interaction between rules that enables the generation of new concepts and categories is not utilized.

Interaction can be made possible, in part, by allowing conflicting (and/or non-conflicting) rules to fire simultaneously and then obtaining the combined results from their interaction. In CONSYDERR inference steps are not carried out one rule at a time as in symbolic production systems, or by any other symbolic strategies such as those used in mathematical logic (cf. Lee & Chang 1973).

[11] A goal symbol is a symbol that can be created by a rule, and thereafter only rules that have that goal symbol in their conditional parts are allowed to fire, until new goal symbols are created. This way attention is focused on only those rules that are most relevant to the current situations, and also an order in which rules are picked to fire can be created dynamically.

Inference in CONSYDERR amounts to a parallel breadth-first search (approximately) as described by the following equations (from Chapter 3):

For the top-down phase,

$$x_i(t+1) = \max_A f(A(t))$$

where A is any node in CL that has $x_i \in CD_A$.

For the settling phase,

$$\Delta A = \alpha \sum W_i I_i(t) - \beta A(t)$$

and

$$\Delta x_i = \mu \sum w_i i_i(t) - \nu x_i(t)$$

where W_i, w_i are link weights (that represent strength or weights in FEL rules), and I_i, i_i are the activation of related facts (conditions or premises).

For the bottom-up phase,

$$B(t+1) = \max(B(t), \sum_{x_i \in CD_B} g(x_i(t)))$$

where B is any node in CL.

Inferences are made in the settling phase, which is highly parallel: all steps that can be explored at a given time are explored together at the same time. According to the differential equations used, activations are propagated towards all post-link nodes in parallel (as in a parallel breadth-first search), and then towards successive nodes, and so on. [12] Thus, with this highly parallel process, different rules interact with each other when their conditions or results overlap.

Overlapping similarity-based representation (with features) in CD introduces another kind of interaction. The CD representations of different concepts may overlap with each other and thus, if activated at the same time, may cause some nodes in CD to become activated more strongly and some others less strongly. The resulting CD activation will then go up to CL during the bottom-up phase. The results of this whole process of interaction, as will be analyzed, are new concepts generated that are some kinds of logical or illogical combinations of the original concepts participating in interaction. This characteristic is even more different from typical symbolic systems than the massively parallel simultaneous firing analyzed earlier: it is not mere syntactical transformation, but meaning-oriented, intensional (i.e. feature-based) reasoning.

Relationships between CL and CD

To understand better interactions involving CD, let us look into the relationships between the corresponding CL and CD representations, given the foregoing analysis regarding representational interactions.

First of all, the relation between a CL node (a concept) and its corresponding CD representation (the feature set, or intension) is not that of necessary and sufficient conditions. According to common logic theories (cf. Leonard 1967), the set of features of a concept that are common and jointly

[12] The only difference from a real breadth-first search is that before the activation of a node is stabilized, successive nodes are also activated, though to a lesser degree. For example, given A and $A \longrightarrow B$ and $B \longrightarrow C$, before B reaches its full activation, C is already activated somewhat. But because of the fact that at equilibrium the differential equations used are equivalent to algebraic equations (see Chapter 3 for an explanation), the above difference is insignificant.

No.	Types	Description	Logical Equivalence	Activation
1	Overlapping	$CD_A \cap CD_B$	$\supset A \vee B$	stronger
2	Additive	$CD_A \cup CD_B$	$A \wedge B$	stronger
3	Subtractive	$(CD_A - CD_B) \cup (CD_B - CD_A)$	no logical equivalence	weaker
4	Subtractive	$CD_A - CD_B$	no logical equivalence	weaker
5	Subtractive	$CD_B - CD_A$	no logical equivalence	weaker

Figure 6.7: Different Types of Interaction Between Two Conditions

peculiar to the concept, called *totally contingent intension*, is necessary and sufficient to the concept (by definition), but it is most likely infinite and hard to come by. Subsets of it, if consisting of only *necessary* members (excluding *merely contingent* ones; cf. also Brachman 1985) of the total contingent intension, are called *total strict intension.* The smallest subset of a total strict intension from which all other features can be deduced is called *definitive intension* or simply *definition* (whose members are naturally *definitive* features). Unfortunately, such a set may not exist; as a matter of fact, it does not exist for most of the concept in the commonsense world (cf. Leonard 1967 and Lakoff 1986).

So what does it mean to have a set of features in CD as a representation of a concept, as done in CONSYDERR? Recall our discussion in Chapter 5 regarding inexactness: most features, like concepts, are vague, fuzzy or probabilistic, so the relationships between a concept and its features are likewise inexact and evidential; that is, a concept is *likely* to have certain features, and a certain set of features is *likely* to determine a concept. Therefore, instead of dividing members of a feature set into the three classes: mere contingent, necessary, and definitive, as in logical theories, we view features as *indeterministically definitive*, lending evidential support for invoking a certain concept, and vice versa. This idea can lead to ideas such as giving different weights to members of a feature set, to correspond to their relative strengths of evidential support, or the idea of a concept having multiple feature sets, to allow for complex disjunctive concepts, etc. (These ideas will not be investigated in this thesis, but are identified as future work).

When similarity matching is involved, that is, when a feature set of a concept is activated and it shares some features with other concepts, the concepts activated by similarity have only a part of its feature set involved. Thus conclusions regarding these concepts are highly uncertain, because the relation between a concept and a subset of its feature set is even more indeterministic. This is in a way analogous to *indexical intension* (cf. Leonard 1967), which is a set of features that does not define a concept but only provides some clue to its presence. Clearly, interactions involving CD produce uncertain, but plausible conclusions, which, for lack of better evidence, can be useful although never fool-proof; this is exactly what commonsense reasoning is supposed to be like.

In the rest of this section we first analyze various cases of representational interactions in an abstract way, and then we show, by example, how interaction and the generative capability can help to deal with inconsistent/incomplete knowledge.

6.3.3 Interaction Among Conditions

Various concepts (e.g. conditions in the LHS of the same rule) can interact with each other to produce some effects that neither of these conditions alone can produce. Suppose we have two conditions in the LHS: A and B (as in the rule $AB \longrightarrow C$), each of which has a corresponding representation in CD. There are several different kinds of interaction that may occur from their interaction, as shown in Figure 6.7.

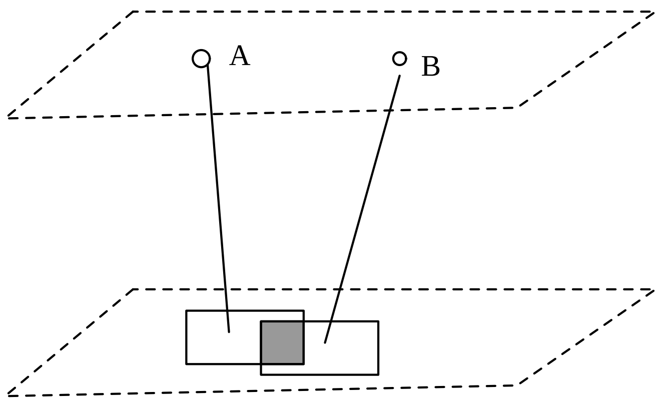


Figure 6.8: The First Type of Interaction: Overlapping
(Shaded areas represent new concepts)

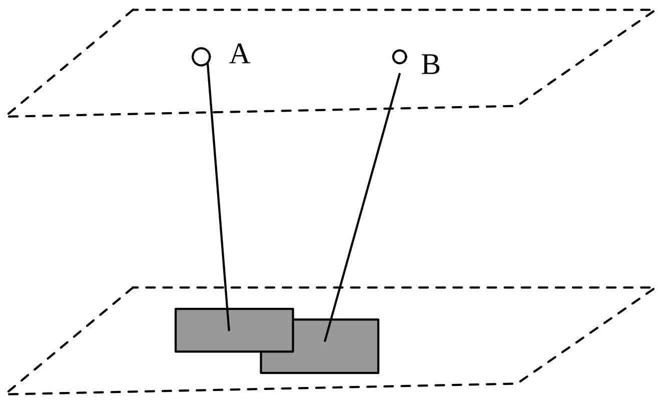


Figure 6.9: The Second Type of Interaction: Additive

According to this table, depending on the relationship between the two feature sets associated with the two concepts, different kinds of concepts can emerge from the interaction, which could be the logical OR or the logical AND, or something else, of the two initial concepts (provided that feature sets involved form totally contingent intensions). The first case is depicted in Figure 6.8. The second case is depicted in Figure 6.9. The third case is depicted in Figure 6.10. The fourth case is depicted in Figure 6.11. The fifth case is depicted in Figure 6.12. Note that the resulting activation of the new concept will be stronger than the two initial concepts (in their respective CD representations) in the first two cases, and weaker in the last three cases.

6.3.4 Interaction Between Conditions and Conclusions

There are two basic types of interaction between the conditions and the conclusion of the same rule: additive interaction and subtractive interaction. Figure 6.13 shows the two cases.

In case of the additive interaction, the condition and the conclusion have overlapping CD representations (i.e. they share some features), and the condition has positive influence on the conclusion (i.e. the link weight is positive). Thus in CD there could be a positive self loop, which will lead to saturation at the highest possible value in these nodes which are in the intersection of the two CD representation. To avoid this, as a general policy, we will not allow any self loop, or links originating and terminating at the same node. This way the nodes in the intersection remain the same in terms of activation levels as before the rule application (or as without the rule).

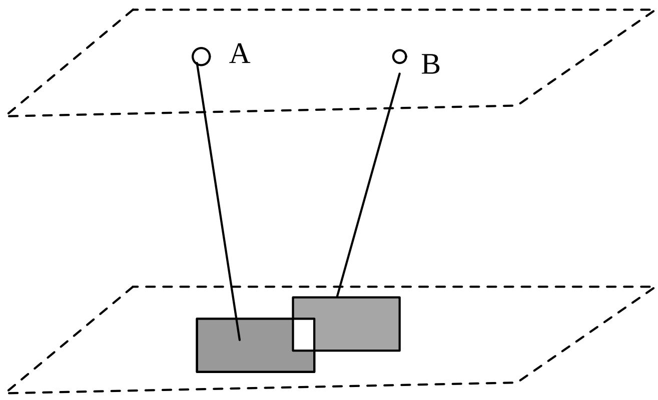


Figure 6.10: The Third Type of Interaction: Subtractive

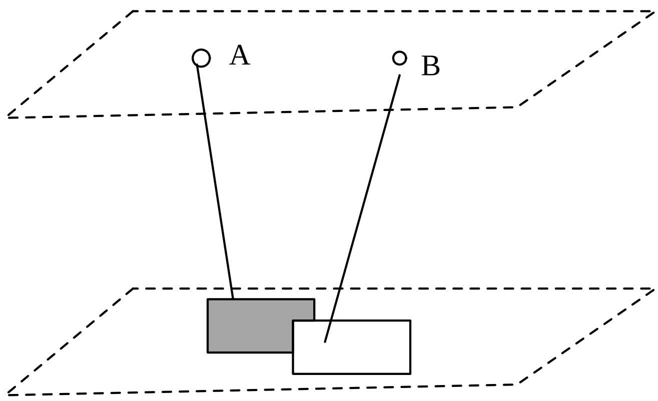


Figure 6.11: The Fourth Type of Interaction: Subtractive

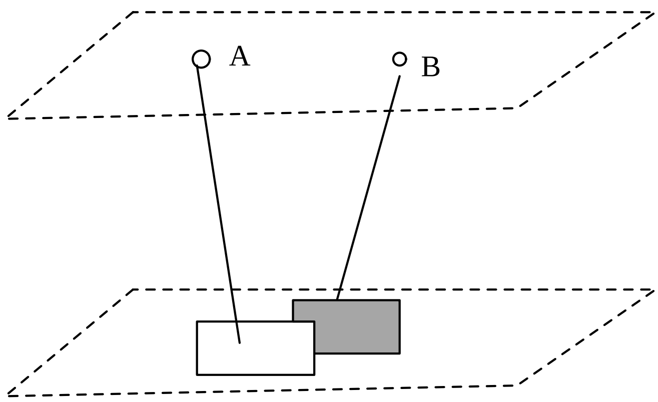
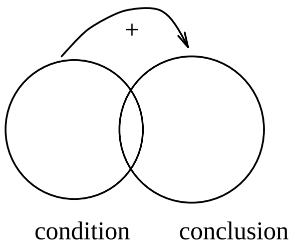
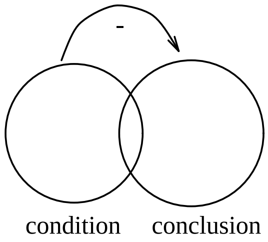


Figure 6.12: The Fifth Type of Interaction: Subtractive



(1) Positive interaction



(2) Negative interaction

Figure 6.13: Two Types of Interaction Between Conditions and Conclusions

In case of the subtractive interaction, the condition has negative influence on the conclusion, and the condition and the conclusion have overlapping CD representations. As before, to avoid negative self loop, we will eliminate all links from and to the same node. The nodes in the intersection of the two CD representations will remain the same as if there is no rule.

6.3.5 Interaction Among Conclusions from Different Rules

The interaction among conclusions reached by different rules is the phenomenon we are most interested in. For interactions of the same conclusion (with different values) reached from different rules, seven different cases can be distinguished (see Figure 6.14). When the same conclusion is reachable from different rules with different values, there is a need to combine the results. There are several ways of combining the values of the same conclusion reached from different rules: ADD, MAX, or combined ADD/MAX, etc. We tabulate them for all the seven types of interactions, in order to compare them and find out which one is more reasonable (see Figure 6.15). The results listed are activations of nodes in the intersection of two CD representations. Among these combination operations, MAX and ADD are simply performing the corresponding operations on the two values; ADD/MAX is an operation which obtains the maximum of the two values when they are of the same sign (representing the confidence when given two pieces of evidence leading to the same conclusion) and adds them when they are of opposite signs (representing the reduced confidence when confronted with contradictory evidence); Heckerman's rule (Heckerman 1986) is a modified version of the MYCIN combination rule (and considered by Shultz 1989 to be the best among all the methods tested), which calculates the sum of the two values divided by 1 plus their product:

$$H(a,b) = \frac{a+b}{1+ab}$$

Let us analyze each combination operation. [13] ADD will produce incorrect results — the results may well exceed the upper limit 1 in case of additive interactions. MAX works fine with additive

[13] Following these combination operations, it is not difficult to come up with the results as contained in the table,

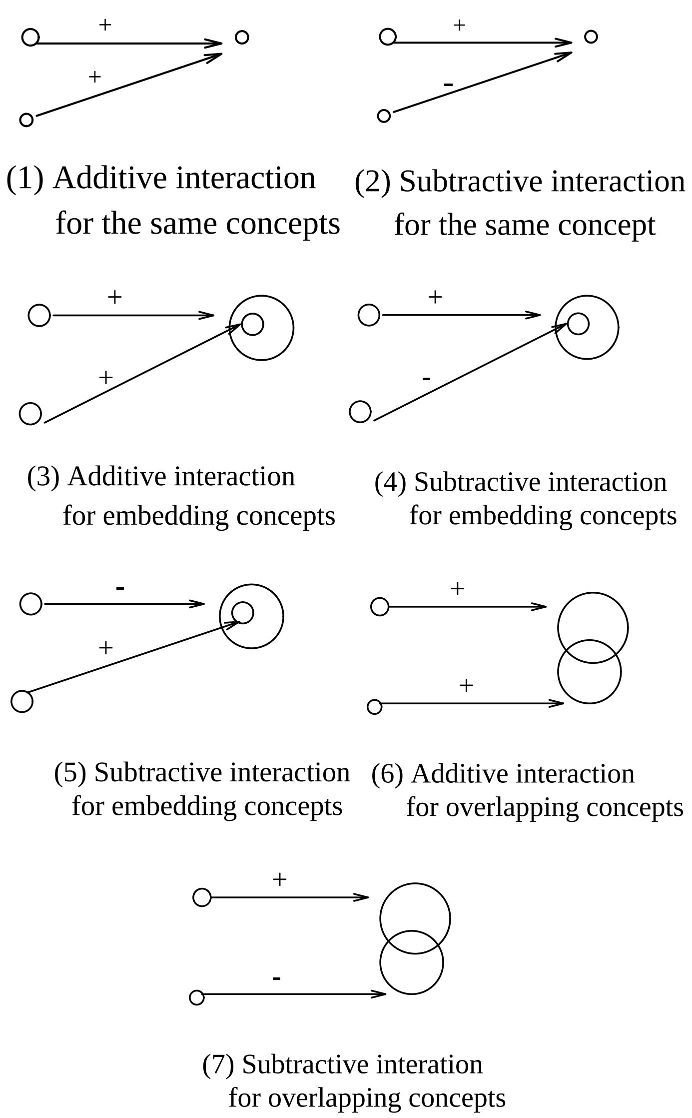


Figure 6.14: Interaction Between Conclusions from Different Rules

Types	MAX	ADD	ADD/MAX	Heckerman
additive, the same concept	max	sum	max	stronger
subtractive, the same concept	max	0	0	0
additive, embedding concepts	max	sum	max	stronger
subtractive, embedding concepts	max	0	0	0
subtractive, embedding concepts	max	0	0	0
additive, overlapping concepts	max	sum	max	stronger
subtractive, overlapping concepts	max	0	0	0

Figure 6.15: Different Operators for Dealing With Interaction Between Conclusions from Different Rules

(Each entry indicates the resulting activation of a CD node in the intersection)

interactions, but in case of subtractive interactions, it has some problems, that is, when the two values are of opposite signs but their absolute values are the same, the result is indeterminate, which results in a random decision with means to be 0 (statistically, half the results are positive and half are negative). This is fine, until we look into the situation where the two values are of the opposite sign and their absolute values are very close but not the same, in which case the result will be the same as one of the values, as if the other value did not exist, whereas the semantics of the problem requires the result to be close to zero. In all of these cases, Heckerman's rule gives proper results, but there is one problem: it will destroy the inheritance and cancellation property we worked hard to establish.

From this analysis, it is evident why ADD/MAX is preferred over ADD or MAX for combining results from different rules. In implementations, the ADD/MAX operation is carried out in a node by a node activation function when combining results from multiple sites. It should be pointed out, however, that if rules and facts are *consistent* (as defined in the previous chapter), the MAX operation will be sufficient, because in this case MAX is equivalent to ADD/MAX (due to the fact that the situation where two values of the same conclusion are of different signs will never occur if a theory is consistent). By now we answered one of the lingering questions we had before regarding node activation functions in specifying the details of the CONSYDERR architecture (see Chapter 3).

Different conclusions reached from different rules at the same time can interact too. It is similar to the interaction among different conditions of a rule; the same sort of analysis as in Subsection One regarding logical characterization of the resulting concepts can also be performed; the concept that emerges from the interaction can be the logical AND, the logical OR, or something else of the two original concepts involved in the interaction; readers are referred to Figure 6.7 in the first subsection. When there are more than two concepts involved in interaction, the situation could be much more complicated, and logical equivalence could be difficult to identify. Nevertheless, the basic approach taken here is still applicable.

The above analysis of representational interaction in CONSYDERR reveals an interesting phenomenon: the generative capability of the CONSYDERR representation. The representation is

when we also have the following in mind:

$$td_a = 1$$
$$bu_b = \frac{1}{g(|CD_b|)}$$
$$lw_{ab} = \frac{1}{f(|CD_a|)}$$

capable of generating new concepts and categories dynamically. What it can do is not limited to what was stored there explicitly and is not limited to the logical closure of its explicit representations. For example,

> If carrying cargo, buy utility vehicles. If carrying passengers, buy passenger vehicles. If carrying both cargo and passengers, what shall one buy?

