FINITE DEFAULT THEORIES

by

DAVID WILLIAM ETHERINGTON

B.Sc., The University of Lethbridge, 1977

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Department of Computer Science)

We accept this thesis as conforming

to the required standard.

THE UNIVERSITY OF BRITISH COLUMBIA

AUGUST, 1982

© David William Etherington, 1982

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Computer Science

The University of British Columbia
1956 Main Mall
Vancouver, Canada
V6T 1Y3

Date 1 September 1982

## Abstract

The thesis presents a survey of formalisms for non-monotonic reasoning, providing a sketch of the "state of the art" in the field. Reiter's logic for default reasoning is discussed in detail. Following this, a procedure which can determine the extensions of general finite default theories is demonstrated.

The potential impact of this procedure on some of the other research in the field is explored, and some promising areas for future research are indicated. Grounds for cautious optimism about the tractibility of default theories capable of representing a wide variety of common situations are presented.

Table of Contents

Table of Figures

## Acknowledgments

The material in this thesis owes much to long hours of fruitful discussions with Ray Reiter and Bob Mercer. They, together with Paul Gilmore, Akira Kanda, and Janine Etherington, also made many helpful comments on earlier drafts of some sections. Dave Touretzky provided insight into the problems introduced by cancellation in inheritance structures, and into the workings of NETL. Janine provided the perspective required to see the enterprise through.

The financial support of a Natural Science and Engineering Research Council of Canada postgraduate scholarship is also gratefully acknowledged.

# Chapter 1

# Introduction

Artificial Intelligence (AI) researchers have shown a great deal of interest in the ability to deal with incomplete or inconsistent information. Much of human intelligent behaviour apparently derives from this capacity to draw conclusions in the absence of total knowledge. This suggests that existing approaches to reasoning and knowledge representation should be carefully re-examined to capitalize on their strengths and redress their weaknesses in this area. This thesis examines several approaches, focussing on the way each attempts to deal with these problems. Default Logic (Reiter 1980) provides a suitable framework for studying many knowledge representation issues. A tractible means for applying Default Logic to many common problem domains, such as semantic networks and databases, is presented.

## 1.1 Incomplete Knowledge

It has been argued that if all of the facts which impinge on any given decision had to be considered, human reasoning – as it is commonly experienced – would be impossible. For example, the sheer volume of things which are _not_ true about the world has forced many to treat much of this knowledge as implicit — so that it need only be represented or manipulated when necessary.

Semantic networks turn this necessity into a virtue. If large amounts of information can be excluded from the knowledge base, the interactions between facts during the "deduction" process are restricted. So constraining the "combinatorial explosion" may result in significant improvements in speed.

In database theory, it is also common not to explicitly represent much of the negative information about a domain (_eg_ the cities a given airline flight does _not_ connect). The implication is that anything not (stated to be) true must be false (_cf_ [Codd 1972, Reiter 1981a]).

The problem of representing and using "negative information" manifests itself in a variety of other ways including, in a slightly different form, the celebrated "frame problem" (McCarthy and Hayes 1969, Raphael 1971). Many ideas have been proposed to help reduce the amount of negative information which must be dealt with. Most provide mechanisms embodying the rule: "If you do not 'know' **X** then infer ¬**X**". This rule hinges on an assumption like: "If **X** were true, I would know **X**", or "Everything about the world is known" (_cf_ Reiter 1978).

Another form of reasoning with incomplete information is characterized by statements such as: "Most A's are B's" or "Typical A's are B's". Such statements capture the intuition underlying network representations of knowledge, that a great deal of knowledge about the world involves "prototypes" of members of classes. Great economies of description can be obtained in this fashion. It becomes unnecessary to attach all of the properties associated with members of a class to every member. When information is required about a particular member of a class, it can be obtained by consulting the description of the class prototype.

Prototypic descriptions are often called "default descriptions" (Reiter 1978a), since they tend to be applied only in the absence of contradictory evidence. For example, one can predict that Clyde the elephant is gray without seeing him or being told his colour, but one must be able to consistently deal with the fact that Fred – while an elephant – is white (or pink). There is a large body of literature detailing attempts to take advantage of the representational power of prototypic descriptions, some of which is discussed in the following chapter.

Prototypic reasoning provides tools which make it easier to deal with the imprecision encountered in everyday life. Unfortunately, these tools are not without their own problems. Admitting imprecision into one's reasoning mechanisms is an open invitation to imprecision in one's conclusions. This may manifest itself in situations in which there are no grounds for deciding between alternatives, or where conclusions drawn are inconsistent with further specification of the situation being reasoned about. (The latter condition is a consequence of <u>non-monotonicity</u>, which is discussed in detail in Chapter 2.)

## 1.2 Overview

Chapter 2 considers several aspects of the problem of dealing with incomplete information, together with some partial solutions which have been proposed. The latter range from the very formal to the ad hoc. One of the more formal, "Default Logic" (Reiter 1980), is examined in detail. This discussion is intended to provide the reader familiar only with first-order logic with the concepts necessary to understand the development of the major contribution of this thesis. This occurs in Chapter 3, which presents a mechanism for determining the beliefs sanctioned by particular default theories.

Chapter 4 outlines directions in which this research might be extended. The discussion breaks down into two categories: problems specific to default logic, and problems of determining the interrelationships between default logic and other work in the field. It will become apparent that the separation is somewhat arbitrary; results in either area are likely to have a significant impact in both.

Chapter 5 evaluates the significance of the results presented. The effects of the restrictions placed on the allowable class of theories is discussed. This is followed by some speculation concerning the possibility of obtaining more powerful results along similar lines. The thesis concludes with a short summary.

The material presented is relatively self-contained, assuming only a familiarity with traditional first order logic (see

[Mendelson 1964] for an introduction). The later chapters, however, draw freely on concepts presented earlier. Readers familiar with non-monotonic reasoning in general, and with default logic in particular, may wish to merely skim Chapter 2. Those interested in an only modestly technical overview of non-monotonic reasoning may wish to restrict themselves to that chapter. Finally, the proofs of the theorems presented in Chapter 3 may be skipped without loss of continuity, provided the theorems themselves are understood.

# Chapter 2

# Non-Monotonic Reasoning

> Traditional logics suffer from
> the 'Monotonicity Problem'
> — Drew McDermott

In traditional logical systems, extending a set of axioms, S, can never prevent the derivation of conclusions derivable from S alone. More formally, if S and S' are arbitrary sets of formulae then:

$$S \subseteq S' \Rightarrow \{w \mid S \vdash w\} \subseteq \{w \mid S' \vdash w\}.$$ [1]

The addition of formulae to a set <u>monotonically</u> increases what can be proved from that set, hence the logics are sometimes called <u>monotonic</u>.[2]

More recently, it has been noted that monotonic logics seem inadequate to capture the tentative nature of human reasoning. Since peoples' knowledge about the world is necessarily incomplete, there will always be times when they will be forced to draw conclusions based on an incomplete specification of pertinent details of the situation. Under such circumstances,

---

[1] $S \vdash w$ means w is provable from premises S.

[2] This property has been called the <u>extension property</u> by Hayes (1973) and others, but the former term appears to be gaining pre-eminence. For this reason, and because "extension" has been used in so many ways in the related literature, "monotonic" will be used in this discussion.

assumptions are made (implicitly or explicitly) about the state of the unknown factors. Because these assumptions are not irrefutable, they may have to be withdrawn at some later time should new evidence prove them invalid. If this happens, the new evidence will prevent some assumptions from being used; hence all conclusions which can be arrived at only in conjunction with those assumptions will no longer be derivable. This causes any system which attempts to reason consistently using assumptions to exhibit non-monotonic behavior.

In AI, attempts to solve the problems presented by incomplete information have fallen into two categories. The first category includes those which assume that all of the relevant positive information (eg which individuals exist, which predicates are satisfied by which individuals) is known. From this assumption, it follows that anything which is not "known" to be true must be false. Negative facts[3] can thus be omitted, since they can be inferred from the absence of their positive counterparts. Such assumptions typify PLANNER's "THNOT" (Hewitt 1972) and related negation operators in AI programming languages (Reiter 1978a), as well as Predicate Completion (Clark 1978), Circumscription (McCarthy 1980) and the Closed World Assumption (Reiter 1978), all of which are discussed below, and many others.

In contrast, many have wanted to represent and use what would generally described as "default" or "prototypic" information. Defaults are used to fill gaps in knowledge. In the absence

---

[3] A fact is negative iff all of the literals in its clausal form are negative.

of specific evidence, they allow a system to make (hopefully) enlightened "guesses" instead of reserving judgement or assuming that whatever is unknown is false. Non-Monotonic Logic (McDermott and Doyle 1980), Default Logic (Reiter 1980), Truth Maintenance Systems (Doyle 1979, McAllester 1978, 1980), and various frame-based procedural knowledge representation schemes (Minsky 1975) all embody this idea.