In this example, we will utilize the additive interaction (as analyzed before). There are two rules in the system:

$$\textit{carrying-cargo} \longrightarrow \textit{utility-vehicles}$$

$$\textit{carrying-passenger} \longrightarrow \textit{passenger-vehicles}$$

And different types of vehicles are represented as features in CD. When the above two rules are both activated (in response to the question), all features corresponding to both utility and passenger vehicles will be activated in CD. All this information will go up to CL, and the things corresponding to the intersection of utility and passenger vehicles will be activated strongly (because they have all the features). So nodes in CL representing something like "van" will win, although there is no explicit rule regarding them.

Another example is for the subtractive interaction:

> A particular plant produces all kinds of passenger vehicles, but none of utility vehicles.

In this example, there are two rules:

$$\textit{plant} \longrightarrow \textit{passenger-vehicles}$$

$$\textit{plant} \longrightarrow \neg\textit{utility-vehicles}$$

When the node "plant" is activated, through the application of the above two rules, "passenger-vehicles" is positively activated and "utility vehicles" is negatively activated. In CD, their corresponding representations are activated too. So at the bottom-up phase, "passenger vehicles" and all other nodes representing some kinds of passenger vehicles are positively activated; "utility vehicles" and all other nodes representing some kinds of utility vehicles are negatively activated; and all nodes representing vehicles that are both passenger and utility vehicles (so they have all the features in their CD representation) have zero (or near zero) activation, because those features that are related to passenger vehicles send positive activation up and those features that are related to utility vehicles send negative activation up.

It is clear from this analysis that in CONSYDERR there is no more *fixed* inheritance hierarchy there. A hierarchy can change dynamically, through similarity-based representation in CD and interactions between concepts and between rules. As demonstrated by the above two examples, and the analysis summarized in Figure 6.7, new categories and concepts can be generated at various places in the hierarchy, from the interaction of feature sets of two concepts at a certain level of a hierarchy. Besides interaction among concepts at the same level of the hierarchy, two concepts at different levels can also interact and generate new concepts, although their final positions in the hierarchy are dependent on several factors and thus difficult to determine. [14]

Interactions can be utilized to deal with incomplete and inconsistent knowledge bases, as well as to provide a compact and flexible representation. The above examples show how to have inconsistent and incomplete rule bases produce consistent and correct results; the key is the *interaction* that helps to generate consistent and complete conclusions out of inconsistent and incomplete rule bases.

[14] A full analysis is beyond the scope of this thesis.

CONSYDERR provides a possible way out of the problem of inconsistent/incomplete rules, one aspect of the brittleness problem as identified before (though a lot more work is needed to explore this possibility).

In some sense, reasoning is not done by isolated rules in CONSYDERR. Statically it is made of individual rules; but dynamically it is a complex, interacting process. One rule does not operate in isolation: related rules can fire at the same time and they interact with each other. Reasoning is accomplished through this complex process.

To conclude, this section shows that various kinds of interaction can occur in representation, which produce some useful generative capabilities. In CONSYDERR, although knowledge is described by rules, reasoning can be viewed as a complex process instead of an oversimplistic picture of rule-following.

6.4 Knowledge Acquisition, Learning and Adaptation

One of the major problems in building an intelligent system is how to acquire all the necessary knowledge for a system. The difficulty lies in that, in order to build systems that can scale up, the method of acquiring knowledge must be somehow systematic and preferably automatic, and moreover the acquired knowledge has to be organized in a way so that it can be revised and updated easily. This difficulty is currently insurmountable with existing approaches or methodologies.

There are nevertheless many ways for ameliorating the situation by using some partially successful techniques, coupled with some clever software engineering. Below we will look into some of these techniques and discuss very briefly their relevance to CONSYDERR.

Knowledge acquisition research produces techniques and software systems that help with the usually tedious and even painful process of obtaining knowledge manually. Marcus (1989) contains a bird's eye view of the field. Potentially, many of the techniques developed in that field can be useful in setting up connectionist knowledge-based systems such as CONSYDERR.

Learning serves as another alternative for system building. With automated procedures, intelligent systems can hopefully be developed with minimal human intervention. Learning algorithms typically extract knowledge from data, including input and sometimes output of a supposedly ideal intelligent system, or they develop their own knowledge and representation through interacting with the external world. Some of these algorithms are undoubtedly useful for building systems based on the CONSYDERR architecture. For example, Neal (1990) presents statistical learning techniques for developing feedforward multi-layer connectionist networks. Although designed for different purposes, the algorithm can be easily adapted to CONSYDERR for setting up rules. A similar algorithm is the IARL algorithm (see Sun & Waltz 1991), which learns rules but not confidence parameters (used to determine activations of rules in CONSYDERR).

Looking at connectionist learning algorithms, One of the most famous is *back-propagation* (Rumelhart et al 1986), which, despite the fact that there is no explicit rule in networks, can learn rule-like behavior (to a certain extent at least). But it is excruciatingly slow, and therefore it is not very likely that it will scale up well. Gallant (1988) presents the Pocket Learning Algorithm, a generalized perceptron learning algorithm for building up connectionist networks from data presentations, one obvious advantage of which is that one can extract rules from the developed network, and therefore make the representation more systematic and explicit.

Evolution or adaptation algorithms can be used to improve system performance on the fly. For example, Holland et al (1986) presents the best known of these techniques, *the genetic algorithm*, for creating new rules and eliminating old ones. These algorithms can potentially be very useful, if adapted to CONSYDERR, for updating, fine-tuning, and revising rules in this type of connectionist networks. Another useful kind of adaptation we can perform in CONSYDERR involves a transformation which turns similarity matching into separate, specialized rules: when we keep deriving the same result based on similarity matching or combined similarity matching and rule application, we can then establish a rule specifically for this inference to facilitate the future performance.

With this brief overview in mind, now let us see how we actually acquire knowledge in CONSYDERR currently. Basically, we try to do it the simplest way possible, and the adoption of many of these above techniques remains to be done in the future. There are basically two questions to be answered for CONSYDERR:

- How do we obtain rules?
- How do we obtain similarity-based representation?

These two questions can be, although not necessarily, dealt with separately.

Rules can generally be obtained from domain experts, textbooks, manuals/instructions, or by using learning algorithms (for example, back-propagation like gradient-descent algorithms). There is no fixed, universally applicable way to do this, or in other words, it is domain-specific. In GIRO, rules are obtained by going through geographical sourcebooks, picking out the relevant information and integrating it into the network with the CFRDN procedure (see Appendix) manually. The rules being put into the system include *Western-Texas is cattle-country*, etc.

Similarity measures can be obtained by an indirect means. We can first identify all the relevant features for the concepts involved. Then clearly the amount of feature overlap determines the similarity between concepts involved. In order to come up with detailed feature representations, we pre-establish a set of feature nodes and then go through sourcebooks, establishing links between a concept in CL and its features in CD, based on what we obtain from the sourcebooks. The features in GIRO include: altitude, rainfall, vegetation, population, temperature, terrain, etc. with various ranges. Another possible way of obtaining similarity measures is to conduct a test, asking a group of subjects to rate the similarity of concepts concerned and then calculating the similarity measures based on the collected test scores with the STSIS procedure (see Appendix). Another possible way is to have the system to develop its own feature representation through perception or interactions with the external world, by using back-propagation or other connectionist learning algorithms (cf. Dyer 1990 for some experiments in doing that). The feature representation developed this way may not be interpretable as in GIRO, but may be better at capturing the natural similarity in concepts realistically.

It should be emphasized that acquiring knowledge in GIRO is accomplished with a rather mechanical process. There is not much of hand-tuning and there is no ad hoc trick (unlike in typical expert systems or some of the early connectionist systems). The process can be incremental, because the architecture is modular: new nodes can be added and the existing ones can be deleted at either level and the effect is localized, without affecting the global structure.

6.5 Summary

This chapter addresses the issues of inheritance, representational interaction (and generative capabilities), and knowledge acquisition.

Regarding inheritance, we show that CONSYDERR is capable of implementing a complete hierarchy (for either top-down inheritance or bottom-up inheritance, or both together), dealing with cancellation, multiple inheritance, and forming new concepts.

Regarding interaction, we show that various kinds of interaction can occur in representation, which produce some useful generative capabilities, and that reasoning is tantamount to a complex process in CONSYDERR.

Various methods for knowledge acquisition are surveyed in the third section. Many possible ways of knowledge acquisition in CONSYDERR are discussed and the methods currently used are explained.

Overall, this chapter touches upon those aspects of CONSYDERR that make it qualitatively different from rule-based or logical approaches in symbolic AI, which relates to the dynamics, not the statics, of the CONSYDERR architecture.

Chapter 7

An Extension: Variables and Bindings

7.1 Overview

But strictly held by none, is loosely bound
By countless silken ties ......
— Robert Frost

In all our previous discussions of CONSYDERR, we intentionally omitted issues relating to variables and variable binding, in order to simplify the discussion. In the same vein, we omitted mentioning predicates and arguments when discussing the logic FEL, so only the propositional version of it is looked at. However, variables are important, or even essential, for connectionist systems, as indicated by many leading researchers (cf. Feldman & Ballard 1982, Pinker & Prince 1988, Fodor & Pylyshyn 1988). The problem of incorporating variable binding into connectionist models is considered to be a difficult problem.

This chapter will thus be devoted to solve this problem, by extending the definitions of FEL and CONSYDERR we outlined before. More precisely, we have the following goals for this chapter:

- Developing a first order version of FEL, which will add more power to this logic.
- Developing mechanisms in CONSYDERR for variables and bindings.
- Developing better understanding of the correspondence between logic and connectionist models (or more specifically, between FEL and CONSYDERR).

The plan for this chapter is as follows: The first section explains the variable binding problem; the second section extends the definition of FEL to the first order case; the third section deals with issues involved in adding variables into the CONSYDERR architecture; and the fourth section introduces some formal accounts of node functions and assemblies for variable binding in CONSYDERR. The Appendix contains some technical details left out in the above discussion.

7.2 The Variable Bindings Problem

In this section, we will look into the *variable binding* problem, a problem that was considered to be a shortcoming or even a serious handicap for connectionist models.

Why are variables important in connectionist models? As in Feldman & Ballard (1982) the following example demonstrates the necessity of having variables: Suppose we have a red circle and a black square, we have to have a way to associate red with circle and black with square; this binding has to occur and cease dynamically when a agent or a system is reasoning with (visual) input data. Variable binding is essential in creating dynamic structures in connectionist networks.

Early research in connectionist symbolic processing and high level connectionist models avoids the variable binding problem by reasoning with one object at a time or by using multiple instantiations, such as in Shastri (1988), Kosko (1988), Blelloch (1986), etc. This approach limits the processing power of the respective systems, and increases greatly the computational complexity. So, in order to enhance the representational and computational capability of CONSYDERR, we need to have variables, and we thus need variable binding mechanisms. It is important to solve the variable binding problem.

Why is it difficult to introduce variables into connectionist models? Recall that in conventional connectionist models, each node performs very simple processing, usually just weighted-sum or similar numerical calculations, and structurally these models are uniform and homogeneous. Therefore, they are inherently structureless and it is extremely difficult to impose any symbolic structures onto them because of the uniformity. The problem of introducing variables into connectionist models amounts to the question of how to add structures to networks without losing too much of that characteristic uniformity and homogeneity and how to enhance the processing power of each processing element to a degree sufficient for symbolic processing while maintaining its simplicity. [1]

A few attempts have been made to solve this problem. Touretzky & Hinton (1985) is the first piece of research that directly confronts the problem. Their system is made up of several modules or subnetworks. With distributed representation in these modules, they are able to match variables against data in the working memory by winner-take-all. The system has very complex structures and is costly in terms of computational resources, but it uses only very simple node functions. It serves as an "existence proof" of solutions to problems of introducing rules, variable bindings and other symbol processing capabilities into connectionist models. Another piece of early work is Smolensky (1987), in which the author proposes using tensor products for variable bindings, with either local or distributed representation. Basically each combination of variables and values is represented in the system by a single processing element, or a group of them, which detects dynamically if the right binding is present. It utilizes simple node functions but highly complex networks.

Recently, Ajjanagadde & Shastri (1989, 1990) present systems that perform backward and forward chaining inference with variable bindings handled by temporal codings of two different types. The later version basically works this way: a node representing a variable produces a temporal activation pattern with a specific frequency which represents a particular binding, when inputs are received that indicate the node should produce that binding. This scheme has relatively simple network structures but needs complex node functions (with temporal properties). Observing the fact that only a temporal frequency is used for representing a particular binding and there is nothing else to it, it follows that this scheme is isomorphic to my solution (to be described later in the chapter; cf. also Sun 1989, Sun & Waltz 1991) by mapping a particular frequency into a numerical value to be passed along. [2] Lange & Dyer (1989) present another solution similar to this.

Besides the basic capability of variable binding, there are many more difficult problems to be solved in order to enable a connectionist reasoning system to possess reasoning power comparable to a sequential system. These issues include unifications, consistency checking, variable generation, functions, or even second order predicates (for more details, see Appendix). Most of these issues are not addressed by the aforementioned work.

Having examined past efforts, in the following sections we will develop a new, better scheme for

[1] My solution developed herein is aimed at devising a network structure as simple as possible while maintaining well-definedness and relative simplicity of node functions at the same time.

[2] With such numerical values my scheme therefore requires little extra processing in each nodes (without the need for temporal properties) while at the same maintain a simple network structure.

solving this problem in CONSYDERR.

7.3 Adding Arguments to FEL

In this section, we will look into the question of how to add variables to FEL, i.e., extending FEL from a propositional logic to a first-order one. The purpose of doing this is to understand better the nature of variables and to explore the consequences of including them in a logic like FEL, so that we will be better prepared when we add variables into the CONSYDERR architecture, since CONSYDERR is, in a way, an "implementation" of FEL.

FEL, like any other propositional logic, has arguments hidden in propositions. For example, "Chaco is cattle country" can be expressed in propositional FEL as

$$Chaco \longrightarrow \textit{cattle-country}$$

This can be easily extended into a predicate version:

$$Chaco(x) \longrightarrow \textit{cattle-country}(x)$$

which means that if x, which is an argument to be bound to something, is Chacos, then x is a cattle country. Other rules and facts we presented before can all be turned into corresponding predicate versions in exactly the same way.

Besides this simple single argument case, we can also deal with multiple arguments in one predicate in a similar way. For example, to express "if one locale is north to another, then that locale may be colder than the other." we use the following predicate rule:

$$\textit{north-to}(x, y) \longrightarrow \textit{colder-than}(x, y)$$

Of course, each of these predicate rules should have weights associated with them according to FEL.

Based on these examples, it seems that the task of turning the propositional FEL into the first order FEL is quite straightforward, so we are ready to describe formally the predicate version of FEL:

Definition 7.1 *A* **Fact** *is an atom or its negation, represented by a letter (with or without a negation symbol), with a fixed ordered set of arguments associated with it and having a value between l and u. The value of an atom is related to the value of its negation by a specific method, so that knowing the value of an atom results in immediately knowing the value of its negation, or vice versa.* For example, a(x), ¬a(u, v), x, u(y), etc. are facts.

Definition 7.2 *A* **Rule** *is a structure composed of two parts: a left-hand-side (LHS), which consists of one or more facts, and a right-hand-side (RHS), which consists of one fact. When facts in LHS get assigned values, the fact in RHS can be assigned a value according to a weighting scheme. Any binding associated with a argument in LHS will be assigned to the arguments with the same name in RHS.*

For example, the rule in the above two examples.

Weighting schemes are defined the same way as before.

We define the notion of a theory again:

Definition 7.3 *A* **Theory** *is a 4-tuple:* $< A, R, W, T >$, *where A is a set of atoms, R is a set of rules, W is a weighting scheme for R, and T is a set of thresholds each of which is associated with one element in A.*

Definition 7.4 *A* **Conclusion** *in FEL is a value associated with a fact, calculated from rules and facts by doing the following:*

(1) for each rule having that conclusion in its RHS, obtain conclusions of all facts in its LHS, or assume some values to be zero if they are unobtainable; and then calculate the value of the conclusion according the weighting scheme;

(2) take the MAX of all these values associated with that conclusion calculated from different rules or given in initial input.

(3) for arguments in the fact, assign correct bindings, if exist, from the corresponding arguments in the condition part of the winning rule (the one having the MAX value).

Now FEL can be defined as follows:

Definition 7.5 *A* **Fuzzy Evidential Logic (FEL)** *is a 6-tuple:* $< A, R, W, T, I, C >$*, where A is a set of atoms, R is a set of rules, W is a weighting scheme for R, T is a set of thresholds each of which is for one rule, I is a set of elements of the form* (f, v) *(where f is a fact with arguments and their bindings, and v is a value associated with f), and C is a procedure for deriving conclusions*

We can use the same proof procedure C as before for this version of FEL. We can also differentiate the predicate FEL into two types: FEL_1 and FEL_2 just as the propositional version, with different ranges for values.

In FEL, unlike in Prolog or other logic programming languages, we do not allow functions to be bound to arguments (it can be done nevertheless; cf. Sun 1991). In other words, there is no function in our universe of discourse. Also, we in principle do not handle recursions; the reason is that we are aiming at modeling commonsense reasoning, not logic programming, so the loss in computational power is more than compensated by the resulting efficiency in reasoning. One the other hand, because of the implementational concerns as will be clear in the next section (mainly the fact that we have to use one node for representing each variable, and there can only be a fixed number of nodes in a network), we have to place an upper limit on the number of arguments allowed in a predicate. This is again a limitation as to the computational power of the logic in general, nevertheless the resulting logic is still sufficient for any particular tasks we are interested in carrying out.

To conclude, FEL is redefined to include arguments for predicates, instead of using propositions. Certain restrictions are imposed due to implementational considerations.

7.4 Adding Variables to CONSYDERR

In this section, the question of how variables are added to the CONSYDERR architecture will be discussed, and some related issues will be addressed in light of what was said regarding FEL.

First of all is the question of how to represent variables. Each variable in each fact (predicate) can be handled by one node. The variable can get bound by an application of a relevant rule that has in its RHS the fact that contains the variable. The binding is done by a numerical value, which represents a particular constant (a binding), being passed along the link (which represents the rule in question) to the node representing the variables, from a node representing a variable having the same name in a fact in the LHS of the rule applied. For example, suppose we have a rule: "if an area grows something, it is an agricultural area." It can be expressed as follows (where A represents "grow", and B represents "agricultural area"):

$$A(x, y) \longrightarrow B(x)$$

and the input is A(a, b) (meaning *area a* grows *b*), where a and b are constants, e.g., a=Florida and b=orange. The binding a is passed to the node representing x, and b to the one representing y in

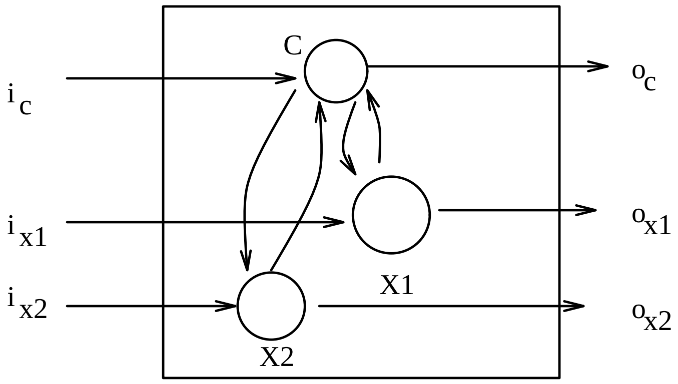


Figure 7.1: An Assembly

A(x,y). Then by applying the rule, a is further passed [3] from the node representing x in A(x, y) to the node representing x in B(x).