The two approaches are not mutually exclusive — in some instances, at least, the latter subsumes the former — but comparisons of their power are most notable for their absence from the literature. The discussion in the remainder of this chapter does not provide such a comparison, although some points of correspondence are indicated.

## 2.1 Negation As Failure To Derive

Negative facts — those which state what is NOT true about the world — vastly outnumber positive facts. For example, in a discussion at a sufficiently high level, everything which is at some place is NOT at EVERY other place. Similarly, if Tumnus is a cat, he is not a dog, fish, tree, etc. The amount of negative information about a world increases geometrically with the size of the Herbrand Universe.[4] One would like to avoid having to represent all such information. The information must remain

---

[4] The Herbrand Universe is the set of all ground terms constructable from the constants and other function symbols.

available, however — at some point it may become useful to know that Tumnus is not a dog.

AI programming languages (eg PROLOG [Roussel 1975], PLANNER) have often addressed this problem by representing only positive information, assuming that if something cannot be shown to be true it must be false. Such systems embody an inference rule of the form:

If $\nvdash$ P then infer $\vdash \neg$P

which can be paraphrased as "If P is not provable from the database, assume ¬P as a lemma of the database". A derivation of ¬P thus consists of an unsuccessful exhaustive search for a derivation of P. This technique is called negation as failure (NAF).

Despite its attractiveness as a means of implicitly representing negative knowledge, NAF is not without shortcomings and pitfalls. The most obvious of these is that there is no room for incomplete knowledge in such systems — anything which is not known will be assumed false. Reiter (1978) calls this assumption of "total knowledge about the domain being represented" the Closed World Assumption (CWA), since it implies a closed domain in which all possibilities are known. To see the problems presented by incomplete information, consider a database consisting of only BLOCK(A) $\vee$ BLOCK(B). Since it is impossible to derive either BLOCK(A) or BLOCK(B), NAF allows the derivation of ¬BLOCK(A) and ¬BLOCK(B). It is easy to see that such situations are not consistent with NAF.

The fact that some classically consistent databases are not consistent with the CWA leads to the question, "Under what circumstances can the negation as failure inference rule (and hence the CWA) be consistently employed?" There is no complete characterization of suitable databases, but it has been shown that a sufficient condition is that the database be Horn[5] and consistent. Purely negative information plays no part in closed-world query evaluation for such databases. It can be ignored without loss of deductive power (Reiter 1978).

Even though a particular database may be consistent with the CWA, implementations using NAF are not guaranteed to be complete. Answers to a query, although implied by the database, may be missed. It has been noted, for example, that NAF must be applied only to ground literals if inferences drawn are to be correct (Clark 1978, Reiter 1978). Significantly, neither PROLOG nor PLANNER make such stipulations. The following example illustrates the problems that this may cause. For the PROLOG database:

```
EQUAL(x,x).
BLOCK(A).
BLOCK(B).
TWO-BLOCKS(x,y) <- ¬EQUAL(x,y) & BLOCK(x) & BLOCK(y).
```

the query:

```
<- TWO-BLOCKS(x,y).
```

always fails. PROLOG's selection rule, always resolve on the leftmost literal first, causes the goal ¬EQUAL(x,y) to be selected with the variables x and y unbound. Thus the subgoal EQUAL(x,y) is not ground and can be proved with the unifying

---

5 A database is <u>Horn</u> if and only if each clause in the clausal form has at most one positive literal.

substitution 'x for y'. No failure proof for ¬EQUAL(x,y) will be found, even though one exists. ¬TWO-BLOCKS(x,y) may be inferred despite the fact that this is inconsistent with the database.

It should be remembered that these are problems with particular implementations of NAF, not with NAF itself. The requirements for the correct functioning of such systems have been clearly laid out. Unfortunately, correctness and completeness are often sacrificed for the sake of sizeable gains in speed. It is assumed that incompleteness is better than inefficiency. Whether this assumption is appropriate depends on the application, and the fact that an implementation is incomplete may not be known to its users.

## 2.2 Database Completion

NAF allows one to act on the assumption that "the objects that can be shown to have a certain property, P, by reasoning from certain facts, A, are all the objects that satisfy P" (McCarthy 1980). It does not, however, allow the reasoner to derive this assumption. Such systems can never be "conscious" of the underlying principles which they are implicitly assuming. Clark(1978) remedies this shortcoming by making the completeness assumptions explicit in the database. All of the information about a particular relation in DB is gathered together and a completion axiom is added which states that a particular tuple satisfies the relation only if DB says it must. Applying this process to all of the relations in DB yields the <u>completed database</u> (C(DB)). This completion of the database makes explicit the assumptions of total world knowledge. For example, if the database contains:

$$ON(A,B) \quad \text{and} \quad ON(B,C) \tag{1}$$

and no general rules about the ON relation, then the completion axiom is:

$$\forall xy.[ON(x,y) \supset (x=A \wedge y=B) \vee (x=B \wedge y=C)] \tag{2}$$

Combining (1), (2), and inequality schemata stating that different names denote different objects results in the conclusion that (A,B) and (B,C) form the complete extension for ON.

Reiter (1981a) explores the effects of adding completion axioms to normal relational databases. He demonstrates that such techniques provide a means for dealing with types of incomplete information commonly encountered in the database field, such as null values and disjunctive information.

Database completion is more powerful than a first order system augmented by NAF. Clark shows that the structure of a failure proof is always isomorphic to that of a first order proof from the completed database. Conversely, the completion of the database:

$$DB = \{ P(a) \}$$

is:

$$C(DB) = \{ \forall x.[P(x) <=> x=a] \}$$

from which $\forall x.[x \neq a \supset \neg P(x)]$ follows by first-order reasoning.

For any particular $x \neq a$, NAF can show $\neg P(x)$, but the universal summary is beyond its capabilities.

Another advantage of database completion is that it does not introduce inconsistencies when applied to databases containing incomplete information. The database:

$$BLOCK(A) \vee BLOCK(B),$$

which is inconsistent with the CWA, can be rewritten as:

$$\forall x.[\neg BLOCK(A) \wedge x=B \supset BLOCK(x)] \quad \text{and:}$$

$$\forall x.[\neg BLOCK(B) \wedge x=A \supset BLOCK(x)]$$

From these, the consistent completed database:

$$\forall x.[BLOCK(x) <=> (\neg BLOCK(A) \wedge x=B) \vee (\neg BLOCK(B) \wedge x=A)]$$

can be derived. Notice that the disjunction in the original database has become an "exclusive or" in the completed database, which states that there is exactly one block, and it must be either A or B.

These techniques do not avoid all of the problems of NAF simply because all of the deductions are first order. There will still be propositions which are undecidable in the completed database, propositions corresponding to those for which the exhaustive search for a failure proof never terminates.

## 2.3 Circumscription

McCarthy (1980) has proposed a rule of conjecture called "predicate circumscription". This rule allows explicit completeness assumptions, similar to Clark's completion axioms, to be derived as they are required.

The formal mechanics of circumscription are beyond the scope of this discussion. Essentially, however, circumscription provides a means for closing off the world with respect to a particular predicate at a particular time. A schema for a set of first order sentences is generated. This schema is then instantiated by substituting arbitrary predicates for the predicate variables it contains. The particular substitution(s) chosen determine which individuals are conjectured to be the entire extension of the predicate being circumscribed. McCarthy considers the blocks-world example, discussed previously, in which all that is known is:

$$\mathrm{BLOCK}(A) \vee \mathrm{BLOCK}(B) \quad [6] \tag{1}$$

If the predicate variable, $\Theta$, in the circumscription of (1):

$$[\Theta(A) \vee \Theta(B)] \wedge \forall x.[\Theta(x) \Rightarrow \mathrm{BLOCK}(x)] \Rightarrow \forall x.[\mathrm{BLOCK}(x) \Rightarrow \Theta(x)]$$

is replaced successively by the predicates (2) and (3):

$$x=A \tag{2}$$
$$x=B \tag{3}$$

the conjecture:

$$\forall x.[\mathrm{BLOCK}(x) \supset x=A] \vee \forall x.[\mathrm{BLOCK}(x) \supset x=B] \tag{4}$$

can be derived. As did the completed database, (4) says that there is only one block: A or B. Again, the conjecture closes the world and puts the "exclusive" interpretation on the original disjunction.

The choice of substituends vitally determines what can be

---

6 Recall that this database is NOT consistent with the CWA.

obtained from the circumscription process. It is not clear, in general, how these substituends are to be chosen. McCarthy suggests that the desired goal directs the choice of appropriate substitutions. It remains to be seen whether this is correct.

It is easy to see that Circumscription subsumes NAF (in all of its forms). McCarthy shows that it can derive the induction axiom for arithmetic, which suggests that it is more powerful than database completion.

All of the ideas discussed so far have provided ways of becoming more "closed-minded". Each functions by restricting the set of models for the given axioms. The goal has been to allow only minimal models (Davis 1980), in which only a minimal set of predicate instances necessary to satisfy the axioms is allowed to be true.

The complementary approach also involves restricting the set of models considered. Rather than focussing on minimality, the systems discussed in the sequel provide more flexibility in determining which models are considered "interesting".

## 2.4 Default Logic

In the following, Default Logic (Reiter 1980, 1981) is surveyed in more detail than those paradigms already discussed. This is necessary both to familiarize the reader with the concepts and notation required to understand the results in

Chapter 3, and to place those results in perspective.

### 2.4.1 Default Theories

Default Logic is based on a first order language, $L_A$, consisting of the first order well-formed formulae (wffs) formed from an alphabet, A, consisting of countably many variables: x, y, z, $x_1$, ...; function letters: a, b, c, ...; and predicate symbols: P, Q, R, ...; together with the usual punctuation signs; logical connectives: $\neg$ (not), $\wedge$ (and), $\vee$ (or), $\supset$ (implies); and quantifiers: $\forall x$ (for all x), $\exists x$ (there exists an x).

A wff containing no free variables is said to be closed. Given an arbitrary set of wffs, S, the logical closure of S, $Th_L(S)$, is defined as:

$$Th_L(S) = \{w \mid w \in L, \text{ w is closed}, S \vdash w\}.$$ [7]

A default is any expression of the form:

$$\frac{\mathbf{A}(\bar{x}) : \mathbf{B}_1(\bar{x}), \ldots, \mathbf{B}_m(\bar{x})}{\mathbf{w}(\bar{x})}$$ [8]

where $\mathbf{A}(\bar{x})$, $\mathbf{B}_i(\bar{x})$, and $\mathbf{w}(\bar{x})$ are all wffs whose free variables are among those in $\bar{x} = x_1, \ldots, x_n$. $\mathbf{A}$, $\mathbf{B}_i$, and $\mathbf{w}$ are called the

---

[7] When the alphabet and/or language are clear from context, the subscripts on $L_A$ and $Th_L(S)$ will be omitted.

[8] This notation differs from Reiter's in the omission of the "M" preceeding each of the $\mathbf{B}_i$'s. Since they are implicit in the positional notation, they have been omitted as a notational convenience.

prerequisite, justifications, and consequent of the default, respectively. If none of $\mathbf{A}$, $\mathbf{B}_i$, and $\mathbf{w}$ contain free variables, the default is said to be closed. If the prerequisite is empty, it may taken to be any tautologous proposition. Two classes of defaults having only a single justification, $\mathbf{B}(\bar{x})$, are distinguished. Those with $\mathbf{B}(\bar{x}) = \mathbf{w}(\bar{x})$, are said to be normal, while those with $\mathbf{B}(\bar{x}) = \mathbf{w}(\bar{x}) \wedge \mathbf{C}(\bar{x})$, for some $\mathbf{C}(\bar{x})$, are called semi-normal.

A default theory, $\Delta$, is an ordered pair, (D,W). D is a set of defaults; W is a set of first order formulae. If all of the defaults in D are normal or semi-normal, $\Delta$ is said to be a normal or semi-normal default theory, respectively. If each default of D is closed, $\Delta$ is a closed default theory.

Defaults serve as rules of inference or conjecture, augmenting those normally provided by first-order logic. Under certain conditions, they sanction inferences which could not be made within a strictly first-order framework. If their prerequisites are known and their justifications are "consistent" (ie their negations are not provable), then their consequents can be inferred. Thus the term "justification" is seen to be somewhat misleading, since justifications need not be known, merely consistent.[9] The consequent's status is akin to that of a belief, subject to revision should the justifications be denied at some future time. It is this characteristic which induces the non-monotonic behavior of defaults.

---

[9] In a modal logic with the operator K (know) the justifications $\mathbf{B}_i$ might appear as $\neg K \neg \mathbf{B}_i$.

Default rules can be seen to have a great deal in common with many previously mentioned approaches. For example, the Closed World Assumption states:

If $\nvdash$ w infer $\vdash \neg$w

which can be represented in Default Logic by:

$$\frac{:\neg w}{\neg w} \tag{1}$$

In fact, (1) will later be referred to as the "Closed World" default. The DEFAULT assignments which can be attached to frame slots in KRL (Bobrow and Winograd 1977) also appear to be related. KRL provides a mechanism for obtaining a value for a slot in the absence of a "better" value. A KRL default value, d, for a slot, s, in a frame instance, f, can be viewed as:

If $\nvdash$ s(f)$\neq$d infer $\vdash$ s(f)=d

or, in Default Logic, as:

$$\frac{:s(f)=d}{s(f)=d} \quad .$$

Similar mechanisms are available in many other frame-based knowledge representation schemes (Minsky 1975).

A closely related approach is Sandewall's (1972) "Unless" operator. "Unless(P)" is interpreted as "$\nvdash$ P", and "Unless" terms are allowed in the construction of wffs, with results like:

A $\wedge$ Unless(B) $\supset$ C

which correspond roughly to:

$$\frac{A\ :\ \neg B}{C} \quad .$$

"Unless" was originally proposed as a solution to the frame problem. Rather than having to have explicit axioms stating that the properties of objects remained invariant from situation to situation unless explicitly changed, Sandewall suggested that these "frame axioms" be replaced by a frame inference rule like:

$$\frac{\begin{array}{l}\text{IS(object,property,situation)}\\ \underline{\text{Unless(ENDS(object,property,Successor(situation,act)))}}\end{array}}{\text{IS(object,property,Successor(situation,act))}}$$

which can be interpreted: If an object has a property in a situation, it can be concluded to retain that property in the successor situation resulting from performing 'act', unless it can be shown otherwise.

No formation rules were provided for "Unless", however, so questionable formulae such as:

$A \supset \text{Unless(B)}$

can be constructed. The semantics of such formulae are, at best, difficult to determine. Sandewall also fails to provide any formal understanding of the impact of the "Unless" rule on the underlying logic. Default Logic has, to some extent, remedied these shortcomings.

### 2.4.2 Closed Default Theories and Their Extensions

Reiter (1980) describes the extensions of a default theory, $\Delta$ = (D,W), as being "acceptable sets of beliefs that one may hold about an incompletely specified world, W". D is viewed as extending the first order knowledge of W in order to provide information not derivable from W.

An extension, E, for $\Delta$ is required to have the following properties:

$$W \subseteq E \tag{1}$$

$$Th_L(E) = E \tag{2}$$

For each default, $\dfrac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{w}} \in D$, if $\mathbf{A} \in E$, and $\neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin E$ then $\mathbf{w} \in E$. (3)

These properties state that E must contain all the known facts, that E must be closed under the $\vdash$ relation, and that the consequent of any default whose prerequisite is satisfied by E, and whose justifications are consistent with E, must also be in E. Reiter defines an extension for a closed default theory to be a minimal fixed point of an operator having the above characteristics.

The extensions of a default theory select a restricted subset of the models of the underlying first-order theory, W. Any model for an extension of $\Delta$ will also be a model for W, but the converse is generally not true.

There is an iterative mechanism for verifying an extension. The method is, unfortunately, not suitable for constructing extensions. This is because an oracle is required which can decide whether a wff's negation is in the given extension. The theorem outlining the mechanism is quoted below, both to provide a better intuition about extensions and because it will be drawn on several times in Chapter 3. Note the explicit reference to E in the definition of $E_{i+1}$.

<u>Theorem</u> (Reiter 1980)

Let $\Delta$ = (D,W) be a closed default theory, and E be and arbitrary set of formulae. Define:

$$E_0 = W$$

and, for $i \geq 0$:

$$E_{i+1} = Th(E_i) \cup$$

$$\left\{\mathbf{w} \middle| \frac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{w}} \in D, \mathbf{A} \in E_i, \neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin E\right\}$$

Then E is an extension for $\Delta$ <u>iff</u> $E = \bigcup_{i=0}^{\infty} E_i$.

Default theories need not always have extensions, even when W is consistent. There are, however, certain classes of theories for which the existence of at least one extension is guaranteed. Normal theories have been shown to always have extensions, and it appears that this is also true for certain classes of semi-normal theories (see Chapter 4 for a discussion).

Some examples of defaults were presented in the preceeding section. The following example illustrates the extensions induced by the closed world default on the theory:

$$W = \{ \text{BLOCK(A)} \vee \text{BLOCK(B)} \}.$$

The closed world default is really a default schema which is applicable to any positive ground literal. In this case, it results in the following set of normal defaults:

$$D = \left\{ \frac{:\neg\text{BLOCK(A)}}{\neg\text{BLOCK(A)}} \quad \frac{:\neg\text{BLOCK(B)}}{\neg\text{BLOCK(B)}} \right\}.$$

The theory, (D,W), has two extensions, $E_1$ and $E_2$.

$$E_1 = Th(\{\neg\text{BLOCK(A)}, \text{BLOCK(B)}\})$$

$$E_2 = Th(\{\text{BLOCK(A)}, \neg\text{BLOCK(B)}\})$$

Note that $\bar{E} = Th(\{\text{BLOCK(A)}, \text{BLOCK(B)}\})$ is <u>not</u> an extension. Like database completion and Circumscription, the closed world default gives the exclusive interpretation of disjunctions to which it is applied. Intuitively, this is because the defaults force as many things to be false as possible, resulting in extensions whose models may be minimal models for W. More precisely, $\overline{E}$ is not an extension because it violates the minimality condition of the definition of extensions. (Were W also to contain both BLOCK(A) and BLOCK(B), $\overline{E}$ would be the only extension.)