Let us look at one level, say CL, to see how this can be done in CONSYDERR. To add variables into CL, the same structure of CL can be kept just as before, but in place of a node now we have an *assembly*, or a collection of nodes interconnected in some way. To serve our purpose of incorporating variables and the variable binding mechanism into the model, an assembly should consist of a certain number of nodes, one of which is for computing and storing confidence values (we will call it the C node), exactly the same as before, and the rest are for variables (we will call them X nodes). As mentioned before, we have to place an upper limit on the number of variables a predicate may have, because we can have only a fixed number of node in an assembly or a network in general. Suppose now we are allowed to have k variables, an assembly then contains k+1 nodes, each of which except one is for storing and passing along bindings of the variable which it represents. Among these nodes, the C node of an assembly receives inputs from C nodes in other assemblies which are conditions (LHS) of a rule that use this assembly for its RHS. Each variable (X) node receives inputs from variable (X) nodes in the same other assemblies as its corresponding C node, picks up its own binding and passes it on. See Figure 7.1. [4]

Taking the example used before:

$$A(x,y) \longrightarrow B(x)$$

and the input is A(a, b), where a and b are constants. We will have two assemblies: one for A(x,y) and the other for B(x). The first assembly has three nodes: C, X, and Y. The second assembly has two nodes: C and X. The C node in the first assembly is linked to the C node in the second assembly. The X node in the first assembly is linked also to the X node in the second assembly. [5] See Figure 7.2. First the C node in the first assembly is activated, and the binding a is passed to the X node in A(x,y), and b to the Y node. Then by applying the rule (i.e., propagating activation along the link), the C node in the second assembly is activated, and a is passed from the X node in A(x, y) to the X node in B(x). Thus a conclusion is reached: B(a).

It should be pointed out that variable nodes do not necessarily contain a particular numerical

[3] However, in order to do this in CONSYDERR, we need to extend the idea of node activation functions as described in Chapter 5 to allow the passing and selection of the numerical values (or symbols) representing bindings in a network. This extension will be detailed in the next section.

[4] Each of the variable nodes in an assembly should also communicate with the C node to send its bindings, so that the C node can check consistency and others, and should receive "instructions" from the C node. The "instructions" are to tell a variable node which binding to take in case there are multiple inputs for a variable node. For details see Appendix.

[5] There should also be some intra-assembly links, which are omitted for simplicity.

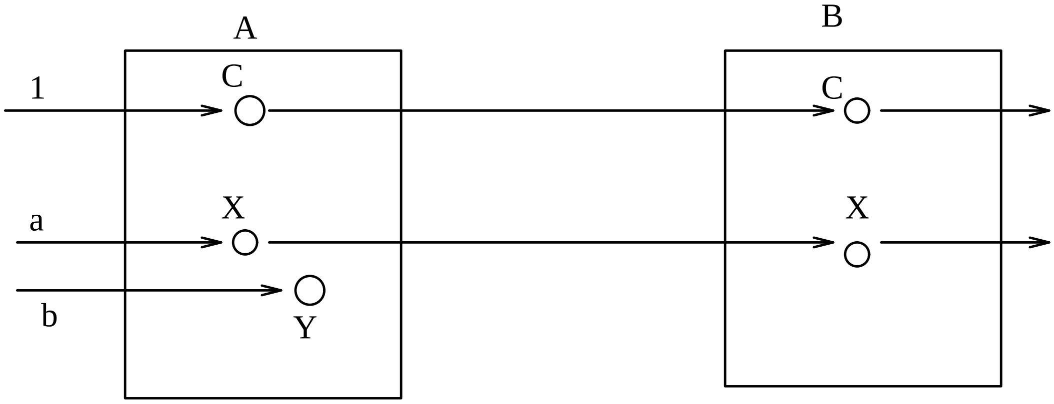


Figure 7.2: The Network for Representing the Rule: if A(X,Y) then B(X)

value or symbol that represents a particular domain object, or in other words, they can be free variables. For example, if we have a rule: $P(x) \longrightarrow Q(x)$, given P(z), we can derive Q(z), by representing z with a special symbol in variable nodes of the assemblies for P(x) and Q(x).

Obviously, this scheme may not work for all cases with respect to FEL. Nevertheless, it is adequate for dealing with all the interesting cases. Details regarding its limits and remedies will be in the Appendix, because readers may not want to go into these details, but all of them should be provided for completeness.

With the above in mind, given a set of rule in FEL forms, one might wonder whether a network for reasoning with that set of rules can be constructed in a mechanical way, without relying on some sort of "art of network building". Although the process of setting up a network given a set of FEL rules is rather complicated, it can be automated by following some fixed steps. All the details are in the Appendix.

The above description regarding CL is readily applicable to CD. From a computational standpoint, CD is exactly the same as CL. The difference is in the representational primitives: instead of concepts and propositions, we have features as basic elements in the CD representation. So whatever works for CL works for CD.

One issue we have to look deeper into is the inter-level connections between CL and CD. The main question is how bindings are passed along when top-down activation, or bottom-up activation, is in effect. A simplistic answer is that we can pass appropriate bindings along with activations the same way as in the intra-level case. One problem that arises from this solution is how we deal with the situation, at the bottom-up phase, where different features of the same CL node have different bindings associated with them (see Figure 7.3). One way to deal with this is to use majority rule principle, choosing the binding that is associated with more features than any others. A related, but complementary, problem is that, at the top-down phase, how to decide a proper binding when there are more than one bindings received at a CD assembly representing a feature, from different CL assemblies (see Figure 7.4). This can occur when two assemblies representing two different facts with different arguments and bindings but sharing some features are activated at the same time. This problem can be solved by choosing the binding associated with the assembly that has stronger activation (and hence stronger confidence). The solutions to the above two problems give the system a flavor of competition and winner-take-all.

To summarize, this section describes rather informally how to add variables into the CONSYDERR architecture. A problem that emerges from the proceeding discussion is exactly what functions we must have in a processing element (node) in order to deal with variables. We need some formal treatments that can qualify the problem clearly, as what follows in the next section.

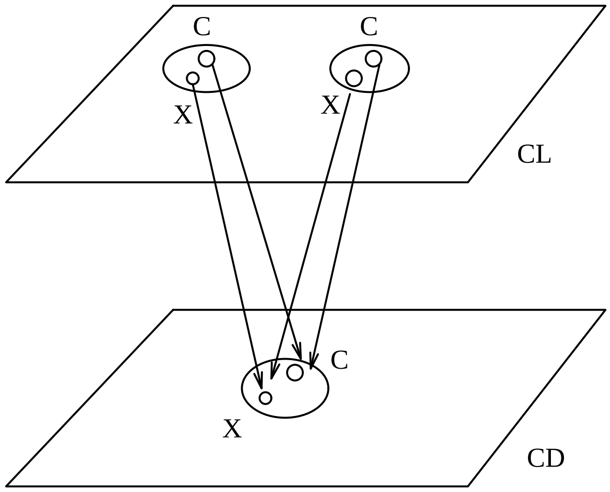


Figure 7.3: Multiple Bindings Received During the Top-Down Phase

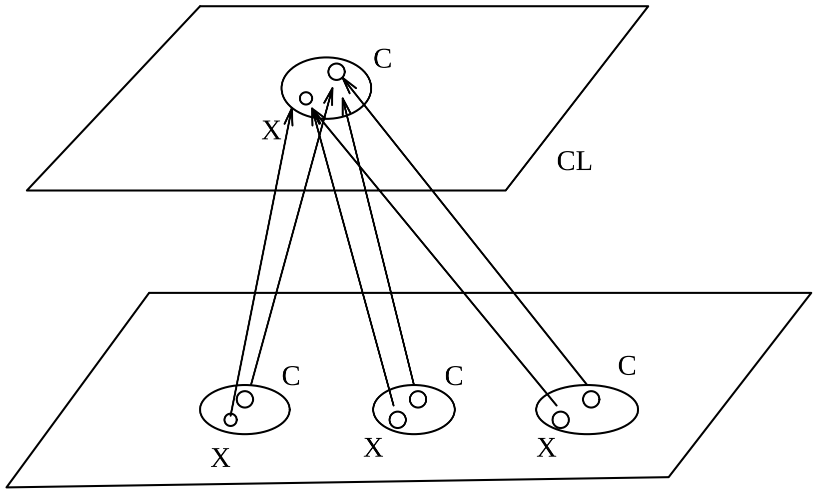


Figure 7.4: Multiple Bindings Received During the Bottom-up Phase

7.5 A Formal Treatment: the DN Nodes and Assemblies

In this section, a formal treatment of the basic elements in CONSYDERR, namely nodes, assemblies and networks, is presented, in light of what was discussed above. First a new model of nodes is presented to subsume the weighted-sum connectionist model, and assemblies are defined thereafter. Finally, we discuss how these elements can be used in CONSYDERR.

In order to incorporate variables and bindings into the model, we will have to extend the definition of a *node* in conventional connectionist models (and consequently the definition we provided for CONSYDERR before). We will use a very general formalism for connectionist models, the *Discrete Neuronal Model*, or the DN model. This formalism is first proposed in (Sun et al 1989, Sun 1989) and later further developed in (Sun 1990, 1991). It is a generalization of conventional connectionist models, aiming at resolving some difficulties inherent in these models (such as variable bindings) by removing some unnecessary restrictions. The generalization in this formalism goes along several dimensions: internal states, differentiated outputs, and temporal responses. The resulting model can deal with variable bindings, rule matching, and confidence value propagation all at the same time. Thus the model forms a reasonable basis for developing a complex reasoning system incorporating variables.

An automaton-theoretic description of a node in a DN model is a 7-tuple:

N=$< S, A, B, I, O, U, V >$
S= the set of all the possible states of a node,
O= the set of all outputs,
A= the set of all the actions (output symbols),
I= the set of inputs,
B= the set of all input symbols,
U= state transition function: S x I $\longrightarrow$ S,
V= action function: S x I $\longrightarrow$ O,

We think of a node as having a set of input and a set of output:

$$I = (i_1, i_2, ..., i_n) \in B'$$

$$O = (o_1, o_2, ..., o_m) \in A'$$

defining

$$B' =_{\text{def}} \Pi B$$

and

$$A' =_{\text{def}} \Pi A.$$

And the state transition and action functions are

$$V : S \times B' \longrightarrow A'$$

$$U : S \times B' \longrightarrow S$$

A network of DN nodes is a graph with (multiple) directed edges between pairs of nodes. An added constraint is if there is a directed link from node i to node j, then $A'_i = B'_j$. Although the formalism is complex (it is aimed for generality beyond this work; cf. Sun 1989, 1990), the *actual* mechanism for variable binding is very simple, as will be seen below.

In this model a set of discrete mapping between inputs and outputs is explicitly specified instead of a continuous activation function. This allows more flexible relationship between inputs and outputs, especially useful for performing variable bindings. An internal state is introduced for a node to have a kind of memory. Internal states can be used for facilitating temporal responses (i.e. inputs at time t can affect outputs at time $t + \tau$). [6] Distinct multiple simultaneous outputs are allowed in this model, which is very important for dealing with variable bindings as detailed in Appendix. Because of the generality, this model can emulate other more conventional connectionist models (see Appendix for details). Conversely, DN nodes can also be implemented with the conventional weighted-sum threshold model (see also Appendix). These two facts together further ensure the compatibility of this formalism with more conventional models.

In the following paragraphs, we will describe how the DN formalism can be used to construct one level of CONSYDERR, given some first order FEL rules with variables. We will use CL as an example for illustration.

We can map from the first-order predicate version of FEL to a CL network with variable binding as follows: A fact (a predicate) is represented by an assembly of DN nodes in CL, and rules are encoded in a network by the connections an assembly has to the other assemblies. For example, Figure 7.5 shows how assemblies representing facts are wired into a network. So CL is made up of assemblies which in turn are made up of nodes representing predicates and arguments, that is,

$$CL =_{\text{def}} \{AS, L\}$$

where

$$L =_{\text{def}} \{(as_1, as_2, i, j) \mid as_1, as_2 \in AS, \quad i, j \in [0, k] \text{ are labels}$$
$$\text{denoting nodes within an assembly}\}$$

$$AS =_{\text{def}} \{(C, X_1, X_2, ..., X_k)\}$$

The connections among nodes are specified in L by labels which indicate which node in one assembly is connected to which node in another assembly. There may also be connections from each X to C and from C to each X, as explained in Appendix.

In each assembly, there are k+1 nodes of the DN type: C,$X_1, X_2, ..., X_k$. The C node contains the confidence value of the fact and connects to other C nodes in other assemblies for representing different facts that are related to it by rules. The X nodes take care of variable bindings (i.e., for arguments to the fact predicate) for a maximum of k variables. The set of states and the set of input/output symbols in the variable nodes are numerical values representing all possible bindings in a domain. The set is fixed a priori. Each element in an assembly, C or X's, is a node with a specific action function (a state transition function is not need for this application), as specified below:

For C:

$$V_0(t) = f(I_1(t), I_2(t), ...I_k(t)) \in A_C$$

> **NOTE**. Action function f here is a weighted-sum function. $I_i's$ are outputs of other C nodes in other assemblies impinging on this node. A_C is the set of output symbols in the node C. Node C can also receive inputs from X nodes in the same assembly, useful in certain situations which will be explicated in Appendix.

And for X_i: i=1,2,....k,

$$V_i(t) = f_i(I_1(t), I_2(t), ...I_n(t)) \in A_{Xi}$$

[6]This mechanism is developed in the process of modeling biological neural networks (see Sun et al 1989), not really necessary for the current CONSYDERR model, but it provides some very useful properties that can be explored in future work.

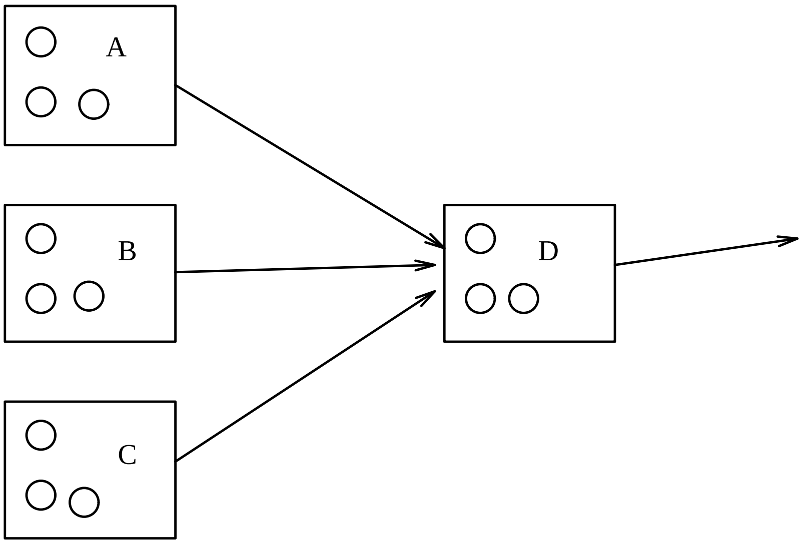


Figure 7.5: The Wiring of a Network for Representing the Rule: if A(X) B(Y) C(Z), then D(W) where X, Y, Z are vectors of arguments.

NOTE. Action function f_i here is a mapping specifying the output of X_i at time t based on $I_i(t)$'s, inputs at that moment from X_i nodes in other assemblies. This function always pass on the binding it receives (some other complications regarding consistency checking, unification, selections, and inputs from C nodes will be explained in Appendix). A_{Xi} is the set of output symbols in the node X_i.

Now we are ready to formally define CONSYDERR again, with variables and the variable binding mechanism included.

Definition 7.6 CONSYDERR *is an architecture composed of a hierarchical CL and a hierarchical CD which is the result of one-to-many mapping from CL, both built with the DN nodes and assemblies, with separate top-down and bottom-up inter-level connectivity and phasic interaction inter-level dynamics.*

All the concepts involved here are discussed before either in this section or in Chapter 3.

To conclude, formal definitions of DN nodes and assemblies are introduced for dealing with variables and variable bindings. Based on these definitions, the CONSYDERR architecture is redefined to accommodate variables and mechanisms for variable binding and for passing along bindings in reasoning. Appendix will provide more details regarding this topics.

7.6 Summary

This chapter discusses the problem of variable bindings as related to the CONSYDERR architecture. In attempting to solve the problem, FEL is first redefined to include arguments for predicates instead of simple propositions. Then we describe informally how to add variables into the CONSYDERR architecture. Some issues are raised that need formal treatments. Therefore formal definitions of DN nodes and assemblies are introduced and the CONSYDERR architecture is redefined to accommodate mechanisms for variable binding.

7.7 Appendix

The first part of the appendix explains the compatibility of the DN model with other models and shows how it can subsume them. The second part of the appendix presents a simple example, to give readers some ideas of what is coming. The third part explains the timing issue, which arises from the example. The fourth part presents a formal procedure for setting up a network given a set of FEL rules. The last part summarizes all types of node functions and assemblies used for representing FEL rules with and without variables.

7.7.1 Compatibility of the DN model with other connectionist models

To see how a DN model emulates other connectionist models, look at various connectionist (neural network) models. In general, connectionist models can be divided into four classes:

- continuous input/discrete activation models (e.g. the linear threshold model (McClelland & Rumelhart 1986)),
- discrete input/discrete activation models (e.g. the Feldman and Ballard (1982) model),
- continuous input/continuous activation models (e.g. McClelland and Rumelhart's interactive activation and competition model (McClelland & Rumelhart 1986)),
- discrete input/continuous activation models (as another possibility).

All of them can be easily handled by our general formalism. To simulate a continuous input/discrete activation model (WLOG, suppose using a weighted-sum threshold activation function), let a discrete neural network model be

$< S, A, B, I, O, U, V >$
where
B=$[0, 1]$,
A=$\{0, 1\}$,
S=$\{\}$
U is not needed,
V= the original weighted-sum thresholding activation function.

This model can emulate the original model and produce the same output: 0 if the weighted-sum of input $<$ threshold and 1 if the weighted-sum of input $>$ threshold. The model can carry out the computation exactly as its conventional counterparts.

Conversely, DN nodes can also be implemented with the conventional weighted-sum threshold model. The question here is how to implement the state variable and the state transition function of a DN node. It has been shown that a multi-layer network can do the job: the hidden layer represents the current state and each input/output value is explicitly represented by an individual cell (cf. Fang and Wilson 1989). See Figure 7.6. The output from the hidden layer is fed back into the input layer to help decide, together with current inputs, which state to enter next.

7.7.2 Implementing FEL Rules in DN Networks

We will derive, in DN networks, a complete representation of a reasoning process for an example in the first order FEL rules.