Notice how the extensions $E_1$ and $E_2$ manifest W's inconsistency with the CWA. The inconsistent assignments for BLOCK(A) and BLOCK(B) are still obtainable, but they are separated into orthogonal, self-consistent extensions. In fact, Reiter has shown that the extensions of any default theory will always be self-consistent provided that the first-order theory W is consistent, and that all the extensions of a normal default theory will be mutually inconsistent.

Reiter gives a proof procedure applicable to normal theories, but no means of determining the extensions of non-normal theories has heretofore been presented. Such a mechanism, applicable to arbitrary finite theories, is presented in Chapter 3.

### 2.4.3 General Default Theories

In contrast to closed defaults, an <u>open</u> default is one in which at least one of $\mathbf{A}(\overline{x})$, $\mathbf{B}_i(\overline{x})$, or $\mathbf{w}(\overline{x})$ contain free variables in $\bar{x}$. Reiter (1980) defines the extensions of an arbitrary default theory as follows:

1) The Skolemized form of a default, $\delta$, is obtained by replacing the consequent of $\delta$ with its Skolemized form (Robinson 1965).

2) A default theory, $\Delta = (D,W)$, is Skolemized if all defaults in D and all wffs in W are in Skolemized form.

3) If $\Delta = (D,W)$ is a Skolemized default theory, $\Sigma$ is the set of Skolem functions of $\Delta$, and F is the set of function letters of the alphabet, A, then CLOSED-DEFAULTS($\Delta$) is defined as:

$$\{\delta(\bar{g}) \mid \delta(\bar{x}) \in D,\ \ \bar{g} \in H(F \cup \Sigma) \text{ is a ground tuple}\}$$

   where H represents the Herbrand Universe.

4) E is an extension for a Skolemized default theory, $\Delta$, iff E is an extension for the theory (CLOSED-DEFAULTS($\Delta$),W).

An open default is interpreted as standing for the set of closed defaults obtainable by replacing its free variables by ground terms. $H(F \cup \Sigma)$ will generally be countably infinite, making (CLOSED-DEFAULTS($\Delta$),W) a default theory with an infinite set of defaults. The repercussions of this will become apparent in Chapter 3.

Most interesting default theories are not closed. Consider what, by now, must be the archetypal default theory:

$$
\begin{aligned}
W = \{ & \forall x.\ \text{Penguin}(x) \supset \text{Bird}(x), \\
& \forall x.\ \text{Penguin}(x) \supset \neg\text{Can-Fly}(x), \\
& \forall x.\ \text{Dead-Bird}(x) \supset \text{Bird}(x), \\
& \forall x.\ \text{Dead-Bird}(x) \supset \neg\text{Can-Fly}(x), \\
& \forall x.\ \text{Ostrich}(x) \supset \text{Bird}(x), \\
& \forall x.\ \text{Ostrich}(x) \supset \neg\text{Can-Fly}(x), \\
& \text{Bird}(\text{Tweety}) \}
\end{aligned}
$$

$$D = \left\{ \frac{\text{Bird}(x) : \text{Can-Fly}(x)}{\text{Can-Fly}(x)} \right\}.$$

The default, which is not closed, might be interpreted as "If x is a bird, and it is consistent that x can fly, conclude that it can". This theory allows one to conclude, for an arbitrary bird (*eg* Tweety), that it can fly — unless one is told that it cannot, or that it is a penguin, an ostrich, or dead. The conclusion may later have to be revoked should Tweety turn out to be a penguin, but common sense seems to sanction the same conclusion. This is partly because people tend to assume that they have the relevant information in most situations (*cf* linguists' use of Grice's Conversational Implicatures [Grice 1975]: one of these is that all information necessary to interpret an utterance is expected to be contained in the utterance.)

## 2.5 Non-Monotonic Logic

McDermott and Doyle (1980, McDermott 1982) propose a formalism complementary to Default Logic, which they call Non-Monotonic Logic (NML). Unlike Default Logic, which uses the notion of consistency only at the "meta" level (in the inference rules), NML centres around the introduction of consistency into the object language. In the first incarnation of NML (McDermott and Doyle 1980), a standard first-order logic was augmented with an "M" operator, roughly equivalent to the familiar "$\nvdash\neg$". The set of theorems was defined as the intersection of all of the fixed points of an operator, NM. Essentially, NM produces the logical closure of the original theory together with as many assertions of the form Mq as possible. The set of theorems can be contrasted with the extensions of a default theory, each of which is a fixed point. This indicates that non-monotonic theoremhood is, in some sense, a more restrictive concept than extension membership. In fact, the two are incomparable in general. The agreement intuitively expected considering that any default:

$$\frac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{w}}$$

can be approximated in NML by:

$$\mathbf{A} \wedge M\mathbf{B}_1 \wedge \ldots \wedge M\mathbf{B}_m \supset \mathbf{w}$$

often occurs. There are, however, default theories which have extensions even though the corresponding non-monotonic theories have no fixed points, and vice versa (see [Reiter 1980] for examples).

Davis (1980) suggests that it might be impossible to assign a reasonable semantics to the M operator were it included in the object language. This and other problems led to the recasting of the theory in terms of a more classical modal logic (McDermott 1982).[10] The resulting non-monotonic S5 is unfortunately redundant, since it is no more powerful than S5. Proofs of the consistency of non-monotonic T and S4 have not been presented. Such proofs are necessary for NML to be successful.

In version II of NML, McDermott considers the advantages of believing formulae other than those in the intersection of all the fixed points. He proposes a "brave robot" which would

---

10 A discussion of modal logics is beyond the scope of this thesis. See [Hughes and Cresswel 1968] for an introduction.

believe all of the formulae of a particular fixed point. Such an approach is required in order to provide an intuitively satisfactory semantics for Mp: "p is consistent with what is believed".

The availability of the "M" terms in the language has advantages and disadvantages. For example, it can be shown that constructs of the form:

$$p \supset Mq$$

where p and q are arbitrary formulae, are either redundant or inconsistent. (This follows because the "theorems" of any NML theory must contain all formulae Mp which are not inconsistent.) Such constructs cannot be formed in default logic, but are readily available in NML (as they are in Sandewall's formalism).

On the positive side, the default rules can be manipulated by the theory. For example, in the normal default theory with no axioms and the defaults:

$$\frac{A : B}{B} \quad , \quad \frac{\neg A : B}{B}$$

nothing can be inferred about B. The corresponding non-monotonic theory:

$$\{A \wedge MB \supset B \quad , \quad \neg A \wedge MB \supset B\}$$

implies MB and MB $\supset$ B, from which B can be inferred. This appears to be more in accord with normal common sense reasoning.

Finally, Lp <=> p is a thesis of NML. While most modal logicians would agree that "p is necessarily true" implies "p is true", the converse is usually not accepted. Hughes and Cresswell (1968, p28) conclude that "no intuitively plausible modal system" would have such a thesis. This indicates that there may be fundamental problems with NML.

## 2.6 Consistency Maintenance

All of the paradigms discussed in this chapter provide means for jumping to conclusions. Jumping to conclusions can be a risky process, since there is an ever-present possibility of refutation. Refutation of a conclusion already entered into the knowledge base has consequences beyond the simple deletion of the offending fact. Every other conclusion reached on the basis of the rejected fact must be sought out and itself rejected. These rejections may have repercussions scattered throughout the knowledge base.

The "reasoner" must either remember the justifications for everything it "believes" and be able to unravel a flawed web of beliefs, or else be prepared to throw everything away and start again from first principles. The latter course is clearly impractical, but the former is by no means trivial. For example, although the justifications which were used for deriving a particular fact may have been invalidated, there may be independent justifications which remain valid. To revoke the fact might recursively force the revoking of a large body of information which will later have to be rederived. Many "belief revision" systems have been designed to study the problems of changing models of situations. Doyle (1979) and McAllester (1978, 1980) present surveys of the problems which must be dealt with, and some suggested approaches.

A discussion of such techniques is outside the intended scope of this thesis, but it should be noted that these issues are real and must be dealt with if any of the material discussed herein is to be practically applicable.

## 2.7 Semantic Networks

Type or IS-A hierarchies are extremely common in AI. In their simplest form, they correspond to logical structures consisting of ground literals and simple, universally quantified, implications of the form $\forall x.[A(x) \supset B(x)]$, where A and B are monadic predicates. "Inferencing" degenerates to path following (computing the transitive closure of a chain of implications), with negation achieved by failure proofs. The logical equivalent of such networks are definite databases.[11] Definite databases are necessarily consistent with the CWA, so the semantics of such networks can be clearly laid out.

In the light of the common need to deal with prototypic descriptions, it is worth considering whether semantic networks can deal with exceptions. Simply incorporating exceptions into the structure described above, as is frequently done, means that a different semantics must be provided. Otherwise, the resulting structures would be inconsistent. Traditionally,

---

[11] A definite database is a Horn database in which each clause has EXACTLY one positive literal.

networkers have circumvented this problem by allowing inconsistent information but using an incomplete inference system incapable of detecting the inconsistencies. Conflicts are thus resolved by giving preference to whatever can be derived via the shortest chain of inferences. Not only is it difficult to determine the semantics of such an inference system, but the results are sometimes counter-intuitive. For example, adding a link which summarizes a chain (perhaps to improve efficiency) might completely change the path lengths and hence what can be derived from the network.

NETL (Fahlman 1979, Fahlman et al 1981), a system designed to allow the construction of large-scale IS-A hierarchies with exceptions (MIGHT-BE-A hierarchies), addresses this problem by making exceptions explicit in the network structure. NETL is intended to be a massively parallel architecture, with one processor assigned to each node of the semantic network it represents. Three of NETL's link types are relevant to this discussion:

1) VC — corresponds to the normal ISA link, except that exceptions (indicated by CANCEL links) are allowed which may disrupt the normal inheritance structures.
Represented graphically by: ——>.

2) CANCEL — cancels inheritance by the tail node and all "lower" nodes of properties associated with the head node and all "higher" nodes.
Represented by: ┼┼┼>.

3) UNCANCEL — undoes the effect of the CANCEL link at its head. The node at the tail and its descendants will inherit as though the CANCEL link were not there.
Represented by: --->.

NETL graphs can be mapped into <u>monadic</u> semi-normal default theories (theories involving only monadic predicates), as follows:

VC links become normal defaults:

A ——> B becomes $\frac{A(x)\ :\ B(x)}{B(x)}$.

CANCEL and UNCANCEL links combine to produce semi-normal defaults. A CANCEL link from A to B on which UNCANCEL links impinge from $C_1$, ..., $C_n$ ($n \geq 0$) becomes:

$$\frac{A(x)\ :\ \neg B(x) \wedge \neg C_1(x) \wedge \ldots \neg C_n(x)}{\neg B(x)}$$

Activation of a node corresponds to asserting a ground instance of that node.

These three links allow the construction of complex structures. For example, Figure 2.1 diagrams the NETL structure and corresponding default theory for the following (Fahlman et al 1981):

A mollusc is a shell-bearer.
Cephalopods are molluscs which are not shell-bearers.
A nautilus is a cephalopod, but <u>is</u> a shell-bearer.

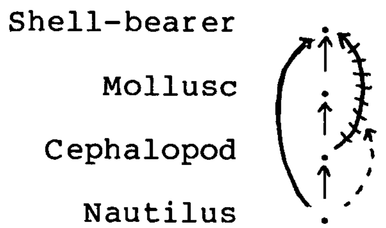



Figure 2.1a — A typical NETL example.

This figure demonstrates that connections can become rather complex in naturally occurring situations. It turns out that inheritance (or inference) in NETL does not have the straightforward semantics one might expect. For example, in Figure

$$\frac{\text{Mollusc}(x) : \text{Shell-bearer}(x)}{\text{Shell-bearer}(x)}, \quad \frac{\text{Cephalopod}(x) : \text{Mollusc}(x)}{\text{Mollusc}(x)}$$

$$\frac{\text{Nautilus}(x) : \text{Cephalopod}(x)}{\text{Cephalopod}(x)}, \quad \frac{\text{Nautilus}(x) : \text{Shell-bearer}(x)}{\text{Shell-bearer}(x)}$$

$$\frac{\text{Cephalopod}(x) : \neg\text{Shell-bearer}(x) \wedge \neg\text{Nautilus}(x)}{\neg\text{Shell-bearer}(x)}$$

Figure 2.1b — Default theory for Figure 2.1a.

2.2a, it is unclear to NETL whether or not A should inherit from E.

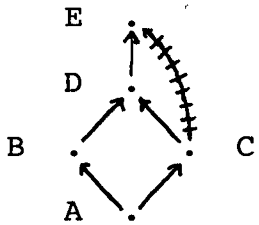



Figure 2.2a — A problematic case.

$$\frac{A(x) : B(x)}{B(x)}, \quad \frac{A(x) : C(x)}{C(x)}, \quad \frac{B(x) : D(x)}{D(x)}$$

$$\frac{C(x) : D(x)}{D(x)}, \quad \frac{D(x) : E(x)}{E(x)}, \quad \frac{C(x) : \neg E(x)}{\neg E(x)}$$

Figure 2.2b — Default theory for Figure 2.2a.

This is because one path (A, B, D, E) indicates "yes" and the other (A, C, E) "no", because of the CANCEL link from C to E. While the former interpretation may seem preferable, since there is a path involving no cancel links, the parallel nature of NETL forces the latter choice. Since this is somewhat counterintuitive, such structures are now declared 'ill-formed' and the user must explicitly resolve the ambiguity. The source of the problem becomes apparent when the corresponding default theory (Figure 2.2b) is considered. This theory has two extensions, containing E and ¬E respectively.

Such difficulties highlight the need for a sound theoretical basis for systems which are intended to represent and use knowledge. Default Logic may provide a suitable vehicle for this work. Chapter 4 surveys this problem in more detail, indicating that the results presented in Chapter 3 may be useful in formalizing the semantics of network representations.

## 2.8 Objections to Non-Monotonic Formalisms

Kramosil (1975) claims to have shown that any formalized theory which allows unprovability as a premise in deductions must either be "meaningless", or no more powerful than the corresponding first order theory without rules involving such premises. He presents two "proofs" to support his claim. Careful examination shows that the first result follows from a definition of "formalized theory" which expressly excludes any theory which exhibits the types of behavior common to non-monotonic theories. The second result is based on an incorrect definition of "proof" and hence of "theoremhood" and is itself meaningless. As the paper stands, it shows only that non-monotonic theories must behave differently than monotonic theories in those cases where the former can derive results unobtainable using the latter.

Kramosil was not the only one to be uncomfortable with opening the "Pandora's Box" of non-monotonicity. Sandewall (1972) notes that the "Unless" operator has "some dirty logical properties". Considering the example:

$$\begin{array}{l} A \\ A \wedge \mathrm{Unless}(B) \supset C \\ A \wedge \mathrm{Unless}(C) \supset B \end{array}$$

he observes that either B and C can be theorems, but, in general, not both simultaneously. Reiter (1978a) makes a similar observation in an early paper, stating that:

> Such behavior [is] clearly unacceptable. At the very least, we must demand of a default theory that it satisfy a kind of 'Church-Rosser' property: No matter what the order in which the theorems of a theory are derived, the resulting set of theorems will be unique.

It appears that the Church-Rosser property is a necessary casualty if non-monotonicity is accepted.

A further problem which must be faced by those embracing non-monotonicity is that the non-theorems of a first order theory are not recursively enumerable. This means that the rules of inference in theories involving the $\nvdash$ operator cannot be effective in general. From this, it follows that the theorems are not recursively enumerable. By contrast, in monotonic logics, the rules of inference MUST be effective and the theorems MUST be recursively enumerable.

Finally, the very non-monotonicity which makes such theories interesting means that "theorems" may have to be retracted if the assumptions on which they are based are refuted (either by new knowledge or changes in the state of the world). To be useful, a non-monotonic reasoning system must be able to remember which assumptions underly each theorem and be able to unwind the potentially complex chain of deductions founded on retracted justifications.

Chapter 3

# Determining the Extensions of Default Theories

## 3.1 Motivation

In the previous chapter, Reiter's Default Logic was examined in some detail. It was originally thought that most commonly-occurring defaults were normal (Reiter 1980) and the results obtained concerning normal theories seemed most encouraging. Reiter and Criscuolo (1981) later observed that interacting defaults could not be satisfactorily dealt with by strictly normal theories. They considered the example:

Typical adults are employed.
Typical high-school dropouts are adults.
Typical high-school dropouts are not employed.

Which could be represented by the following normal theory:

$$\frac{\mathrm{Adult}(x)\ :\ \mathrm{Employed}(x)}{\mathrm{Employed}(x)} \tag{1}$$

$$\frac{\mathrm{Dropout}(x)\ :\ \mathrm{Adult}(x)}{\mathrm{Adult}(x)} \tag{2}$$

$$\frac{\mathrm{Dropout}(x)\ :\ \neg\mathrm{Employed}(x)}{\neg\mathrm{Employed}(x)} \tag{3}$$

Given the fact that John is a dropout, this theory has two distinct extensions, asserting either that John is employed or not employed, respectively. Intuitively, it seems more likely that John is unemployed. Unfortunately, there is no way to force this interpretation on the normal default theory without replacing (3) with a first-order axiom of the form:

$$\forall x.[\mathrm{Dropout}(x) \supset \neg\mathrm{Employed}(x)]. \qquad (4)$$

This fails to capture the tentative nature of the English description — discovering an employed dropout would result in a logical inconsistency. Using this and several other examples, Reiter and Criscuolo illustrate a need for semi-normal defaults such as:

$$\frac{\mathrm{Adult}(x) \ : \ \mathrm{Employed}(x) \wedge \neg\mathrm{Dropout}(x)}{\mathrm{Employed}(x)} \qquad (5)$$

instead of (3) or (4), in order to fully characterize common uses of default reasoning.