We will use the following example: proving that Socrates is mortal, based on the general knowledge: if one is human, one is mortal, and the knowledge that Socrates is human. The reasoning goes like this:

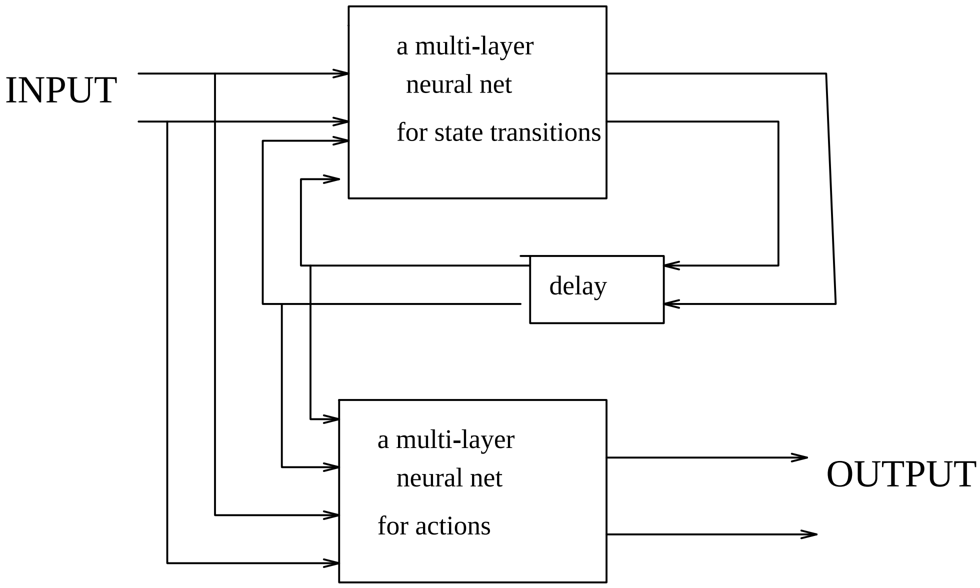


Figure 7.6: Implementing a Finite State Automaton in Weighted-sum Threshold Nets

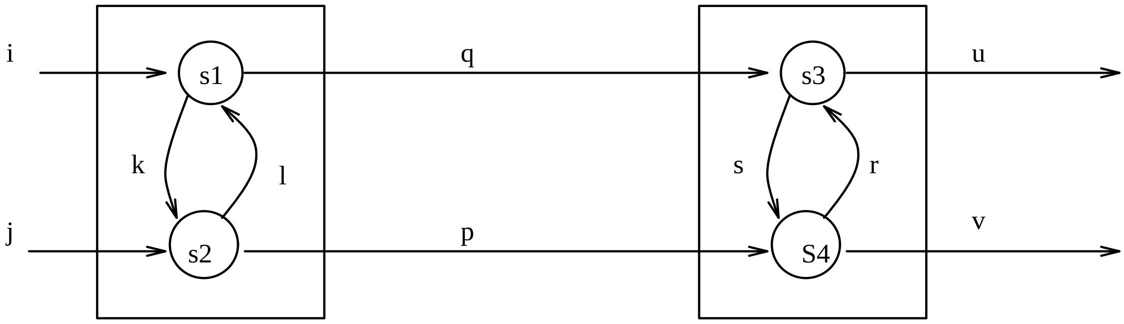


Figure 7.7: An Example Network — Socrates

> if one is human, one is mortal.
> Socrates is human,
> so Socrates is mortal.

The rule to be represented is

> if x is human, x is mortal.

Or

$$human(x) \longrightarrow mortal(x)$$

First we will set up a network for representing the rule. Applying the DN formalism, we build up two assemblies (see Figure 7.7). Each assembly contains a C node and a X node (a variable node for an argument). In the above figure, S1 and S2 are C nodes and S3 and S4 are X nodes. The first assembly represents "X is human" and the second assembly represents "X is mortal". and the input i is the value of the fact "Socrates is human", which is one in this case. The output u for this case is the string "Socrates" or its numerical representation, and the output v is the value of the fact "Socrates is mortal", which is also one in this case.

For C1,

$$I = \{i, l\}$$

$$O = \{k, q\}$$

$$S = real$$

where B_i = real, B_l = string, A_k = real, and A_q= real.

The signals from variable nodes to C nodes are to inform them if there is a right binding present at each variable node. If the bindings are inconsistent, or do not satisfy what is required of the arguments for a particular predicate, the C node representing the predicate should not be activated. For example, for A(x, x, y), the first two arguments must be the same for the predicate to be activated. The signals from C nodes to variable nodes are to tell them if they should pass on bindings. In case the C node is not activated, the variable nodes in the same assembly should not, or need not, work on the variable bindings. Although these issues are not important for this simple example, we will still keep these two types of links there for the sake of generality.

Now the state transition function and action function for C1: Let R stand for real, and Σ stand for string.

$$U : R \times (R \times \Sigma) \longrightarrow R$$

or

$$S \times (i, l) \vdash i$$

$$V : R \times (R \times \Sigma) \longrightarrow A_k \times A_q = R^2$$

or

$$S \times (i, l) \vdash (k, q),\ where\ k = i,\ q = i$$

Expressing them in functional forms, we have

$$U(S, (i, l)) = i$$

and

$$V(S, (i, l)) = (k, q) \vdash (i, l)$$

For X1,

$$I = \{j, k\}$$

$$O = \{l, p\}$$

$$\Sigma = string$$

where B_j = string, B_k = real, A_l = string, and A_p= string.

The state transition function and action function are

$$U : \Sigma \times (\Sigma \times R) \longrightarrow \Sigma$$

or

$$S \times (j, k) \vdash j$$

$$V : \Sigma \times (\Sigma \times R) \longrightarrow \Sigma^2$$

or

$$S \times (j, k) \vdash (l, p),\ where\ l = j\ if\ k \neq 0,\ l = \perp\ otherwise;\ p = j$$

Expressing them in functional forms, we have

$$U(S, (j, k)) = j$$

and

$$V(S,(j,k)) = (l,p) \vdash (i,(i\ if\ l \neq 0,\ \ \perp\ if\ l = 0))$$

It is the same for C2 and X2.

For C2:

$$U(S,(q,r)) = q * c$$

where c is a constant.

$$V(S,(q,r)) = (u,s) \vdash (q * c, q)$$

For X2:

$$U(S,(p,s)) = p$$

$$V(S,(p,s)) = (v,r) \vdash ((p\ if\ s \neq 0,\ \ \perp\ otherwise), p)$$

Simplifying the above notations, we come up with simple assignment statements (so that they will be easily programmable) as follows: assume prime signs represent the next time period.

For C1,

$$S' = i$$

$$q' = l$$

$$k' = i$$

For X1,

$$S' = j$$

$$l' = j$$

$$p' = j\ if\ k \neq 0, \perp\ otherwise$$

For C2,

$$S' = q * c$$

$$u' = q * c$$

$$s' = q$$

For X2,

$$S' = p$$

$$r' = p$$

$$v' = p\ if\ s \neq 0,\ \perp\ otherwise$$

Based on this, the whole process of reasoning can be described as follows:

$$u'' = q' * c$$

and

$$v'' = p'\ if\ s' \neq 0,\ \perp\ otherwise$$

And according to the above equations regarding p', q' and s', we have

$$v'' = j\ if\ i \neq 0,\ \perp\ otherwise$$

$$u'' = i * c$$

where u is the output from the C node and v is the output from the variable node, all in the "mortal" assembly. In this case, the input i is 1, and the input j is "Socrates". So the output u is 1, and the output v is "Socrates", which together mean that Socrates is mortal.

7.7.3 Timing

There are some subtle timing considerations that need to be addressed. Outputs of a node depend on not only inputs from other assemblies, but also inputs from other nodes in the same assembly. Thus it is important to make sure that there is no mutual waiting situation. This, in a way, resembles the deadlock problem in communicating concurrent processes: an order has to be imposed on the input/output relations, so that certain communication is always facilitated to keep the flow going. In this case, intra-assembly communication does not depend on input from other nodes in the same assembly, only on input from other assemblies. This way intra-assembly communication can finish first, and then outputs to other assemblies will be computed without any further impediments.

So we refine the model by dividing I and O into two parts as follows:

$$I = I_1 \cup I_2$$

$$O = O_1 \cup O_2$$

where subscript 1 denotes intra-assembly input/output, and subscript 2 denotes inter-assembly input/output. And also for state transition and action functions:

$$V = V_1 \cup V_2$$

where

$$V_1 : S \times B^{I_2} \longrightarrow A^{O_1}$$

$$V_2 : S \times B^{I} \longrightarrow A^{O_2}$$

$$U : S \times B^{I} \longrightarrow S$$

In this refined definition, V_1 is done during the first part of a cycle, communicating within an assembly. V_2 is done during the second part of a cycle, sending outputs to other assemblies downstream.

7.7.4 Compilation Procedures

CFRDN (Compilation of FEL Rules into DN Networks) is a set of procedures for automatically transforming first order FEL rules into a forward chaining reasoning network.

The main procedures for compiling a set of rules into a network are as follows.

Assume each node has a set of intra-assembly input and a set of inter-assembly input; likewise, each node has a set of intra-assembly output, and a set of inter-assembly output; there is no recursion in the rule set – the rule set is hierarchical.

A bottom-up procedure for setting up rules are SET-UP-RULE-NET:

```
Initialize the rule set to include all rules to be implemented.
Repeat until the set is empty:
  Choose a rule ( P1(X1) P2(X2) ...... Pn(Xn) --> Q(X) | w ) to work on,
       if there is no rule in the set that has Pi as conclusion.
  Identify all existing assemblies for Pi, i=1,2,....,n,
     if there is none make one.
  Identify all existing assemblies for Q,
     if there is none make one.
  If there is only one assembly for each Pi and there is no in-link to Q,
     link each Pi to Q, assign weights to each Pi.
  If there is in-link to Q,
```

```
    link Pi's to an intermediate assembly, then or-link this assembly
    with existing Q to a new Q assembly which assumes all out-links of Q,
    and assign weights as appropriate.
  If there are multiple assemblies for Pi,
    or-link all assemblies for that Pi to an intermediate assembly,
    and then link that intermediate assembly to Q,
    and assign weights as appropriate.
  Delete the rule implemented from the set.
```

Note that we are concerned only with hierarchical FEL. So the relations between LHS and RHS of various rules impose a partial ordering among the rule set. What the algorithm does is to choose a full ordering out of the partial ordering.

The facts (predicates) in the rule set can be divided into three classes:

$$P = P_I \cup P_P \cup P_O$$

where

- P_I is the set of facts that do not appear as conclusions in any rules; it constitutes the input layer.
- P_P is the set of facts appearing in as both conditions as and conclusions; it constitutes the processing layer. This layer can be further divided into multiple layers.
- P_O is the set of facts appearing only as conclusions; it constitutes the output layer.

Notice also that we do not use any nodes with multiple sites here in this procedure. That is because we simply eliminate the need by introducing multiple nodes, each of which replaces one site, for the sake of clarity. Since we always put a bunch of related nodes together in an assembly, this means that instead of one assembly with nodes having multiple sites, now we simply have multiple assemblies consisting of nodes with only a single site.

The procedure for setting up each individual assembly is SET-UP-ASSEMBLY:

```
  Set-Up-C
  Set-Up-X
```

SET-UP-C:

```
    (I= (I1, I2, .....In) is the input from other assemblies)
    (I'= (I1', I2', ...)  are the inputs from Xs in the same assembly)
  output to other Cs:  W * I, if I' are all 1's
                       0      otherwise
  output to Xs:  1 if I <> 0
                 0 otherwise
  state: W * I
```

The output to Xs are to inform them if they should pass on bindings. I' is used to determine if there is right binding present at variable nodes.

SET-UP-X:

```
              (I is the input from Xs in other assemblies )
              (I' is the input from C)
  output to C: 1 if I is not empty
              0 otherwise
  output to Xs in other assemblies: I if I' <>0
                                    nothing, otherwise
  state: I
```

OR-LINK-ASSEMBLY:

```
For C node:
        (I is the input from other assemblies. k denotes the group)
        (each group is a vector by itself)
  output to other Cs: W * I(k) if W * I(k) = max( W * I(j))
  output to Xs: k if I(k) <>0
                0 otherwise
  state: same as output to other Cs.

For X nodes:
           (I is the input from other assemblies)
           (Each element of I is a scalar)
           (I' is the input from C)
  output to C:  I=(I1, I2, ...., In)
  output to Xs in other assemblies: Ik if I'=k
  state: same as output to Xs
```

Sometimes we may need to generate values in a variable node representing an argument in a fact (predicate). For example, A(X1) $-->$ B(X1,X2,C). Two numbers have to be generated to cover X2 and C. The number generated for C is a genuine binding, but the number generated for X2 is not really bound to anything, just denoting a possible binding, or *pseudo-binding.*

GENERATE-ARGUMENT:

```
For C node:
  output to Xs: m
  ( where m > M is a number specific to the node )
  (the same as above otherwise)

For X:
  output to Xs: m if I' = m > M
  state: m if I' = m > M
  (the same as above otherwise)
```

I' is the intra-assembly input to X nodes, and M is a fixed number, greater than n – the number of inputs to a X node, such that if $I' > M$, I' denotes a new binding. Each C node has some fixed m unique to it for this purpose.

We may also need to check the consistency of bindings. [7] For example, A(X1,X2) B(X2,X3) $-->$ C(X1).

CONSTRAIN-ARGUMENT:

```
For C node:
  if constraints are satisfied, then outputs and states are
    the same as above
  otherwise, output 0.
```

Furthermore, we can perform unification of terms, based on the action function. This action function can be implemented as a table lookup. This is a relative easy task, since there is no function allowed in FEL for arguments in a fact (predicate).

UNIFY-ARGUMENT

```
For X node:
               (I is the input from Xs in other assemblies )
```

[7] This is implicit in the semantics of logic. But it has to be dealt with explicitly in implementation.

```
              (I' is the input from C)
  output to C: 1 if all inputs can be unified and I is not empty
              0 otherwise
  output to Xs in other assemblies:
              the result of the unification if I' <>0
              nothing, otherwise
  state: I
```

An example is in order here. Suppose we want to implement the following rule set:

```
P(x1, a) Q(a, x3) --> R(x1, x3)
S(x4, x4) --> P(x4, a)
P(x5, x6) --> Q(x6, x5)
P(x7, x8) --> R(x8, x8)
```

This example involving many issues discussed above, such as multiple rules having the same conclusion, constraint-checking, generating values for arguments, etc. The solution obtained according to the algorithms is shown in Figure 7.8. Part 1 of the figure shows the connections of the network, and part 2 shows a diagram of one assembly. The weight assignment is not shown here.

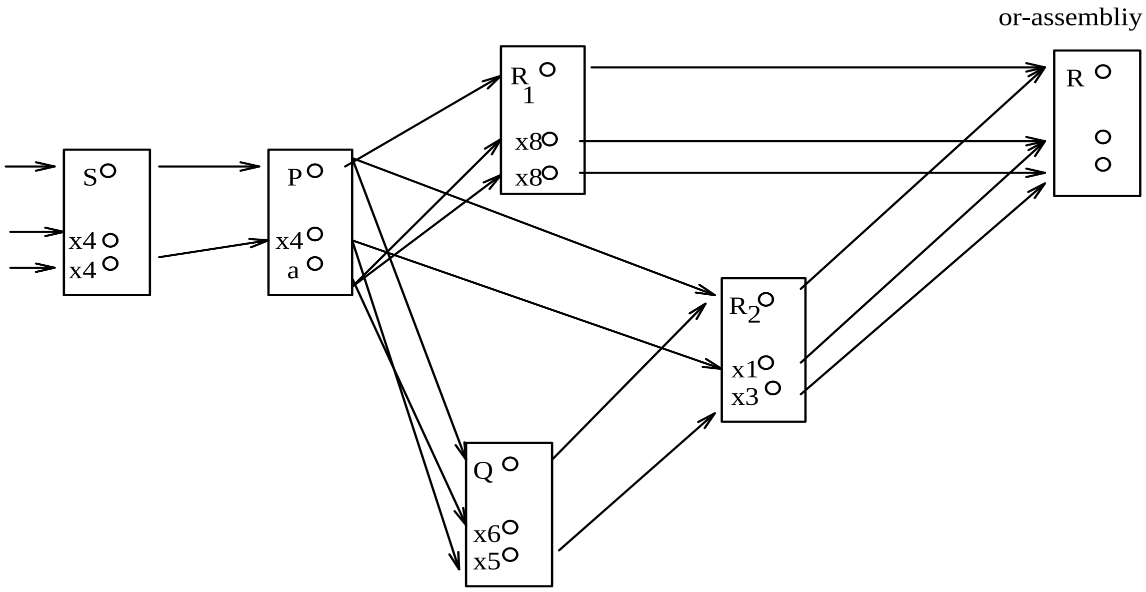
We will summarize below what is needed in a node (i.e., different node functions) and what is needed in an assembly. Types of common computation in a node include the following:

- Conditionals (e.g., if input > 0 then output=1),
- Addition,
- Multiplication,
- Weighted-sum,
- MAX,
- Equality (e.g., I(1)=I(2)) and inequality,
- Selection (e.g., if I(1)=5 then output I(5)),

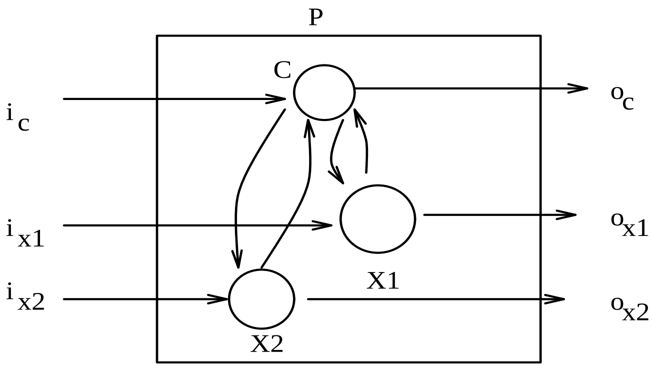
and their various combinations.

There are basically three different types of assemblies (see Figure 7.9):

- Ordinary assemblies – computing weighted sums of inputs and passing along bindings for the variables.
- OR assemblies – computing the maximum (the MAX operation) of inputs and choosing one binding out of many based on who is the winner (the maximum), using the selection operation.
- Complex assemblies – the same as ordinary assemblies, except performing constraint checking and/or unification, and also variable generation when necessary, using the equality and inequality operations.



(1) the network



(2) an assembly in the network

Figure 7.8: An Example Network for a Complex Rule Set

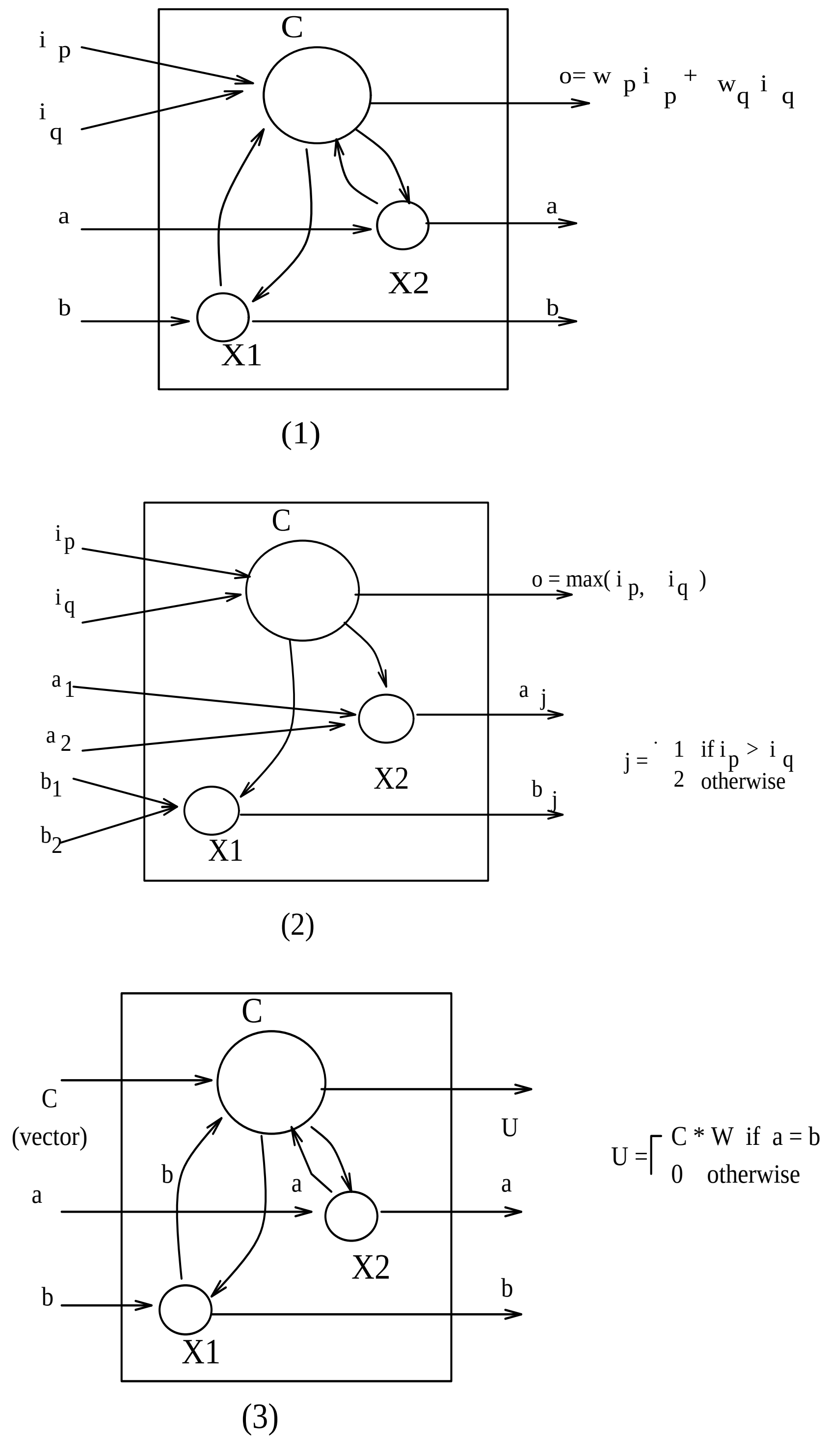


Figure 7.9: The Three Types of Assemblies

7.7.5 Correctness of the Implementation

In the above algorithms, we assume that there is no recursion whatsoever (i.e. hierarchical). Under this condition, the implementation scheme is complete and sound with respect to FEL. [8]

With recursive rules, either direct or indirect, the system will lose its completeness, if implemented with cyclic connections. However we mainly interested in hierarchical FEL, so this is not a problem. Besides, there are also a number of ways we can circumvent recursions. One way is to replicate the predicate a number of times in the network, up to the maximum expected depth of the recursion, provided that we have that information a priori. Another possible way is to use an external stack, which stores and pops out multiple bindings when necessary.

For a propositional FEL theory, the above algorithm for implementing rules also applies, except there is no need for variable nodes for dealing with arguments in facts. It is interesting to note that in case of propositional FEL, no matter whether there is recursion or not, the implementation is always sound and complete. This is because there is no multiple bindings (unification) problem at all for propositional logic.

[8] It is also sound and complete in the binary case with respect to first order Horn clause logic. The completeness comes under the assumption that you replicate and rename a predicate (and all rules involving that predicate) throughout the entire rule set if it appears more than once with different arguments in RHS of rules. Otherwise, the outcomes from these different rules will be MAXed, and therefore the implemented system loses its completeness.

Chapter 8

Reviews

8.1 Overview

Comparisons do ofttime great grievance.
— John Lydgate, *Bochas*

By now we have presented all the rationales for and technical details of developing a connectionist model of commonsense reasoning. In order here is a careful examination of the history of ideas contained or developed in this work, so that we can identify a place for CONSYDERR in our mental space of ideas, approaches and paradigms.

In this examination, we will emphasize the following points:

- How good is each of these ideas in dealing with the kind of reasoning we are interested in?
- How are these different approaches related to each other?
- How does each of these various ideas, approaches and paradigms fare compared with CONSYDERR?

The following three sections will look into three different approaches: rules and logics, connectionism, and case-based reasoning, and will try to answer these questions.

8.2 Rules

In this section, we will discuss rule-based accounts of commonsense reasoning and some attempts at solving the brittleness problem associated with them.

Reasoning is a fundamental aspect of human cognition. It is involved in all kinds of cognitive processes, ranging from language understanding to decision making. Therefore it is of uttermost importance to achieve a better understanding of reasoning as a fundamental instrument of cognition. There are various kinds of reasoning, among which commonsense reasoning is the most frequently occurring and the most fundamental, because it represents the most basic and prevailing reasoning activities. Therefore it is well-justified for us to concentrate on the study of commonsense reasoning, which by itself is almost an all-embracing concept.

Among the existing paradigms of commonsense reasoning, rule-based paradigms are by far the most successful ones, in which generally some generic, common syntactic forms that consist of premises and consequences (or conditions and conclusions) are used as a basic representational means (cf. Klahr et al 1987 and Newell & Simon 1972). However, this type of accounts is not free from problems. The most acute problem is the brittleness (or rigidity) problem. There are various attempts toward amending it, such as non-monotonic logics, circumscriptions and fuzzy logic.

In the rest of this section, we will first discuss rules in general (the first subsection), and then we will discuss the formal logic accounts of reasoning as a special case of the rule-based approach (the second subsection). We will look into some logics for dealing with non-monotonicity in reasoning (the third subsection). We will also discuss fuzzy logic (the fourth subsection). Finally, we will look into probabilistic views of rule-based reasoning (the fifth subsection).

8.2.1 Rules in General

There is a long history of believing that rules can be used to explain human behavior, including human reasoning. It is at the core of the rationalism that has been underlying Western culture (cf. Dreyfus 1972). Human intelligence is believed, in Plato's work for example, to be from rational verbal inferences from facts; syllogism, modus ponens and modus tolens etc. are believed to be the essential forms of reasoning, according to philosophers such as Aristotle. Kant, the great philosopher of the 18th century, is another firm believer of rationalization; in his famous "A Critique of Pure Reason", cognition is said to be pure reason that works on the constructed world built through sensory, *lebenswelt* or life-world, with innate frameworks and rules ("synthetic a priori") that determines ubiquitously correct forms of reasoning.

The modern day advocates of this approach in AI include Prof. Herbert Simon, Allen Newell and others. Their *Physical Symbol Hypothesis* spawned and was used to justify enormous research effort in symbolic AI, which typically uses discrete symbols as basic representational primitives, performs symbol manipulation for reasoning, and typically does it in a sequential, analytical manner, without complex interactions and mutual constraints in representation. There are certain areas in which symbolic AI achieved certain successes, the most notable of which is rule-based expert systems.

Since rule-based reasoning is the most popular paradigm of symbolic AI so far, there are numerous variants and implementations, for example, the OPS family of languages (Klahr et al 1987), the ACT family of production system models of cognition and development (Anderson 1983), and the whole area of expert systems (Hayes-Roth et al 1986). But a severe problem that has been plaguing this paradigm, and symbolic AI in general, is the brittleness of these systems: they do not generalize very well, and they only work within a predefined narrow domain with precisely prespecified situations. This brittleness problem makes the rule-based reasoning unwieldy in accounting for more realistic reasoning tasks and in building more intelligent machines, as alluded to before.

Some people pointed out early on that there was something wrong with a pure rationalistic research agenda. A persistent opponent of symbolic AI, Hubert Dreyfus, wrote:

> Intuitive expertise, acquired through concrete experience, is an endangered species. We must resist the temptation to exalt calculative reason as personified by the computer. Instead, we must recognize that facts, rules and logic alone can not produce either common sense, the ability to go to the heart of a problem, or intuition, our capacity to do what works without necessarily knowing why. (Dreyfus 1972)

In the area of planning, Agre, through his study of human planning activity, recognized the importance of intuition and subconscious actions, or in his words, reactive and improvisatory planning and routines (Agre 1988). With his amazingly simple system, he demonstrated that simple reactive routines can be used to accomplish tasks that are very difficult for traditional AI systems in terms of computational resources and other issues.

It is evident from psychological and psychobiological research (cf. Davidson & Davidson 1983) and the arguments presented in earlier chapters, that intelligent systems for reasoning (including human mind/brain) are complex structures: some parts have to be able to handle rules and analytical knowledge, and some other parts to handle intuitions, reactive routines, etc. There is clearly a need to structure these components so that they fit together forming a functioning intelligent system.

Finally, it is also important to deal with uncertainty, approximity, and adaptivity. we can take a look at the genetic algorithm (Holland et al 1986), which, within the framework of rule-based systems, [1] tackles the problems of approximate match, credit assignment and forming new rules (by mutation and other methods). These capabilities are not what one usually associates with rule-based systems. The genetic algorithm is a unique combination of reasoning and learning models, aimed to amend the brittleness problem of rule-based systems. The capabilities of it stem in part from its representation, which is capable of being rigorous and being flexible at the same time. The rigorous side of it includes precise specifications of LHS and RHS, exact bit patterns of messages posted, and actions being carried out in exact fashion. The flexible side of it includes allowing approximate match of conditions, and forming of plausible, but not exactly correct new rules. [2] Thus it avoids some of the brittleness problem, yet retains the precision and systematicity of rule-based reasoning.

8.2.2 Formal Logics

Formal logics are simple, formally defined languages, capable of expressing rules in a rigorous way, the essential purpose of which is to formalize in symbolic terms the reasoning processes in mathematics and in daily life. However, it has only limited successes so far toward the grandiose goal of accounting for all kinds of reasoning.

There are basically two types of formal logics: propositional logic and predicate logic. Propositional logic deals with declarative sentences (propositions) that can be either true or false. Propositions can be combined, using logical connectives, to form compound propositions. Many different axiomatic systems are studied regarding the truth value of a proposition or the relations between truth values of different propositions. For example, A can be used to represent a proposition "Florida is a warm area", B can be used to represent a proposition "Florida is a flat area", C can be used to represent a proposition "Florida is a rice-growing area", and $A \cap B$ can be used to represent a proposition "Florida is a warm and flat area".

Rules are expressed as logical implications, that is, they have to follow the formal definition and axioms regarding logical implications. Usually rules are encoded using the Horn Clause formalism, in which conditions are the conjunction of atomic propositions, and conclusions are single atomic propositions. For example, $A \cap B \longrightarrow C$.

Predicate logic (first order logic) is made up of predicates which contains arguments, terms for arguments and quantifiers constraining arguments. This type of logic is more complicated because of the introduction of arguments and consequently its bindings to terms, but it is more expressive, and it can represent n-ary relations. For example, "if an area is bigger than another, that former area has a larger population than the latter" can be represented by

$$bigger(X,Y) \longrightarrow \textit{more-populated}(X,Y)$$

However, the formal logic is all too restrictive to account for human reasoning. It has to have all conditions precisely specified in order to perform one step of inference. It does not take into account the gradedness of concepts, propositions, and rules. There is no built-in device for inheritance, similarity-based inference, or reasoning with incomplete information. For example, in FEL (or CONSYDERR) we have a rule

$$A\ B\ C \longrightarrow D$$

[1] The genetic algorithm has a natural rule-based implementation, i.e. the classifier system.

[2] However it cannot account for similarity-based inference and some other capabilities

with weights equal to (0.3 0.3 0.3). When only A (with a confidence 1) and B (with a confidence 1) are known, we can deduce a conclusion D with a confidence 0.6. When A is known instead with a confidence 0.6, then the confidence of D is 4.8. But there is no way in formal logic that this type of reasoning can be done. Similarly, neither can the classic formal logic deal with non-monotonic reasoning, i.e., drawing tentative conclusions from partial, inexact information (one of the aspects of the brittleness problem we are trying to deal with).

Comparing symbolic logics with FEL, although FEL was presented in a symbolic framework, it differs from symbolic approaches (including formal logic) in that it does not perform inferences sequentially and symbolically, and it does not require indexing, retrieving, reorganization, or updating of large databases. The computation is done in place, without the hassle of moving data around. This results in massive parallelism, and in addition, the reasoning process is cognitively more plausible.

There is a long tradition of psychological study of reasoning using logic approaches. Jean Piaget attempted to build models of human reasoning in which logic serves as a foundation. In his highly influential theory, the cognitive development in children is described in terms of acquiring a system of mental logic gradually, and changes are interpreted as emergence of new logical competence (cf. Girotto & Legrenzi 1990).

8.2.3 Extensions to Formal Logics

There are some recent extensions to formal logic which try to remedy the aforementioned problems. Those extensions include autoepistemic logic, default logic, and circumscriptions, etc.

Autoepistemic logic (Moore 1985) is similar to, and actually an improvement of, non-monotonic logic by McDermott (1980, 1982). The basic idea is to model an ideal rational agent's belief of its own knowledge, and to capture the intuition one may have with such a system of beliefs. Formally, a modal operator L is introduced to denote *belief.* Formulas are defined as in standard logics, except that L is allowed to appear with propositions. An autoepistemic theory that describe a rational agent's beliefs satisfies the following conditions (where T is the closure of an agent's beliefs):

1. If $P_1P_2......P_n \in T$, and $P_1P_2......P_n \longrightarrow Q$, then $Q \in T$
2. If $P \in T$, then $LP \in T$.
3. If $P \notin T$, then $\neg LP \in T$.

which basically mean that, if a fact is true, an agent will believe it, and if a fact is not known to be true, an agent must not believe that it is true.

This logic has a number of good properties. For one thing, autoepistemic logic has clear (semantical) soundness and completeness conditions. It is non-monotonic, allowing an agent to draw conclusions based on his lack of information, in which case a tentative conclusion is drawn until more information becomes available. For example, the statement "if I do not believe P, then Q is true" can be expressed as

$$\neg LP \longrightarrow Q$$

So if P is not in a theory T (i.e., the set of belief of an agent), then $\neg LP$ is in T. In this case, the agent can derive Q. However, if P is known to be true, Q is no longer derivable.

Shoham's Logic of Chronological Ignorance (i.e., *Causal theory*; See Shoham 1987) is more or less a subset of Moore's logic, with efficient inference algorithms derived. The restriction is mostly in the form of formulas, which are basically limited to implications. However, this logic is intended to model causal reasoning, and the modal operator L (i.e. $\Box$) is interpreted as *necessity* and M (i.e. $\Diamond$, or $\neg L \neg$) as *possibility*. It can perform non-monotonic reasoning as well. We have discussed this logic extensively in Chapter 5.

Variable Precision Logic (Michalski & Winston 1986) is similar in spirit to Shoham's formalism. The general format for rules in this logic is

IF *condition*
THEN *conclusion*
UNLESS *censor*

Or equivalently [3]

$$condition \quad \neg censor \longrightarrow conclusion$$

However, as pointed out by Michalski & Winston, *censors* play a different role in inference than ordinary conditions. For example, an unlimited amount effort is put into showing that conditions are true, but only a limited amount of effort is used for censors, because the truth values of censors are usually more salient. Comparing this logic with Shoham's logic, conditions in the former logic are, roughly speaking, *necessary* conditions in the latter, and censors are the negation of *possible* conditions, although in Shoham's logic, possible conditions are required not to be provably false based on current information, which is not the case with Michalski & Winston. The basic idea behind both, "not all conditions are equal", is the same, and it has also been applied in FEL and extended into the idea of weighting schemes.

Default Logic is proposed by Reiter (1980) for modeling default reasoning: when no exact conclusions may be reached because of incomplete information, people customarily make assumptions which are consistent with their current knowledge of the world but may be changed later on; these assumptions are *defaults*, which can be expressed as:

$$\frac{A(x) : MB_1(x)......MB_n(x)}{C(x)}$$

This formula means that if A(x) is believed and there is nothing to contradict $B_1(x)$, $B_2(x)$, ..., $B_n(x)$, then conclude the default C(x). This is the most general form of defaults. Several subclasses of this form which can handle most of the common situations have been worked out.

Circumscription (McCarthy 1980) is an attempt at capturing a range of issues in modeling commonsense reasoning, but mainly at modeling the conjectual reasoning regarding properties of some objects: assuming that the objects which one can determine to have a property or a relation are the only objects that do. The general form of circumscription is as follows: the circumscription of P with regard to A is

$$A(\Phi) \wedge \forall X(\Phi(X) \longrightarrow P(X)) \longrightarrow \forall X(P(X) \longrightarrow \Phi(X))$$

where $A(\Phi)$ is the result of replacing all occurrence of P in A with formula Φ. It means that only those tuples X that satisfy P are those that do, given A. Circumscriptions are rules of conjecture to be used together with rules of inference of standard logics. It is non-monotonic in that when new information becomes available previous conclusions may be revised.

Halpern (see Halpern 1990 etc.) proposed a set of logics for reasoning about knowledge, probability, beliefs and uncertainty (Halpern's Logic of Probability was discussed briefly in Chapter 6). In his formalism, there are a fixed set of *primitive propositions* ϕ_i, and *connectives* $\wedge$, $\vee$, and $\neg$ just as in ordinary propositional logic, and *weight formulas* in the form of

$$\sum \alpha_i \beta(\phi_i) >= c$$

[3] Actually there is another interpretation of *censors* as exclusive-or conditions, which gets more attention in their paper.

The logic is shown to be sound and complete for reasoning about probability based on possible world semantics. The difference between this and other logics is that it introduces numerical values into the logic and thus opens up many avenues with regard to expressive and reasoning capabilities.

Although these logics are good normative models of knowledge, belief and reasoning, they have some fundamental shortcomings from the standpoint of modeling commonsense reasoning. These shortcomings include, as mentioned before in Chapter 5, lack of capability of dealing with approximate and incomplete information, partial matches, inconsistency, and most of all, reasoning as a complex, interacting process (cf. Perlis 1986 and Israel 1977).

FEL is aimed at dealing with these above issues (see Chapters 2, 5 and 6). Besides addressing these issues, FEL can handle non-monotonic reasoning as well. For example, we can have the following rules in FEL:

$$A \longrightarrow C \;\; weight = (0.8)$$

$$A \; B \longrightarrow D \;\; weight = (0.5, 0.5)$$

When only A is known (with confidence 1), C is concluded. When A and B both are known (with confidence 1), D is concluded, because D is activated more strongly than C. This flexible handling results from the parallelism and the numerical weighting scheme of the connectionist model. In typical logic-based or otherwise rule-based systems, one rule and only one rule will be picked to fire at a time, but in CONSYDERR all reachable conclusions can be reached, and the final outcome is the combination of these partial results. It seems (see Sun 1991) that FEL satisfies all relevant axioms of various recent nonmonotonic systems (such as *reflexivity, left transitivity, right weakening, cut, reciprocity, distribution*, etc.).