Semi-normal default theories can be used to represent many forms of knowledge, including semantic networks, databases, and the interactions between defaults, This will be little more than a logical curiosity, however, if the extensions of the resulting default theories cannot be computed, and the properties of non-normal default theories have so far been largely unexamined. The following section addresses this problem.

## 3.2 A Constructive Mechanism

Reiter provides a test to determine whether a set of formulae is an extension for an arbitrary default theory, but he provides no constructive mechanism which yields the extensions of such theories. Of course, the extensions of default theories are not recursively enumerable in general (Reiter 1980). All that can be hoped for is a procedure which yields the extensions of theories restricted to decidable subcases of the predicate calculus (<u>eg</u> sentential, monadic, or finite theories).

This chapter provides such a mechanism applicable to arbitrary finite default theories. In what follows, the alphabet, A, discussed in Chapter 2, is restricted to contain only finitely many variables, constant symbols, and predicate letters, plus the punctuation signs and logical connectives. No function letters are allowed (except the 0-ary function letters, or constants). The discussion is further restricted by the requirement that for any given default theory, $\Delta$ = (D,W), D must be finite.

The effect of the above restricions is to ensure that the Herbrand Universe associated with A is finite and hence that the closure of $\Delta$ (discussed previously) is finite. The reasons for these restrictions, their repercussions, and the effects of relaxing them are discussed in the following sections.

A definition of an iterative procedure for determining the extensions of an arbitrary finite default theory is now given. This is followed by the derivation of the two halves of a completeness result which shows that the procedure can return all (Theorem 3.6) and only (Theorem 3.4) the extensions of the default theory to which it is applied.

Definition 3.1

Consider an arbitrary closed default theory, $\Delta = (D,W)$. A sequence of "hypothesized extensions", $H_i$, $(i \geq 0)$ is then defined as follows:

$H_0 = Th(W)$

For $j > 0$ define

$h_0^j = Th(W)$ and $GD_0^j = \emptyset$

For $i \geq 0$ let $D_i^j$ be the set of defaults whose preconditions have been satisfied and whose justifications have not been denied. More precisely:

$$D_i^j = \{\delta \mid \delta = \frac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{w}} \in D,$$

$$\mathbf{A} \in h_i^j,\ \neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin (h_i^j \cup H_{j-1})\ \}$$

† Choose $\delta' = \dfrac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{w}}$ arbitrarily from $(D_i^j - GD_i^j)$,

from those defaults eligible to be, but which have not yet been, applied.

Then define:

$$GD_{i+1}^j = GD_i^j \cup \{\delta'\}$$

$$h_{i+1}^j = Th(h_i^j \cup \{\mathbf{w}\})$$

$$H_j = \bigcup_{i=0}^{\infty} h_i^j$$

Observe that $H_j$ is logically closed. Since D is finite, i may be bounded above by the cardinality of D without loss of generality.

The application of the Th( ) operator in the above definition makes it immediately clear that the $H_i$'s will not be well defined in any situation where first-order theoremhood is undecidable. Finite theories, such as those discussed here, are decidable. This should be kept in mind in the following discussions.

The following lemmas will be required in order to obtain the desired completeness result:

Lemma 3.2

For $H_i$ $(i \geq 0)$ defined as in Definition 3.1,

$$\text{if } \delta = \frac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{w}} \in D,$$

$$\text{then } \mathbf{w} \in Th(W) \Rightarrow \forall j.\ \mathbf{w} \in H_j$$

The proof is an obvious consequence of the fact that, for all j,

$$h_0^j = Th(W) \quad \text{and} \quad h_0^j \subseteq H_j.$$

The next lemma has a bit more substance.

Lemma 3.3

For $H_i$ and $\delta$ defined as in Lemma 3.2,

$$\text{if } \neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin (H_{j-1} \cup H_j) \text{ and } \mathbf{A} \in H_j \text{ then } \mathbf{w} \in H_j.$$

Proof

First, observe that:

$$\forall \theta.\ \theta \in H_j \Rightarrow \exists i.\ \theta \in h_i^j \qquad \text{(i)}$$

$$\forall \theta.\ \theta \notin H_j \Rightarrow \forall i.\ \theta \notin h_i^j \qquad \text{(ii)}$$

It follows that $\mathbf{A} \in H_j \Rightarrow \exists\ i.\ \mathbf{A} \in h_i^j$.

Let k be the least such i. Clearly, $\forall\ i \geq k$,

$$\mathbf{A} \in h_i^j \text{ and } \neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin (H_{j-1} \cup H_j) \Rightarrow \forall\ i \geq k.\ \delta \in D_i^j \subseteq D.$$

Let c be the cardinality of D. Since D is finite, c is finite. It can be shown that $\delta \in GD_{k+c}^j$, as follows:

Clearly, if $GD_i^j \supseteq GD_{i+1}^j$ for some i, $k \leq i < k+c$, then $\delta \in GD_i^j$.

$\forall i.\ k \leq i < k+c$:

$\delta \notin GD_{k+c}^j \Rightarrow \text{card}(GD_i^j)$ must be monotone increasing.

$\Rightarrow \text{card}(GD_{k+c}^j) \geq c$

but it must be that $GD_i^j \subseteq D$.

Therefore $GD_{k+c}^j = D \Rightarrow \delta \in GD_{k+c}^j$ (contradiction)

If $\delta \in GD_{k+c}^j$ then $\mathbf{w} \in h_{k+c}^j \subseteq H_j$.

Thus $\mathbf{w} \in H_j$.

QED

The proof of lemma 3.3 relies on the finiteness of D, explaining the restrictions placed on the alphabet and default theories at the beginning of this chapter. It is reasonable to ask whether such restrictions are necessary. The answer to this question appears to be "yes". Certainly, it can be shown that the lemma does not hold for infinite D. To see this, consider the theory with an infinite set of constants, $\{a_i\}$, an infinite set of defaults,

$$D = \{ \delta_i = \frac{A : P(a_i)}{P(a_i)} \} \cup \{ \delta = \frac{A : Q(a_1)}{Q(a_1)} \},$$

and $W = \{A\}$. It should be obvious that $A \in H_1$ and that $\neg Q(a_1) \notin (H_0 \cup H_1)$. It should also be apparent that the sequence of $h_i^1$'s defined by $GD_i^1 = GD_{i-1}^1 \cup \{\delta_i\}$, $(i > 0)$, satisfies Definition 3.1, but that $Q(a_1) \notin H_1$.

The related question, whether there is any iterative mechanism which yields the desired results will be addressed in Chapter 5.

The term "Hypothesized Extensions" for the $H_i$'s will now be justified by the first half of a completeness result. The sequence of $H_i$'s is said to <u>converge</u> if $\exists$ n. $H_n = H_{n+1}$. It is shown that if the sequence of $H_i$'s ever converges, $H_n$ is an extension. Furthermore, if any of the $H_i$'s is an extension, the sequence converges at that point.

<u>Theorem 3.4</u>

A sequence $H_0$, $H_1$, ... defined as in Definition 3.1 contains two consecutive, identical elements, $H_{n-1}$ and $H_n$, <u>iff</u> $H_{n-1}$ is an extension for $\Delta$.

<u>Proof</u>

(=>) Following Reiter (1980, thm 2.1), define

$E_0 = W$ and, for $i \geq 0$,

$$E_{i+1} = Th(E_i) \cup \{\mathbf{w} | \frac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{w}} \in D,$$

$$\mathbf{A} \in E_i,\ \neg\mathbf{B}_1,\ldots,\neg\mathbf{B}_m \notin H_n\}$$

then $H_n$ (= $H_{n-1}$) is an extension for $\Delta$ <u>iff</u> $H_n = \bigcup_{i=0}^{\infty} E_i$.

Thus it must be proved that

(a) $\bigcup_{i=0}^{\infty} E_i \subseteq H_n$ and

(b) $H_n \subseteq \bigcup_{i=0}^{\infty} E_i$

a) This can be shown by induction.

Clearly $E_0 \subseteq H_n$.

Assume $E_i \subseteq H_n$ and consider $\theta \in E_{i+1}$.