8.2.4 Fuzzy Logic

Fuzzy logic and its basis, the fuzzy set concept, have been around for more than thirty years, and have many important theoretical results and some significant applications. Specifically, the idea of *grades of membership* originates from Zadeh (1965), and that of *linguistic variables* from Zadeh (1975). Later, the idea of *possibility* and *possibilistic logic* was discussed by Zadeh (1978). Since there are many different versions of fuzzy logics, we will inevitably pay attention to only some of them in our following discussion (for an in-depth coverage, see, for example, Dubois & Prade 1988).

The basic idea follows: for each concept, there is a set of objects satisfying that concept to a certain degree, which form a subset, namely *fuzzy subset*. This fuzzy subset contains as its elements pairs consisting of an object and its grade of membership, which represents the degree (i.e. the confidence) with which it satisfies the concept associated with the subset.

Given the above, it is easy to construct a logic with which one can reason about the truth values of various concepts. One version of fuzzy logic is as follows: for an implication or *generalized modus ponens* [4]

$$\textbf{if} \;\; X \;\; is \;\; B \;\; \textbf{then} \;\; Y \;\; is \;\; C$$

given

$$X \;\; is \;\; A$$

we derive

$$Y \;\; is \;\; A \circ (\neg B \bullet C)$$

where X and Y are variables, and A, B and C are fuzzy sets, and the membership function for the result is

$$m_{A\circ(\neg B \bullet C)}(v) = max_u(min(m_A(u), 1, (1 - m_B(n) + m_C(v))))$$

[4]It is adopted from Zadeh (1988).

where m's are grades of membership. And B can be of the form of conjunctions, for example,

$$B = B_1 \wedge B_2 \wedge B_3$$

Transforming this formalism into forms comparable to FEL, we have a rule:

$$ABC \longrightarrow D$$

and given

$$A',\ B',\ and\ C'$$

where A, B, C, A', B', C' and D are fuzzy subsets, we can deduce the result the same way: let x, y, z, w be elements of these subsets,

$$m_{(ABC \longrightarrow D)}(w) =$$

$$\max_{x,y,z} \min(\min(m'_A(x), m'_B(y), m'_C(z)), \min(1, (1 - \min(m_A(x), m_B(y), m_C(z)) + m_D(w))))$$

where m is the grade of membership with respective fuzzy sets.

FEL is inspired by and similar to fuzzy logic (and thus many ideas there can be traced back to fuzzy logic literatures, such as Zadeh 1983), but is based on and implemented by connectionist networks. The difference is that FEL uses the weighted-sum computation of connectionist models for combining evidence cumulatively to reach a conclusion from a rule. [5] By using MAX/MIN, fuzzy logic (Zadeh 1983) does not accumulate evidence (only counting the evidence with the highest grade of membership), which is nevertheless important for commonsense reasoning as discussed before based on the Collins Protocols. FEL provides a scheme for plausible reasoning, which accumulates evidence by using a weighted-sum computation, and it has a sound basis in logic in terms of reducing to classical logic when all values are zeros or ones. Another difference is that, unlike in fuzzy logic, in FEL we do not have to know all grades of memberships in order to perform an inference step: If we have no information regarding the value of a fact, we simply assume it to be zero. This way we can easily allow partial match to a rule, and draw tentative conclusions (non-monotonicity). Also, by using weighted-sum computation and allowing the total weight of a rule to be less than one, FEL rules can be uncertain (probabilistic), making it more expressive of real world situations. With such rules, the confidence for conclusions reached along a chains of inferences weakens along the way, as discussed before in Chapter 5.

An issue here is that uncertainty and vagueness (see Dubois and Prade 1985 and Sun 1985) are mixed together in FEL. Vagueness (or fuzziness) is the ill-definedness of concepts. It is usually handled with a linguistic variable (Zadeh 1983). For example, in the statement "John is tall" , it is very difficult to know what "tall" means exactly: 6 ft, 6.5 ft, or 7 ft? A linguistic variable can be defined that draws its value from a fuzzy subset that defines "tall" with a grade of membership (Zadeh 1983). On the other hand, uncertainty refers to the non-determinism of an event, which has to be dealt with statistically. For example, "John might be home by 8 o'clock", which is a probabilistic statement. The above method handles both the same way, and this mixing-together provides a uniform treatment of various forms of inexactness, and therefore provides a computationally efficient way of dealing with these forms of inexactness. The overall measure serves as a unified device for measuring subjective confidence and for weighing different pieces of information and treat them differently according to their relative importance in reaching the final conclusion.

Summarizing the major differences of FEL as compared with Fuzzy Logic:

- It uses addition and multiplication, which are natural for connectionist networks, instead of MAX and MIN.

[5] Alternatively, we can view FEL as providing a semantics for the weighted-sum connectionist models. Thus FEL provides a foundation for carrying out rule-based reasoning in connectionist networks.

- It allows partial matching and accumulation of evidence.
- It distributes weights to different pieces of evidence, and thus can emphasize certain things and disregard others as appropriate.
- It treats not only linguistic vagueness but also other kinds of inexactness in a uniform framework.
- It accounts for the phenomenon that, in a chain of reasoning, the certainty of conclusions weakens along the way.

8.2.5 Probabilistic Approaches and Uncertainty Calculi

The probabilistic approaches toward reasoning treat beliefs or dispositions about facts as probabilistic events, and utilize probabilistic laws for belief combinations.

The basic probabilistic laws used are for predicting an event based on others known events:

$$p(B) = p(B|A_1)p(A_1) + p(B|A_2)p(A_2) + ... + p(B|A_n)p(A_n)$$

where A's are independent from each other and are exhaustive (notice the striking similarity between this and the weighted sum formula in CONSYDERR [6]); for updating probability based on evidence received, the Bayesian rule can be used:

$$p(B|A) = \frac{p(A|B)p(B)}{p(A)}$$

Pearl (1988) investigates network implementations of Bayesian reasoning. With certain restrictions, some Bayesian reasoning can be carried out by local computations in a parallel network. However the results so far seem to indicate that only very simple types of Bayesian reasoning can be handled efficiently by these kinds of networks.

A question one may ask is: why should beliefs which concerns unrepeatable and often unobservable events be combined by the law that governs repeatable trials in the physical world (i.e. probabilistic events)? Pearl (1988) offers the following answer:

> The fortunate match between human intuition and the law of proportion is not a coincidence. It came about because beliefs are formed not in a vacuum but rather as a distillation of sensory experiences. For reasons of storage economy and generality we forget the actual experiences and retain their mental impressions in the forms of averages, weights, or abstract qualitative relationships that help us determine future actions. The organization of knowledge and beliefs must strike a delicate balance between the computational resources these relationships consume and the frequency of their use.

Although in general I agree with this argument, it does not by itself endorse the *precise* probabilistic laws to be used in descriptive modeling of reasoning; This is because there are many simplifications, heuristics, and rules of thumb involved due to time and space resource limitations, the need for an

[6] The probabilistic law is similar in form to the calculation we perform according to the following rule in FEL: $A_1A_2......A_n \longrightarrow B$ with weight= $(w_1, w_2, ...w_n)$, that is,

$$B = \sum_i A_i w_i$$

economic organization, and the need for generalization capabilities in commonsense reasoning (cf. Tversky 1982 and Posner 1989, etc.).

MYCIN, the first successful expert system, employs an uncertainty calculus of its own (see Buchanan & Shortliffe 1984). The uncertainty calculus is basically a set of heuristic methods, obtained from attempting at fitting the system to the data, like many other aspects of the MYCIN system. However, Heckerman (1986) presents an analysis that gives a probabilistic interpretation to the MYCIN rules of evidential combination, and thus, it is clear that uncertain calculi are not at all a great departure from more traditional theories.

The Dempster-Shafer calculus (Shafer 1974) is another evidential reasoning scheme. It decomposes the world into sets of primitive events, and assigns probability to these primitive events, and then combines them with various rules in ways consistent with the semantics of the decomposition (notably they are not limited to additive rules as in weighted-sum schemes). Thus a judgment is broken down into smaller pieces that can be dealt with easily and that can be later combined.

Nilsson's Probabilistic Logic (Nilsson 1986) represents a different approach in probabilistic reasoning. He made precise the notion of the probability of a sentence through possible world analysis: it is the sum of the probability of the set of possible worlds in which the sentence is true, and the logical entailment is thus reduced to matrix equations and linear programming for solving these equations. The approach applies to any logical system for which the consistency of a theory can be established. Thus, uncertainty is treated in a principled way in this framework. However the computational cost for this logic is tremendous, and the process does not seem to be cognitively plausible.

The arguments against probabilistic approaches, such as those discussed above, are the following:

- These approaches seem too complex to be used in human commonsense reasoning, and too complex to be implemented in "brain-like" network models directly. (Note that FEL can be viewed as a simplification of precise formal models of probability, under the assumption of independent evidence).
- It is clear that human reasoning does not always conform to the assumptions and the laws of probability theory, probably in part because of the complexity of the formal models (cf. Tversky 1983 and Cherniak 1986).
- It is not always possible to obtain precise probability measures

One attempt at applying probabilistic approaches in analyzing human reasoning is Anderson (1990), which is termed the rational analysis by the author. He proposes that human cognition should be analyzed at a more abstract level, from purely a phenomenological point of view, divorced from implementational consideration. And implementation in a cognitive architecture should be performed at later stages. He analyzed a vast array of areas, proposing probabilistic formulations for each of these areas. He has had some success in accounting for experimental data. However, it seems clear that *not all* data can be dealt with in this way.

8.3 Connectionism

Connectionist networks are fine-grained, massively parallel computational systems that account naturally for the parallelism in human memory and reasoning, including the kind of commonsense reasoning we are working on. Connectionism aims at developing systems that can learn from experience and that are capable of generalization from data (and can thus avoid brittleness), by using numerical functions and learning incrementally; thus it seems that connectionism can be a good antidote to the brittleness disease in symbolic AI. Although connectionism provides very promising alternative for the task of accounting for complex commonsense reasoning and avoiding the problem of brittleness, there is currently little effort utilizing the connectionist paradigm directly for the study of commonsense reasoning. In this section we will look into some past efforts that are related to reasoning, rules and high level cognition within the connectionist framework. We will see how they fare as remedies to the brittleness problem.

The first subsection will review connectionism in general; the second subsection will look into connectionist model of high level cognitive processes; the third subsection will concentrate on the problem of rules in connectionist models.

8.3.1 Connectionism in General

Connectionism is a relatively new approach towards modeling intelligence and cognition. It started with building networks imitating nervous systems for solving reasoning-related problems (such as McCulloch & Pitts 1949), and throughout its ups and downs, it always has as one of its goals the modeling of reasoning, thinking, and intelligence. It utilizes, to a large extent, the interaction of a network of simple nodes, and the properties emergent from this interaction (though not all connectionist models fit this description perfectly). Thus connectionist models provide the inherent advantages of uniform representation, simple and generic mechanisms, and mathematical analyzability, which make them much more appealing than some symbolic AI systems that often, though not always, amount to ad hoc tricks.

Although, compared with symbolic AI, connectionism currently seems to be better in modeling only some aspects of intelligence, most notably in low level processes, such as vision, pattern recognition, sensory-motor control, etc., connectionism is also an attractive alternative in modeling high level cognitive processes; It comes with the promise of robust, flexible, fault tolerant parallel distributed systems that use very simple processing elements but render tremendous processing power, that can learn from experience, and that are capable of generalization from data. This power is derived in part from the basic representational primitives of connectionist systems; by using continuous numerical values and functions and conducting small, incremental changes in developing internal representation, connectionist systems are more versatile, reliable and less susceptible to noises, etc.

8.3.2 High Level Connectionist Models

Let us look at connectionist models of high level cognitive processes in general, to see how these models help to alleviate the brittleness problem. There are a large number of high level connectionist models around. For example, McClelland & Rumelhart (1986) contains many such models, ranging from sentence understanding to schemata processing. Because of the sheer number, it is impossible to have a thorough review and comparison — only a few of them can be looked.

An interesting approach is that of Grossberg and associates (see Grossberg 1987). In studying learning and categorical recognition, Grossberg came up with a number of network models, to model classical conditioning and other phenomena, which are ultimately combined into the activation resonance theory (or ART for short). He developed mechanistic explanations for a wide range of classical conditioning data, and posited cognitive constructs, such as drive representation, incentive, expectancy, and frustration, etc. to account for various situations, contexts, and motivational forces behind actions. It is interesting to note that this work was developed at approximately the same time as symbolic AI, but it went quite the opposite way.

In Kosko (1988) causal inferences [7] based on the fuzzy set theory are combined with connectionist models and fuzzy/inexact information is used in the inference. Each node in the system represents a fixed concept and has continuous activation values, but there is no variables in the system and there

[7] The way causal reasoning is carried out in Kosko (1988), unlike in CONSYDERR, is not fully justified by available evidence.

is no mechanism for variable bindings. This limits the capability of the system for representing causal relations, because of the increased computational complexity due to the use of multiple instantiation.

In Derthick (1988), a connectionist knowledge representation system, μKLONE, is presented, the core of which is a connectionist reasoning system. Inferences in this system are carried out by constraints satisfaction through minimizing energy which is proportional to the outstanding constraints. By constructing the energy function in a way that captures the underlying relations between logical formulas, this process can produce an optimal (or near optimal) interpretation of the inputs. The actual process of energy minimization is done by the Boltzman machine settling procedure. One problem with this approach is that the Boltzman machine is extremely slow, and may not produce the best results. A real problem for me, however, is that not all logical relationships between sentences and all inference modes can be captured in energy functions. As a matter of fact, most of the commonsense reasoning, as shown by the protocols, can not be done by this energy minimization procedure in any obvious way.

In Hendler (1989), a hybrid system of marker passing and connectionist networks is presented, to augment symbolic planning systems in their ability to pick up the right move by priming the relevant concepts. The role of the connectionist network is that of association, and therefore there are no rules and no variable binding capabilities. Although this works well in that particular application, for more complicated tasks we have to go beyond pure associations, and into more elaborate mechanisms for more systematic reasoning, within the framework of connectionism.

One important issue in modeling high level cognitive capabilities by connectionist networks is how to deal with rule-following behaviors and how to carry out rule-based reasoning. Because it is such an important issue by itself, we will discuss it separately in the next subsection.

8.3.3 Rules and Connectionism

Connectionism has made some claims about rule-based behavior and reasoning. One point of view is that rule-like behavior is the result of complex interactions of network components, in deterministic or statistical ways, and therefore there is no fundamental difference between rules and non-rules. Another viewpoint advocates implementing rules directly in connectionist networks. Among the works that are skewed in this direction are Touretzky & Hinton (1987), Touretzky (1986), Barnden (1988), and Dolan & Smolensky (1987), etc. Two systems developed recently, concurrent with CONSYDERR, are Ajjanagadde & Shastri (1989) and Lange & Dyer (1989).

Now let us take a detailed look at some early work that is designed specifically to deal with rule-based reasoning in a connectionist framework: Touretzky and Hinton (1987) and Barnden (1988). These are all important pioneering efforts in this field; nevertheless, they each suffer from a number of problems from the standpoint of accounting for commonsense reasoning. In each of these schemes, parallelism is lost in some way, either because of the matching process of hardwired rules or a centralized working memory. As a result, the parallel and spontaneous nature of commonsense reasoning is not adequately accounted for, which in turn results in brittleness. For example, Touretzky & Hinton (1985), the first work in implementing rule-based reasoning in connectionist networks, fits the above description. They basically emulate the structure of a symbolic rule-based (production) system, with separate modules for working memory, rules, and facts; an elaborate pull-out network is designed to match working memory data against rules and to decide which matching rule is to fire. Competition and winner-take-all processes are used for the matching purpose. The mechanism does one match at a time and thus slows down reasoning. There is no mechanism for backtracking, and there is no room for approximate reasoning. The resulting system is the equivalent of a simple sequential symbolic rule-based system. Barnden (1988) represents another early attack on the same problem. In his system, data reside in grid like networks (called Configuration Matrices), coded with the help of adjacency relations and highlighting techniques; Rules are hardwired with circuitry for detecting

	Parallelism for Rules	Numbers of Variables	Time Complexity	Space Complexity	Inference Directions
CONSYDERR	Yes	No Limit	Low	Low	Forward
A & S	Yes	No Limit	Low	Low	Forward and Limited Backward
L & D	Yes	No limit	Low	Low	Forward
T & H	No	One	Very High	High	Forward
Barnden	No	No Limit	High	High	Forward

Figure 8.1: Comparison Regarding Performance

the presence of data that match a particular rule, and an "Action Part" module associated with each rule is used to add a new data structure representing the conclusion from the matched rule. The problem here is the symbolic manipulation necessary to match the rule against data, which is a slow and complicated process in a connectionist network; as a result the intrinsic parallelism and spontaneity of commonsense reasoning are not fully captured. Although there is some parallelism in the model, the system is a basically sequential rule-based system, carrying out symbolic syllogistic reasoning. There is no approximate reasoning capability either.

We shall now turn our attention to two pieces of more recent work on connectionist models of rule-based reasoning (both appeared first in the Proceedings of Cognitive Science Conference in 1989, the same as Sun 1989, which began to develop ideas manifested in CONSYDERR). Comparing one of them, Ajjanagadde & Shastri (1990), with CONSYDERR, we see that both models have a connectionist network that can do logical reasoning and handle variable bindings, and both use local representation. Ajjanagadde & Shastri explored the temporal dimension of activation patterns for representing variables and bindings (which is a novel idea), while our model extended the alphabet of signals between nodes in a network to handle variable bindings. However, it is easy to see that different frequencies in node outputs in Ajjanagadde & Shastri can be readily represented by a single output value (symbol) as in our model. Ajjanagadde & Shastri's system performs both backward-chaining reasoning, while CONSYDERR performs only forward-chaining (note that human reasoning processes are mostly forward chaining. cf. Posner 1989). Because of the nature of forward-chaining reasoning, the only constraint on the representation for CONSYDERR is acyclicity, and thus it is more general in representation. While both systems can employ classical logic, by introducing a fuzzy evidential reasoning scheme CONSYDERR has the additional power of approximate reasoning and it can accommodate partial information and inconsistency.

Lange & Dyer (1989) present another interesting rule-based reasoning system built out of connectionist models. This system is very close to CONSYDERR, in both variable bindings and rule activation mechanisms. However, compared with CONSYDERR, their system does not have a unifying underlying network model that can generate all kinds of mechanisms appearing in the system; signature (used for variable bindings) and numeric activation form two separate components. Our work is based on the foundation of the DN formalism that has been well studied in its various aspects and is the force integrating the whole system. Our system also provides a logically sound scheme (and a semantics, a way of interpreting the weighted-sum connectionist models) for commonsense reasoning in connectionist models.

Although almost all of these aforementioned connectionist rule-based systems share much commonality with CONSYDERR in its CL part, they do not deal with similarity matching as done in the CD part of CONSYDERR, and consequently they are unable to utilize the synergy between the two types of representations that has proven to be essential to CONSYDERR. Therefore, they are unable to do more than what a typical symbolic rule-based system is capable of, besides massive

	Activation Values	Temporal Patterns	Symbolic or Subsymbolic	Approximate Reasoning
CONSYDERR	Numerical	No	Both	Yes
A & S	Numerical	Yes	Symbolic	No
L & D	Non-numerical	No	Symbolic	Limited
T & H	Numerical	No	Symbolic	No
Barnden	Non-numerical	No	Symbolic	No

Figure 8.2: Comparison Regarding Implementation Mechanisms

parallelism, and do not deal with reasoning as complex processes. See Figure 8.1 and Figure 8.2 for a complete comparison.

In addition, compared with these systems, CONSYDERR has the following advantages: the overall architecture of CL/CD dual representation is a principled way (see Chapter 3) of exploring the synergy between rule-based reasoning and connectionism, or between rule-based components and similarity based components. This integration on one hand adds continuity (embodied in similarity-based connectionist networks) to the discretized rule-based system, so that it can better model continuous thought processes; on the other hand, it adds structures (namely rules, as well as variables and bindings) to a structureless, associationistic connectionist network, giving it the rigor, precision and directedness which it needs to model some cognitive tasks. The architecture also provides a unified framework (i.e., a connectionist framework) for both symbolic (or conceptual) reasoning and non-symbolic (or subconceptual, intuitive) reasoning, although in a limited way. This integration is novel and based on the dichotomy of conceptual and intuitive reasoning (cf. Dreyfus 1987). Secondly, the way in which CONSYDERR handles evidence, weighted and cumulative, is equated with a logic FEL which in turn is a generalization of a modal logic accounting for causality. Thus CONSYDERR is well-equipped for capturing reasoning processes based on commonsense causal knowledge. [8] Some limitations and shortcomings of CONSYDERR are listed in Chapter 9.

8.4 A Brief Look at Case-Based Reasoning

Here we will take a very brief look at case-based reasoning (CBR), and try to explicate the relation between CONSYDERR and CBR.

The case-based reasoning approach is best represented by Riesbeck & Schank (1989). Riesbeck & Schank (1989) believe that "case-based reasoning is the essence of how human reasoning works", or in other words, case-based inference is believed to be the most basic element of reasoning. [9] They characterize case-based reasoning as performing the following steps:

- **Input** — receive data
- **Index** — find relevant pointers

[8] As a historical footnote, Grossberg' work is, to a large extent, for modeling some basic rule-like behavior: classical conditioning, but as in CONSYDERR it also takes into consideration things that are internal, inexact and holistic in nature, not just rule-like regularity. Rule-like regularity is embodied in network connections, each of which can be interpreted as a rule but together they form a complex process. It can be viewed as one of the precursory work in combining rules with connectionist architectures.

[9] Although they believe that "in a broad sense, everything is a case", I do not think that it is appropriate to call abstract rules cases, which is usually used to refer to concrete scenarios and episodes, and it is neither the way their systems are constructed. So it is appropriate to evaluate their ideas with the usually meaning of the word *case* in mind.

- **Retrieve** — find relevant cases based on indices
- **Adapt** — modify the retrieved cases for the current situation
- **Test, Repair and Store** — try out the solution and store it into the case base after it is tuned into a correct form.

This entire process is construed to be sequential and iterative. They present many interesting examples of systems performing case-based reasoning, in planning and problem solving.

There are some similarities and differences between philosophies behind CONSYDERR and CBR. The similarity is self-evident, especially if we look at one of the applications – GIRO. The differences on the other hand are the following:

- Rules (i.e. abstract knowledge) are taken in CONSYDERR to be the most important cognitive mechanisms, although no doubt there are other kinds of knowledge. Only when there is no rule directly applicable, will cases (or analogous knowledge in general) be used.

  For example, a rule is: **if** a place is warm, flat and with enough fresh water supply, **then** it can be a rice-growing area. When there is such a rule available, we only need to apply such a rule in deciding if an area is a rice-growing area; only when there is no such rule, or the conclusion of the rule is indecisive, may we apply analogous knowledge (or cases). Of course, such rules are not context free and work in isolation as often seen in expert systems. Even in existing case-based reasoning systems, various forms of rules are used, which is more evidence for the primary role of rules in cognition.
- There ought to be, therefore, both rules and similarity matching (of analogous knowledge) in the system, and rules should take precedence.
- Similarity matching should be done in a massively parallel fashion (see Waltz 1989 and Thagard & Holyoak 1989 for similar points) and spontaneously, without incurring huge computational overhead, because, for commonsense reasoning, the matching and adaptation process cannot be very analytical and deliberative. Simpler algorithms are thus preferred.
- Similarity matching should be done at a lower level, that is, matching should be an intuitive, holistic process (see Dreyfus & Dreyfus 1987).
- Rules, in terms of its form, can be a useful representational framework for both concrete and abstract knowledge. CONSYDERR adopts a simple, uniform representation that encompasses both cases and rules.

On the other hand, symbolic programs are certainly easier to use in writing special purpose routines than connectionist networks in which designing special circuits is undeniably difficult. As a consequence of that, symbolic programs can represent episodic knowledge easily, while it is not obvious for CONSYDERR how to represent such knowledge (see the section on planning in Chapter 4 for some thoughts on that).

It should be mentioned that in CBR there are a variety of opinions that are not represented by Riesbeck & Schank (1989), for example, the collections of papers in the proceedings of DARPA case-based reasoning workshops contain a large variety of different ideas. It is beyond the scope of this thesis to try to cover them all.

Barnden and Srinivas (1990) describe an attempt at utilizing case-based reasoning in connectionist networks, for the purpose of combating the rigidity that is undermining both symbolic reasoning systems and purely rule-based connectionist systems such as Barnden (1988) or Ajjanagadde &

Shastri (1990), which is exactly the same idea as what motivates the CONSYDERR architecture. Thus this system, COMPOSIT, is of special interest to me. The system is adopted from their original rule-based connectionist system as discussed before; it has multiple configuration matrices (CMs), each of which contains a case for short term processing; a relative small set of gateway CMs provides the interface between short term processing and long term memory, which is not CM but instead encodes cases in connection weights; cases in non-gateway CMs compete to have their contents copied into gateway CMs, where it can cause cases similar to it to be retrieved from long term memory, and some symbol substitutions take place after the retrieval to adapt the cases to the current situation.

Comparing CONSYDERR with Barnden & Srinivas (1990), we notice the following differences:

- CONSYDERR is more massively parallel: there are no gateways and complex copying and substitution processes that may hamper parallelism.
- Matching in CONSYDERR is more automatic (spontaneous): there is no special matching mechanism. (Of course this potentially limits the matching capability.)

Overall, the two different systems represents two different attacks on the same problem, and therefore there is a chance that the two approaches might converge somehow.

In summary, it seems that although case-based inference (or the use of analogous knowledge in general) is very important for commonsense reasoning, it is supplementary to abstract knowledge encoded in rules. In other words, rules are primary, and concrete analogous knowledge are secondary, and they should be integrated into a unified framework, such as done in CONSYDERR.

8.5 Summary

This chapter examines, explains, and contrasts existing approaches and systems, identifying their similarities and their differences compared with CONSYDERR. Through this process, the advantages and characteristics of CONSYDERR are better demonstrated.

Chapter 9

Conclusions

9.1 Some Concluding Remarks

When we have done our utmost to arrive at a reasonable conclusion, we ... must close our minds for the moment with a snap and act dogmatically with our conclusion.

— George Bernard Shaw, *Androcles and the Lion*

So, after all is said and done, what have we achieved thus far? Or asking the question another way: How well have we succeeded in the tasks we set out to do at the beginning of this thesis? Let us examine these questions by looking into the models, the architectures and the systems we have developed.

A connectionist architecture CONSYDERR are developed that integrates rule-based reasoning into connectionist networks and couples localist networks with similarity-based distributed representation and reasoning. It turns isolated, context-free, and all-or-nothing type rules into an interacting process enmeshed in a network with graded links and activations that combines pieces of evidence and produces plausible conclusions based on given input regarding particular situations.

The Collins Protocols are well accounted for by systems built with the CONSYDERR architecture. All the data explained in Chapter 2, including that which involve rules, that which involve mixed rules and similarity, and that which involve inheritance, are dealt with uniformly with the same simple and integrated architecture. The patterns that underlies the protocols come out naturally as a result of the working of a two-level structure with a weighted-sum computation in each element.

A model of causality is developed to account for commonsense causal knowledge, and can be carried out naturally by weighted-sum connectionist models in general and by CONSYDERR in particular. The fortuitous but extremely fortunate match between connectionist models and models of causal reasoning results in a effacacious analytical and computational tool, namely, the CONSYDERR architecture, that captures a certain range of commonsense causal reasoning.

CONSYDERR was developed with the aim of being an *integrated* model that can deal effectively and efficiently with a (seemingly disparate) set of important problems in commonsense reasoning (the basic elements of commonsense reasoning): rule application, evidential combination, similarity matching, inheritance (in both ways), etc. The important point to emphasize here is that these problems are handled together in a single unified model that has no special mechanism for any single one of these problems. Through data and conceptual analyses, these problems are reduced into a simple framework: rule application plus similarity matching (or, more elaborately, a rule encoding scheme FEL plus a two-level structure that includes intensional representation for similarity matching). This set of different problems turns out to be the different manifestations of the same problem. And this problem is handled by CONSYDERR, which thus serves as a unifying model for commonsense reasoning.

Having assessed our results, the next question is: What lessons have we learned from the work? One of the lessons we learned concerns an explanation of the capabilities of CONSYDERR — How can we achieve the power as demonstrated above in such a simple architecture? The answer is, simply put, that the power of the CONSYDERR system derives from:

1. The capability of the weighted-sum computation for representing and utilizing commonsense causal knowledge;

2. The intensional approach, i.e., combining intensional (feature-based) representation with extensional representation, so that their *synergy* can be achieved.

Another lesson concerns the relationship between rules, connectionist models, and causal representation. Through a logical formalism that connects a modal logic and the weighted-sum computation of typical connectionist models, rules and connectionist models are shown to be comparable in representation and reasoning power. Moreover, the logic formalism is meant to capture causality in representation and reasoning; thus, it is clear that connectionist models (especially ones with the weighted-sum computation) are capable of and suitable for capturing the commonsense causal knowledge in reasoning.

9.2 Limitations and Future Work

Now we can ask another set of questions that are complementary to the previous questions: What are the limitations of CONSYDERR and what further improvements are desirable?

9.2.1 Limitations

There are undeniably shortcomings and limitations to CONSYDERR. Below we will discuss briefly some of them.

One of the potential limitations is the simplicity of the adopted feature representation. Although this simple representation is justified in some ways and works well for the task at hand, it may have to be extended to more complex representations for more complex applications, for example, natural language processing, story understanding, or planning, judging from related research in AI.

Other serious limitations involve the scarcity of the treatment of learning in this thesis, including (1) learning of new rules, (2) incorporation of new rules into CD, and (3) learning of feature representation, etc. These issues are to be addressed in subsequent work.

A limitation in the experimental study (covered in chapter 4) is that no detailed domain analysis is performed, and consequently no complete domain model is developed, and also most of the examples used are small.

Yet another limitation is in analogy-making, which is a natural extension of similarity matching dealt with in this thesis. Viewed from the perspective of modeling analogy, the way in which the similarity is found and the way in which the adaptation to current situations is made in CONSYDERR are not very sophisticated, and the system deals only with the most rudimentary (and the most common) forms of analogy. Thus the scope of applicability for CONSYDERR is potentially limited, and extensions have to be developed for more complex applications.

Beside the above, there could also be more complex interactions between the two levels, so that more sophisticated reasoning patterns can emerge into existence, and more levels with different granularity in representation, so that finer and finer distinctions can be made.

In terms of the brittleness problem, cautions have to be given; that is, although all the aspects identified (in Chapter 2) are dealt with to a certain extent (except learning of course) within a unified framework, it should be emphasized that it is not at all the case that the brittleness problem is completely solved in this work; rather, some steps are taken here toward the eventual full understanding and solution of the problem.

There are some other things that limit the capability of CONSYDERR in its present form. These things might be handled somehow for further improving the CONSYDERR architecture, but have not been accomplished in this work for various reasons, for example, the space limitations, or the focus of this thesis, etc. These things include:

- Temporal aspects of Shoham's logic, which is ignored in this work for some reasons (see Chapter 5 for some discussions).
- Temporal reasoning in general, which is a broad and difficult problem that is dealt with here (see Chapters 4 and 5 for some suggestions).
- Backward chaining and other inference algorithms (except forward chaining) in FEL, which might be needed in some commonsense reasoning tasks and are nice to have for the sake of completeness in terms of reasoning modes.
- Non-hierarchical rule sets, or recursive structures, in CONSYDERR, which is an unresolved problem and a major limitation of CONSYDERR in terms of its logical capability (see Chapters 5 and 7). This issue may interact with the temporal reasoning issues, and thus they may have to be studied together.
- Complete variable binding capabilities, especially for functional symbols, which may require much more detailed technical work (see Chapter 7, including its appendix, for some suggestions regarding this).
- Higher-order predicates beyond first order ones, in FEL and in CONSYDERR. This is a further extension of the variable binding problem listed above.
- Complete analysis of interaction/generativity in CONSYDERR, especially a clear characterization of its scope (Chapter 6 contains some preliminary discussions of it).

The above issues form the outboundary of the CONSYDERR architecture in terms of its applicability.

9.2.2 Future Directions

It will be useful to look into the question of what lies ahead in terms of further developing the ideas embodied in the CONSYDERR architecture, in order to make CONSYDERR more complete, more accurate and more computationally efficient. We can look into a number of aspects which might help the most in further improving the architecture and in applying it to a wider range of domains.

One of these aspects, as mentioned in early chapters, is how **temporality and sequences** can be represented in CONSYDERR so that it can be applied readily to many other domains, for example, planning, hierarchical decision making, and some aspects of natural language processing. As hinted before, temporal simulation can be used to deal with temporality issues. In so doing, some sort of feedback connections might be needed in one level or the other in order to update the current state of the reasoning system, to set the stage for the next step, to keep track of sequences, and also to assert choices, preferences and constraints.

Another aspect that may be of great importance for the further development of CONSYDERR is **automatically developing distributed representation**, through grounding high level processes into low level processes (i.e. symbol grounding; cf. Harnad 1990), and through mapping syntactic representation into semantic representation by learning algorithms (cf. Dyer 1990). Hopefully, features developed this way (which may be either interpretable or uninterpretable) may better capture the similarity between concepts involved in given situations, and may lead to more accurate models of commonsense reasoning based on analogous knowledge.

In addition, it is important to study in greater detail the subsymbolic featural representation and its interaction with explicit representation, especially with regard to the exploration of the synergy which interaction generates. For one thing, a concept, especially when developed in the above way, should have **weighted features** with weights indicating their relative importance or salience in relation to a given concept, so that the mutual influence between the concept and a particular feature is proportional to the weight the feature has to the concept (relative to other features). In this way, a minor, less salient feature of a concept, if activated, will have less possibility of firing up the concept, and the concept, if activated, will in turn activate the feature to a lesser degree, so that other concepts that share this feature will be activated to a relatively lesser extent – all proportional to their correlational strengths. Put it another way, a feature with a high weight to a concept is more contingent to the concept, and a feature with a low weight is less contingent to the concept (cf. Leonard 1967).

An important aspect to be pursued is **learning**, which includes, as indicated earlier, learning rules in CL, incorporating rules into CD with respect to the particular CD representation adopted, and rule interactions in learning; This is difficult especially when applied to problems of realistic sizes (i.e. not toy problems). The study of learning can intermix with the study of representation, and together a better understanding of connectionist models of reasoning can be achieved.

I believe the issues of **interaction and generativity** are important and should be further studied. Chapter 6 touches upon this issue, taking a cursory look at how interaction and generativity work in CONSYDERR to fill in knowledge gaps and to produce new concepts on the fly. Further explorations may reveal additional properties which are absent from traditional rule- or logic-based reasoning systems. Related problems that are worth pursuing include what the requisite minimal built-in structure is, how consistency can be maintained, and how the recruitment learning algorithm can be used to select CL nodes for newly-generated concepts.

Yet another aspect is the development of better mechanisms for taking **contexts** into consideration. This includes learning of context rules, hierarchical structures of contexts, and more elaborate interactions of contexts and reasoning, etc.

A better understanding of these above issues may lead to the development of brand new types of **massively parallel knowledge-based systems**, which are not mere parallelizations of serial inference systems but represent a completely different kind of reasoning processes — an interacting, complex process, as alluded to before. This kind of reasoning processes can produce results that are not explicitly put in the system, and has a range of properties not seen in serial logic-based inference systems. Further studies are needed to explore issues involved in developing such systems for practical purposes.

Finally, along a different line, it will be interesting to explore the relation we hypothesized in chapter 3 regarding different **styles of thinking** that CL models (partially) deliberative, analytic thinking while CD models (partially) intuition. Some analogies was made regarding this in Chapter 3, but there was never any in-depth study of it; further explorations may produce some new insights into the working of and the interaction between various reasoning capacities in a cognitive agent.

9.3 Summary

It is the end that crowns us, not the fight.
— Herrick, *Hesperides*

Let us summarize in very brief terms what has been done in this thesis (so that readers can obtain a complete picture of what this work is all about). In short, a connectionist model for robust reasoning, CONSYDERR, is proposed to account for some common reasoning patterns found in commonsense reasoning and to remedy to a certain extent the brittleness problem found in typical symbolic systems (although, of course, it cannot completely solve the brittleness problem). Different from other existing connectionist systems, a dual representational scheme is devised, which has extensional objects (localist representation) as well as intensional objects (distributed representation with features). By using feature-based distributed representation in addition to the localist representation, we are able to explore the synergy resulting from the interaction between these two types of representations and from the interaction between rule-based reasoning and similarity-based reasoning. This synergy helps to deal with problems such as partial information, lack of exact match, property inheritance, representational interaction, etc. In addition, a logic is developed for capturing and utilizing commonsense causal knowledge, which is equivalent to a weighted-sum connectionist model. Therefore, combining the two aspect, the CONSYDERR system is capable of accounting for many difficult patterns in reasoning. This architecture demonstrates that connectionist models equipped with symbolic capabilities are effective tools for modeling reasoning capacities and suggests that it can also be useful for constructing efficient practical systems (utilizing massive parallelism). This work shows that connectionist models of rule-based reasoning are not mere implementation of their symbolic counterparts.