There are two cases for $\theta$:

i) $\theta \in Th(E_i)$

=> $\theta \in H_n$ since $E_i \subseteq H_n$ and the $H_i$ are closed under $Th(\ )$.

ii) Otherwise $\theta$ is $\mathbf{w}$ where:

$$\frac{\mathbf{A} : \mathbf{B}_1,\ldots,\mathbf{B}_m}{\mathbf{w}} \in D,\ \mathbf{A} \in E_i,\ \neg\mathbf{B}_1,\ldots,\neg\mathbf{B}_m \notin H_n.$$

But $\mathbf{A} \in E_i$ => $\mathbf{A} \in H_n$. Thus, by lemma 3.3, $\mathbf{w} \in H_n$.

It follows that $\bigcup_{i=0}^{\infty} E_i \subseteq H_n$.

b) By definition, $H_n = \bigcup_{i=0}^{\infty} h_k^n$. It suffices, therefore, to show that $\forall$ k. $h_k^n \subseteq \bigcup_{i=0}^{\infty} E_i$. This is demonstrated by induction on k. The base step is provided by:

$$h_0^n = Th(W) = Th(E_0) \subseteq E_1 \subseteq \bigcup_{i=0}^{\infty} E_i.$$

Hence, assume $h^n_{k-1} \subseteq \bigcup_{i=0}^{\infty} E_i$, for some $k > 0$. Consider

$h^n_k = Th(h^n_{k-1} \cup \{\mathbf{w}\})$.

Clearly $\mathbf{w} \in \{\mathbf{w} \mid \dfrac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{w}} \in D,$

$\mathbf{A} \in h^n_{k-1}, \neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin (h^n_{k-1} \cup H_{n-1})\}$

(Note that $(h^n_{k-1} \cup H_{n-1}) = H_n$.)

Since $\bigcup_{i=0}^{\infty} E_i$ is closed under Th( ), and $h^n_{k-1} \subseteq \bigcup_{i=0}^{\infty} E_i$ by hypothesis, it suffices to show that $\mathbf{w} \in \bigcup_{i=0}^{\infty} E_i$.

$\mathbf{A} \in h^n_{k-1} \Rightarrow \mathbf{A} \in \bigcup_{i=0}^{\infty} E_i$. (hypothesis)

Thus $\dfrac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{w}} \in D$, $\mathbf{A} \in \bigcup_{i=0}^{\infty} E_i$, $\neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin H_n$.

Therefore $\mathbf{w} \in \bigcup_{i=0}^{\infty} E_i$. From this, the result follows.

Combining (a) and (b) yields:

$H_{n-1} = H_n = \bigcup_{i=0}^{\infty} E_i$.

$\Rightarrow H_{n-1}$ is an extension for $\Delta$.

(<=) It will now be shown that if $H_{n-1}$ is an extension for $\Delta$ then $H_{n-1} = H_n$ by showing that:

(a) $H_n \subseteq H_{n-1}$, and

(b) $H_{n-1} \subseteq H_n$.

By definition, $H_n = \bigcup_{k=0}^{\infty} h_k^n$. Induction yields the desired result, as follows:

a) Consider $\theta \in H_n$. There are two cases:

i) $\theta \in h_0^n \Rightarrow \theta \in Th(W) = h_0^{n-1} \subseteq H_{n-1} \Rightarrow \theta \in H_{n-1}$.

ii) otherwise, assume $h_{k-1}^n \subseteq H_{n-1}$.

Thus $\theta \in h_k^n \Rightarrow \theta \in Th(h_{k-1}^n \cup \{\mathbf{w}\})$,

for some $\delta = \dfrac{\mathbf{A} : \mathbf{B}_1,\ldots,\mathbf{B}_m}{\mathbf{w}} \in D_k^n$.

$\Rightarrow \mathbf{A} \in h_{k-1}^n$

$\Rightarrow \mathbf{A} \in H_{n-1}$ (hypothesis)

also, $\neg\mathbf{B}_1,\ldots,\neg\mathbf{B}_m \notin (h_{k-1}^n \cup H_{n-1})$.

$\Rightarrow \neg\mathbf{B}_1,\ldots,\neg\mathbf{B}_m \notin H_{n-1}$.

Therefore, $\mathbf{w} \in H_{n-1}$ (lemma 3.3)

$\Rightarrow \theta \in H_{n-1}$ ($H_i$ closed under Th( ))

b) As noted above, $H_{n-1}$ is an extension $\Rightarrow H_{n-1} = \bigcup_{i=0}^{\infty} E_i$.

Consider $\theta \in H_{n-1}$ ($\Rightarrow \theta \in \bigcup_{i=0}^{\infty} E_i \Rightarrow \exists\, i.\ \theta \in E_i$).

Continuing by induction, it is shown that $\theta \in H_n$ as follows:

i) $\theta \in E_0 \Rightarrow \theta \in W \Rightarrow \theta \in h_0^n \Rightarrow \theta \in H_n$

Therefore $E_0 \subseteq H_n$.

ii) Assume $E_{i-1} \subseteq H_n$ for some $i > 0$.

$$\text{Then } \theta \in E_i \Rightarrow \theta \in Th(E_{i-1}) \cup \{\mathbf{w} \mid \frac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{w}} \in D,$$

$$\mathbf{A} \in E_{i-1}, \neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin H_{n-1}\}$$

Cases:

1) $\theta \in Th(E_{i-1}) \Rightarrow \theta \in H_n$ (hypothesis and closure)

2) otherwise, $\theta \in \{\mathbf{w} \mid \frac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{w}} \in D,$

$$\mathbf{A} \in E_{i-1}, \neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin H_{n-1}\}$$

$\mathbf{A} \in E_{i-1} \Rightarrow \mathbf{A} \in H_n$ (hypothesis)

It was proved in (a) above that $H_n \subseteq H_{n-1}$.

Thus $\neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin H_{n-1} \Rightarrow \neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin H_n$.

Therefore, by lemma 3.3, $\theta \in H_n$.

Combining (a) and (b) gives the desired result, namely:

$H_{n-1} = H_n$.

QED

The following corollary underlines the stability of the method.

Corollary 3.5

If $\Delta = (D,W)$ and $H_i$ are defined as in definition 3.1 then: for arbitrary $j > 0$

$$H_{j-1} = H_j \Rightarrow H_j = H_{j+1}.$$

Proof

$H_{j-1} = H_j \Rightarrow H_j$ is an extension. By Theorem 3.4, $H_j = H_{j+1}$.

An example may illustrate the algorithm. Consider the default theory, $\Delta = (D,W)$, with:

$$D = \{ \delta_1 = \frac{A : B}{B} , \delta_2 = \frac{A : \neg B}{\neg B} \}$$

$$W = \{ A \}$$

This theory has two extensions:

$$E_1 = Th(\{A, B\})$$

$$E_2 = Th(\{A,\neg B\})$$

The first iteration of the algorithm results in:

$$H_0 = Th(\{A\})$$

$h_0^1 = H_0$ $\quad$ $GD_0^1 = \emptyset$ $\quad$ $D_0^1 = \{\delta_1,\delta_2\}$

At this point there are two ways to proceed, depending on whether $\delta_1$ or $\delta_2$ is chosen in forming $h_1^1$. Each case will be considered in turn.

1) If $\delta_1$ were chosen:

$h_1^1 = Th(\{A,B\})$ $\quad$ $GD_1^1 = \{\delta_1\}$ $\quad$ $D_1^1 = \{\delta_1\}$

All succeeding $h_i^1$ will be identical to $h_1^1$, since there are no more eligible defaults. Hence:

$$H_1 = \bigcup_{i=0}^{\infty} h_i^1 = Th(\{A,B\}) = E_1.$$

$h_0^2 = Th(\{A\})$ $\quad$ $GD_0^2 = \emptyset$ $\quad$ $D_0^2 = \{\delta_1\}$

$h_1^2 = Th(\{A,B\})$ $\quad$ $GD_1^2 = \{\delta_1\}$ $\quad$ $D_1^2 = \{\delta_1\}$

$$H_2 = Th(\{A,B\}) = E_1.$$

Of course, this was to be expected: since $E_1$ is an extension for $\Delta$ and $H_1 = E_1$, Theorem 3.4 guarantees that $H_2 = H_1$. It can easily be seen that the presence of B in $H_1$ prevents the application of $\delta_2$ in the generation of $H_2$ and so produces the desired result.

2) If $\delta_2$ were chosen:

$h_1^1 = Th(\{A,\neg B\}) \qquad GD_1^1 = \{\delta_2\} \qquad D_1^1 = \{\delta_2\}$

Again, there are no more eligible defaults, so:

$H_1 = Th(\{A,\neg B\}) = E_2.$

$h_0^2 = Th(\{A\}) \qquad GD_0^2 = \emptyset \qquad D_0^2 = \{\delta_2\}$

$h_1^2 = Th(\{A,\neg B\}) \qquad GD_1^2 = \{\delta_2\} \qquad D_1^2 = \{\delta_2\}$

$H_2 = Th(\{A,\neg B\}) = E_2.$

Thus the method is capable of converging on each of the extensions for $\Delta$.