Appendix A

References

D. Ackley, Stochastic iterated genetic hillclimbing, TR CMU-CS-87-107, Carnegie-Mellon University, 1987

P. Agre, The Dynamic Structure of Everyday Life, Technical Report, MIT AI Lab, 1988

V. Ajjanagadde and L. Shastri, Efficient Inference with Multi-place Predicates and Variables in a Connectionist System, *Proc.11th Cognitive Science Society Conference*, pp.396-403, Lawrence Erlbaum Associates, 1989

I. Aleksander, The logic of connectionist system, in: I. Aleksander ed. *Neural Computing Architectures*, MIT Press, 1989

R. Allen, Connectionist state machines, Technical Report, Bell Communication Research, 1988

J. R. Anderson, *The Architecture of Cognition*, Harvard University Press, 1983

J.R. Anderson, *The Adaptive Character of Thoughts*, Lawrence Erlbaum Associates, 1990

J. Anderson and E. Rosenfeld (eds.) *Neurocomputing*, MIT Press, 1988

D. Ballard, Parallel logical inference and energy minimization, TR 142, Department of Computer Science, University of Rochester, 1986

J. Barnden, The right of free association: relative-position encoding for connectionist data structures, *Proc.10th Conference of Cognitive Science Society*, pp.503-509, Lawrence Erlbaum Associates, 1988

J. Barnden and K. Srinivas, Overcoming Rule-Based Rigidity and Connectionist Limitations Through Massively Parallel Case-based Reasoning, *International Journal of Man-Machine Studies*, pp.??, 1991

P. Besnard, *An Introduction to Default Logic*, Springer-Verlag, 1989

G. Blelloch, AFL-1: A Programming Language for Massively Parallel Computers, Master's Thesis, MIT AI Lab, 1986

L. Bookman, A Connectionist Scheme For Modeling Context, In D. Touretzky et al. eds. *Proc.1988 connectionist Summer School*, pp.281-290. San Mateo, CA: Morgan Kaufman, 1989

R. Brachman, "I lied about the trees", *The AI Magazine*, pp.80-93, Fall 1985

B. Buchanan and E. Shortliffe,(eds.) *Rule-Based Reasoning: the Mycin Experiment*, Addison-Wesley, 1984

R. Carnap, *The Logic Structure of the World*, University of California Press, 1969

C. Chang and R. C. Lee, *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, 1973

J. Chomsky, Rules and Representation, *Behavioral and Brain Sciences*, pp.1-16, 1980

A. Collins and J. Loftus, Spreading Activation theory of semantic processing, *Psychological Review*, vol.82, pp.407-428, 1975

A. Collins, Fragments of a Theory of Human Plausible Reasoning, Proc.TINALP, pp.194-201, 1979

A. Collins and R. Michalski, The Logic of Plausible Reasoning: A Core Theory, *Cognitive Science*, Vol 13, No 1, pp.1-49, 1989

M. J. Creswell and G. E. Hughes, *An Introduction to Modal Logic*, Routledge, 1968

D. Davidson, Action, Reasons, and Causes, *Journal of Philosophy*, pp.??, 1963

J. Davidson and R. Davidson, *The Psychology of Consciousness*, Plenum Press, 1980

J. de Kleer and J. Brown, Theories of Causal Ordering, *Artificial Intelligence*, 29, pp.33-61, 1986

D. Dennett, Cognitive Wheels: the Frame Problem of AI, Technical Report, Tufts University, 1983

M. Derthick, Mundane reasoning by parallel constraint satisfaction, TR CMU-CS-88-182, Carnegie-Mellon University, 1988

H. Dreyfus, *What Computers Can't Do*, Harper and Row, 1972

H. Dreyfus and S. Dreyfus, *Mind Over Machine*, The Free Press, 1987

C. Dolan and P. Smolensky, Implementing a connectionist production system using tensor products, In D. Touretzky et al. eds. *Proc.1988 connectionist Summer School*, pp.265-272. San Mateo, CA: Morgan Kaufman, 1989

D. Dubois and H. Prade, An Introduction to Prossibilistic and Fuzzy Logics, in: P. Smets et al (eds.) *Non-Standard Logics for Automated Reasoning*, Academic Press, 1988

B. Duval and Y. Kodratoff, Automated Deduction in an Uncertain and Inconsistent Data Basis, *Proc.European Conference of AI*, 1986

M. Dyer, Distributed symbol formation and processing in connectionist networks, *J. Expt. Theor. Artificial Intelligence*, 2, pp.215-239, 1990

R. Fagin, J. Halpern and N. Megiddo, A Logic for Reasoning about Probabilities, *Information and Computation*, Vol.87, No.1/2, pp.87-128, 1990

L. Fang and W. Wilson, A Study of Sequence Processing on Neural Networks, *Proc.IJCNN*, Washington DC, p.588, IEEE Press, June, 1989

M. Fanty, Learning in structured connectionist networks, TR-252, Department of Computer Science, University of Rochester, 1988

J. Feldman, Neural Representation of Conceptual Knowledge, Technical Report 189,Department of Computer Science, University of Rochester, 1986

J. Feldman and D. Ballard, Connectionist models and their properties, *Cognitive Science*, pp.205-254, July 1982

J. Fodor and Z. Pylyshyn, Connectionism and Cognitive Architecture: A Critical Analysis, in: Pinker and Mehler (eds.) *Connections and Symbols*, MIT Press, 1988

L. Fu, Recognition of Semantically Incorrect Rules, *Proc.IEA/AIE-90*, pp.??, 1990

S. Gallant, Connectionist Expert Systems, *Communication of ACM*, V.31(2), pp.152-169, 1988

J. Gelfand, D. Handelman and S. Lane, Integrating Knowledge-based Systems and Neural Networks for Robotic Skill Acquisition, *Proc.IJCAI*, pp.193-198, Morgan Kaufman, 1989

V. Girotto and P. Legrenzi, New European Contributions to the Psychology of Reasoning, *GOLEM*, pp.103-118, 1990

S. Grossberg, *The Adaptive Brain*, North-Holland, 1987

J. Halpern, The Relationshps between Knowledge, Belief and Certainty, IBM Research Report 64833, International Business Machine Corp., 1989

S. Hanson and D. Barr, What does the connectionist model learn: learning and representation in connectionist models, *Behavior and Brain Sciences*, 13(3), pp.1-54, 1990

S. Harnad, The symbol grounding problem, *Physica D*, 42(1-3), pp.335-346, 1990

P.J. Hayes, In defence of logic, *Proc.5th IJCAI*, pp.559-565, Morgan Kaufman, 1977

F. Hayes-Roth, D.A. Waterman and D.B. Lenat, eds. *Building Expert Systems*, Addison-Wesley, 1983

D. Heckerman, Probabilistic Interpretation for MYCIN's certainty Factors, in: L.N. Kanal & J. F. Lemmer (eds.) *Uncertainty in Artificial Intelligence*, pp.167-196, Elsevier Science Publishers, 1985

D. Heckerman, Modularity of Rule-Based System, *Proc.AAAI*, pp.??, Morgan Kaufman, 1987

C. Hempel, *Aspects of Scientific Explanation*, The Free Press, 1965

J. Hendler, Marker Passing and Microfeature, *Proc.10th IJCAI*, pp.151-154, Morgan Kaufman, 1987

R. Hink and D. Woods, How Human Process Uncertain Knowledge, *AI Magazine*, No.8, pp.41-53, 1987

D. Hofstadter, *Metamagical Themas*, Basic Books, 1985

J. Holland, Escaping Brittleness, in: R. Michalski,J. Carbonell and T. Mitchell eds. *Machine Learning (Vol.2)*, pp.593-625, Morgan Kaufman, 1986

J. Holland, N. Nisbitt, T. Thagard and J. Holyoak, *Induction: A Theory of Learning and Development*, MIT Press, 1986

J. Hopcroft and J. Ullman, *Introduction to Automata, Language and Computation*, Addison-Weslley, 1979

D. Hume, *A Treatise of Human Nature*, ???, 1740

D. Israel, What's wrong with non-monotonic logic? in: Ginsberg (ed.) *Readings in Non-monotonic Reasoning*, pp.53-55, Morgan Kaufman, 1987

Y. Iwasaki and H. Simon, Causality in device behavior, *Artificial Intelligence*, 29, pp.3-32, 1986

K. S. Johannessen, Rule-following and intransitive understanding, in: Artificial Intelligence, Culture and Language, Springer-Verlag, 1990

P. Johnson-Laird and P. Wason, eds. *Thinking*, Cambridge University Press, 1977

E. Kandel and J. Schwartz, *Principle of Neural Science*, 2nd Ed. Elsevier, 1984

I. Kant, *A Critique of Pure Reason*, McMillan and Co., 1953

J. Klahr et al, *Production System Models of Learning and Development*, MIT Press, 1989

B. Kosko, Fuzzy Cognitive Maps, *International Journal of Man-Machine Studies*, 24, pp.65-75, 1986

B. Kosko, Hidden Patterns in Combined and Adaptive Knowledge Networks, International *Journal of Approximate Reasoning*, No.2, pp.??, 1988

S. Krause, D. Lehmann and M. Magidor, Nonmonotonic reasoning, preferential models and cumulative logics, *Artificial Intelligence*, 44, pp.167-207, 1990

B. Kuipers, A. Moskowitz and J. Kassirer, Critical Decision under Uncertainty: Representation and Structure, *Cognitive Science*, 12, pp.177-210, 1988

G. Lakoff, *Women, Fires and Dangerous Things*, MIT Press, 1986

T. Lange and M. Dyer, Frame selection in a connectionist model, *Proc.11th Cognitive Science Conference*, pp. 706-713, Lawrence Erlbaum Associates, 1989

H. Leonard, *Principle of Reasoning*, Dover, 1967

D. Lewis, Causation, in: E. Sosa, (ed.) *Cause and Conditionals*, Oxford University Press, 1975

R. Michalski and P. Winston, Variable Precision Logic, *Artificial Intelligence*, 29, pp.121-146, 1986

J. Mackie, Causes and Conditions, in: E. Sosa, (ed.) *Cause and Conditionals*, Oxford University Press, 1975

S. Marcus, (ed.) Special Issue on Knowledge Acquisition, *Machine Learning*, Vol.3, 1989

J. McCarthy, Circumscription –a form of non-monotonic reasoning, *Artificial Intelligence*, 13, pp.27-39, 1980

W. McCulloch and W. Pitts, A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bull. Math. Biophy.*, 1953

D. McDermott and J. Doyle, Non-monotonic Logic I, *Artificial Intelligence*, 13 (1, 2), pp.41-72, 1980

D. McDermott, Non-monotonic Logic II: Nonmonotonic Modal theories, *Jounal of ACM*, 29(1), pp.33-57, 1982

H. Mili and R. Rada, Inheritance generalized to fuzzy regularity, *IEEE Trans.SMC*, Vol.20 No.5, pp.1184-1197, 1990

M. Minsky, *The Society of Mind*, Simon and Schuster, 1985

M. Mitchell and D. Hofstadter, The Role of Computational Temperture, *Proc.11th Cognitive Science Society Conference*, pp.765-772, Lawrence Erlbaum Associates, 1989

R. C. Moore, Semantical Considerations on nonmonotonic logic, *Artificial Intelligence*, 25, pp.75-94, 1985

R. Neal, Learning Stochastic Feedforward Networks, TR CRG-TR-90-7, University of Toronto, 1990

A. Newell and H. Simon, *Human Problem Solving*, Prentice-Hall, 1972

N. Nilsson, *Principle of Artificial Intelligence*, Tiogo, 1980

N. Nilsson, Probabilistic Logic, *Artificial Intelligence*, 28, pp.71-87, 1986

L. Norman, *Human Information Processing*, Academic Press, 1977

R. Nosofsky, Attention, similarity and the identification-categorization relationship, *Journal of Experimental Psychology:General*, Vol.115 No.1, 39-57, 1986

G. C. Oden and R. Jenison, Fuzzy implication formation in disributed cognitive modeling, *Proc. Cognitive Science Society Conference*, pp.860-867, Lawrence Erlbaum Associates, 1990

J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufman, 1988

D. Perlis, On the consistency of commonsense reasoning, *Computational Intelligence*, 2, pp.180-190, 1986

S. Pinker and J. Mehler (eds.), *Connections and Symbols*, MIT Press, 1988

S. Pinker and A. Prince, On Language and Connectionism, in: S. Pinker and J. Mehler (eds.), *Connections and Symbols*, MIT Press, 1988

S. Pinker and A. Prince, On Language and Connectionism, in: *Parallel Distrbuted Processing: its implication*, MIT Press, 1990

J. Pollack, Recursive Auto-Associative Memory, *Proc.Cognitive Science Conference*, pp.33-39, Lawrence Erlbaum Associates, 1988

M. Posner, (ed.) *Foundations of Cognitive Science*, MIT Press, 1989

J. Pustejovsky, *Generative Lexicon*, MIT Press, 1991

R. Reiter, A Logic for Default Reasoning, *Artificial Intelligence*, 13, pp.81-132, 1980

C. Riesback and R. Schank, *Inside Case-based Reasoning*, Lawrence Erlbaum Associate, 1989

D. Rumelhart, J. McClelland and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, MIT Press, 1986

D. Rumelhart, J. McClelland and R. Williams, Learning Internal Representation by Error Propagation, in: D. Rumelhart, J. McClelland and the PDP research Group, *Parallel Distributed Processing*, MIT Press, 1986

W. Salmon, *Scientific Explanation and Causal Structure of the World*, Princeton University Press, 1984

G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1974

R. Schank, Conceptual Dependency: A Theory of Natural Language Understanding, *Cognitive Psychology*, 3(4), pp.552-632, 1972

M. Schoppers, In defense of reaction plans as caches, *AI Magazine*, Vol.10, No.4,pp.51-60, Winter 1989

L. Shastri, A Connectionist Approach to Knowledge Representation and Limited Inference, *Cognitive Science*, 12, pp.331-392, 1988

L. Shastri and V. Ajjanagadde, From simple association to systematic reasoning, Tech Report MS-CIS-90-05, University of Pennsylvania, 1990

Y. Shoham, Reason about Change: Time and Causation from the standpoint of Artificial Intelligence, Ph.D Dissertation, Computer Science Department, Yale University, 1987

Y. Shoham, Non-monotonic Reasoning and Causation, *Cognitive Science*, pp.213-252, 1990

Y. Shoham and T. Dean, Temporal Notation and Causal Terminology, *Proc.9th Cognitive Science Conference*, pp.90-99, Lawrence Erlbaum Associates, 1985

T. Shultz et al, Managing Uncertainty in Rule-Based Reasoning, *Proc.11th Cognitive Science Conference*, pp.227-234, Lawrence Erlbaum Associates, 1989

H. Simon, Causal Ordering and Identifiability, in: D. Lerner (ed), *Cause and Effect*, The Free Press, 1965

P. Smith and O.R. Jones, *The Philosophy of Mind*, Cambridge University Press, 1986

P. Smolensky, On variable binding and representation of symbolic structure, Tech Report, University of Colorado, Boulder, 1987

P. Smolensky, On the proper treatment of connectionism, *Behavioral and Brain Sciences*, 11, pp.1-43, 1988

E. Sosa, (ed.) *Cause and Conditionals*, Oxford University Press, 1975

J. Sowa, *Conceptual Structure*, Addison-Wesley, 1982

R. Sun, The logic for approximate reasoning combining probability and fuzziness, 1st Congress of International Fuzzy System Association, Spain, 1985

R. Sun, E. Marder and D. Waltz, Model local neural networks in the lobster stomatogastric ganglion, *Proc.IJCNN*, p.II-604, IEEE Press, 1989

R. Sun, A discrete neural network model for conceptual representation and reasoning, *Proc.11th Cognitive Science Society Conference*, pp.916-923, 1989

R. Sun, Designing inference engines based on a discrete neural network model, *Proc.IEA/AIE*, ACM Press, p.1094, 1989

R. Sun and E. Schalit, Is the tag necessary? *Brain and Behavior Sciences*, 12 (3), p.145, 1989

R. Sun and D. Waltz, Neural networks and human intelligence, *Journal of Mathematical Psychology*, Vol.34, No.4, pp.483-488, 1990

R. Sun, Rules and Connectionism, *Proc.INNC-Paris*, p.545, Kluwer, 1990

R. Sun, Reasoning at Conceptual and Subconceptual Level: a Connectionist Model, TR 90-136, Brandeis University, Computer Science Dept., 1990

R. Sun, Integrating Rules and Connectionism for Robust Reasoning, Proc.AAAI Workshop on Integrating Neural and Symbolic Processes, 1990

R. Sun, The Discrete Neuronal Models, *Proc.INNC-Paris*, pp.902-907, Kluwer, 1990

R. Sun, The Discrete Neuronal Models and the Discrete Neuronal Models, in: B. Soucek ed. *Neural and Intelligent System Integration*, John Wiley & Sons, 1991

R. Sun and D. Waltz, Neurally Inspired Massively Parallel Model of Rule-Based Reasoning, in: B. Soucek ed. *Neural and Intelligent System Integration*, John Wiley & Sons, 1991

R. Sun, Chunking and Connectionism, *Neural Network Review*, Vol.4, No.2, pp.76-78, 1991

R. Sun, Reasoning, Causality and Connectionist Models, TR-CS-91-155, Brandeis University, 1991

P. Suppes, *A Probabilistic Theory of Causation*, North Holland, 1970

R. Sutton, Learning of World Models by Connectionist Networks, *Proc.9th Cognitive Science Conference*, pp.54-64, Lawrence Erlbaum Associates, 1985

P. Thagard and K. Holyoak, How to compute semantic similarity, *Proc.DARPA Case-Based Reasoning Workshop*, pp.85-88, 1989

D. Touretzky and G. Hinton, Symbols among neurons, *Proc.9th IJCAI*, pp.238-243, Morgan Kaufman, 1987

D. Touretzky, Representing and transforming recursive objects in a neural network, *Proc.IEEE Neural Network Conference*, pp.12-16, IEEE Press, 1986

S. Turkle, AI and Consciousness, *Daedalus*, 1987

A. Tversky, Features of Simiarity, *Psychological Review*, 84(4), pp.327-352, 1977

A. Tversky and D. Kahneman, Extensional Versus Intuitive Reasoning: The Conjuction Fallacy in Probability Judgement, *Psychological Review*, pp.439-450, 1983

M. Waldmann and K. Holyoak, Can causal induction be reduced to associative learning? *Proc.12th Cognitive Science Sociaty Annual Conference*, pp.190-197, Lawrence Erlbaum Associates, 1990

D. Waltz and J. Pollack, Massive Parallel Parsing, *Cognitive Science*, 9, pp.51-74, 1985

D. Waltz and J. Feldman, Connectionist Models and Their Implications, in D. Waltz and J. Feldman (eds.), *Connectionist Models and Their implications*, pp.1-7, Ablex, 1986

D. Waltz, Connectionist models: not just a notational variant, not a panacea, Proc.TINALP, pp.1-8, 1988

D. Waltz, Is indexing used for retrieval? *Proc.DARPA Case-Based Reasoning Workshop*, pp.41-45, Morgan Kaufman, 1989

M. Waterman, *An Introduction to Expert Systems*, Addison-Wesley, 1985

D. Wilkins & K. Tan, Knowledge base refinement as improving an incorrect, inconsistent and incomplete domain theory, *Proc.6th Workshop on Machine Learning*, pp.332-337, Morgan Kaufman, 1989

Y. Wilks, What Sort of Taxonomy of Causation Do We Need for Language Understanding? *Cognitive Science*, I, pp.235-264, 1977

R. Yager, Inheritance and possibility, *IEEE Trans.SMC*, pp.248-252, 1989

L. Zadeh, Fuzzy Sets, *Information and Control*, 8, pp.338-353, 1965

L. Zadeh, The Concept of a Linguistic Variable and Its Application to Approximate Reasoning, *Information Science*, 8(3), pp.199-249; 8(4), pp.301-357; 9(1), pp.43-80, 1975

L. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems*, 1, pp.3-28, 1978

L. Zadeh, Test-score semantics for natural languages and meaning-representation via PRUF, in: B. Rieger, (ed.) *Empirical Semantics*, Bochum: Brockmeyer, pp.281-349, 1981

L. Zadeh, Commonsense knowledge representation based on fuzzy logic, *Computer*, Vol.16, No.10, pp.61-66, 1983

L. Zadeh, Fuzzy Logic, *Computer*, Vol.21, No.4, pp.83-93, April 1988