Comforting as the results presented may be, they do nothing to indicate that the method could ever arrive at an $H_n$ which is, indeed, an extension. The following theorem partially rectifies this, proving that for an arbitrary extension, E, there is a sequence of $H_i$'s which converges to E. In fact, there is a sequence which converges in a single iteration.

## Theorem 3.6

Let $\Delta = (D,W)$ be a default theory with extension E. There is a sequence $H_0, H_1, \ldots, H_{n-1}, H_n$, defined as per definition 3.1, such that $H_{n-1} = H_n = E$. In particular, there is a sequence such that $H_1 = H_2 = E$.

Proof

If W is inconsistent, then E = L. (Reiter 1980, Cor 2.2)

$H_0 = Th(W) = L$

For arbitrary $i > 0$,

$h_0^i = Th(W) = L$

Clearly all $h_j^i \subseteq L$

$$\Rightarrow H_i = \bigcup_{j=0}^{\infty} h_j^i = L = E.$$

Hence assume that W is consistent.

The set of generating defaults for E with respect to $\Delta$ is defined as follows (Reiter 1980):

$$GD(E,\Delta) = \{ \frac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{W}} \in D \mid \mathbf{A} \in E, \neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin E \}$$

Lemma 3.7 (below) shows that a non-trivial subset of the possible sequences of $H_i$'s defined by definition 3.1 can be obtained by limiting the choice of $\delta$ at $\dagger$ in the definition to an arbitrary member of $((D_i^j - GD_i^j) \cap GD(E,\Delta))$. This device will be employed to complete the proof.

It suffices to show that $H_1 = E$, since $H_1 = H_2$ then follows directly from Theorem 3.4.

Clearly, $W \subseteq H_1 \subseteq Th(W \cup CONSEQUENTS(GD(E,\Delta)))$

$\Rightarrow H_1 \subseteq E$. (Reiter 1980, Thm 2.5)

It remains to show that $E \subseteq H_1$.

Define $E_i$ as before, and recall that, since E is an extension for $\Delta$, $E = \bigcup_{i=0}^{\infty} E_i$.

Clearly $E_0 \subseteq H_1$.

For $i \geq 0$, assume $E_i \subseteq H_1$, and consider $\mathbf{w} \in E_{i+1}$.

There are two cases:

i) $\mathbf{w} \in Th(E_{i+1}) \Rightarrow \mathbf{w} \in H_1$ (hypothesis and closure).

ii) Otherwise, there is a default,

$$\delta = \frac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{w}} \in D,\ \mathbf{A} \in E_i,\ \neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin E.$$

Clearly, $\mathbf{A} \in E$ and $\mathbf{A} \in H_1$ ($E_i \subseteq E$, hypothesis)

Therefore, $\delta \in GD(E, \Delta)$.

$\mathbf{A} \in H_1,\ \neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin (H_0 \cup H_1) \Rightarrow \mathbf{w} \in H_1$. (Lemma 3.3)

It remains, therefore, to show that $\neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin (H_0 \cup H_1)$.

If this is not true, then there are two cases:

i) $\exists i.\ \neg\mathbf{B}_i \in H_0 = Th(W) \subseteq E$

$\Rightarrow \neg\mathbf{B}_i \in E$ (contradiction)

ii) $\exists i.\ \neg\mathbf{B}_i \in H_1$

It was shown above that $H_1 \subseteq E$.

$\Rightarrow \neg\mathbf{B}_i \in E$ (contradiction)

Therefore, $\neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin (H_0 \cup H_1)$.

Thus $E_{i+1} \subseteq H_1$. It follows by induction that $E \subseteq H_1$.

QED

The above result hinges on the fact that it is always possible to restrict the eligible choices of defaults to those which, in some sense, "must" be made to yield the extension in question. The following lemma shows that this restriction is always possible – provided all previous choices have been so restricted – without eliminating all of the possibilities.

Lemma 3.7

Given a default theory, $\Delta = (D,W)$, with an extension, E, if $GD_i^1 \subseteq GD(E,\Delta)$ then

$$((D_i^1 - GD_i^1) \cap GD(E,\Delta)) = \emptyset \text{ iff } (D_i^1 - GD_i^1) = \emptyset$$

Proof

(<=) Trivial

(=>) The proof proceeds by induction on i.

Note: $\forall$ i. $(h_i^1 \cup H_0) = h_i^1$ $\qquad (H_0 = Th(W) \subseteq h_i^1)$

As a notational convenience, the eligible defaults, $ED_i^j$, are defined as $(D_i^j - GD_i^j)$: the applicable defaults less those which have already been applied.

i = 0

The inductive proof for the i = 0 case hinges on the following observations:

i) $D_0^1 = \{\delta \mid \delta = \dfrac{\mathbf{A} : \mathbf{B}_1,\ldots,\mathbf{B}_m}{\mathbf{w}} \in D,\ \mathbf{A} \in h_0^1,\ \neg\mathbf{B}_1,\ldots,\neg\mathbf{B}_m \notin h_0^1\}$

ii) $\mathbf{A} \in h_0^1 = Th(W) \Rightarrow \mathbf{A} \in E$

iii) $\neg\mathbf{B}_i \notin E \Rightarrow \neg\mathbf{B}_i \notin h_0^1$

iv) $ED_0^1 = (D_0^1 - GD_0^1) = (D_0^1 - \emptyset) = D_0^1$.

Assume $(GD(E,\Delta) \cap ED_0^1) = \emptyset$ and $ED_0^1 \neq \emptyset$.

$$\Rightarrow \exists\ \delta = \frac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{w}} \in D,\ \mathbf{A} \in Th(W),$$

$$\neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin h_0^1 = Th(W),$$

and, for some i, $\neg\mathbf{B}_i \in E$.

Recall that, since E is an extension for $\Delta$,

$$E = \bigcup_{j=0}^{\infty} E_j. \qquad \text{(Reiter 1980, Thm 2.1)}$$

By induction on j, it will be shown that $\neg\mathbf{B}_i \notin E$.

Clearly $E_0 = W \subseteq Th(W) \Rightarrow \neg\mathbf{B}_i \notin E_0$.

Let $E_j \subseteq Th(W)$ and consider $E_{j+1}$.

$$\text{Let } D_j = \{\delta \mid \delta = \frac{\mathbf{C} : \mathbf{F}_1, \ldots, \mathbf{F}_r}{\mathbf{v}} \in D,\ \mathbf{C} \in E_j,$$

$$\neg\mathbf{F}_1, \ldots, \neg\mathbf{F}_r \notin E\}$$

$E_{j+1} = Th(E_j) \cup CONSEQUENTS(D_j)$

$Th(E_j) \subseteq Th(W)$ (hypothesis)

$\neg\mathbf{B}_i \in E_{j+1} \wedge \neg\mathbf{B}_i \notin Th(W) \Rightarrow \neg\mathbf{B}_i \in CONSEQUENTS(D_j)$.

But $D_j \subseteq GD(E,\Delta)$ and $D_j \subseteq D_0^1 = ED_0^1$

$\Rightarrow D_j \subseteq (GD(E,\Delta) \cap ED_0^1)$

$\Rightarrow D_j = \emptyset$

$\Rightarrow CONSEQUENTS(D_j) = \emptyset$

$\Rightarrow E_{j+1} = Th(E_j) \subseteq Th(W)$

and $\neg\mathbf{B}_i \notin CONSEQUENTS(D_j)$ (contradiction)

Thus $\neg\mathbf{B}_i \notin \bigcup_{j=0}^{\infty} E_j \Rightarrow \neg\mathbf{B}_i \notin E$ (contradiction)

Therefore $(GD(E,\triangle) \cap ED_0^1) = \emptyset \Rightarrow ED_0^1 = \emptyset$.

<u>i > 0</u>

For the inductive step, allow the conjecture to hold for $ED_{i-1}^1$ and consider $ED_i^1$ $(i \geq 1)$.

Assume $(GD(E,\triangle) \cap ED_i^1) = \emptyset$ and $ED_i^1 \neq \emptyset$.

The discussion below makes use of the following identities:

$$\begin{aligned}(GD(E,\triangle) \cap ED_i^1) &= (GD(E,\triangle) \cap (D_i^1 - GD_i^1)) \\ &= ((GD(E,\triangle) - GD_i^j) \cap D_i^1) \\ &= \emptyset \qquad \text{(hypothesis)}\end{aligned}$$

$ED_i^1 \neq \emptyset \Rightarrow \exists\, \delta = \dfrac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{W}} \in D$ such that $\delta \in ED_i^1$.

$\Rightarrow \delta \in D_i^1 \Rightarrow \mathbf{A} \in h_i^1 \wedge \neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m \notin h_i^1$.

$\delta \in ED_i^1 \wedge \delta \notin GD(E,\triangle) \Rightarrow \mathbf{A} \notin E \vee \exists\, i.\ \neg\mathbf{B}_i \in E$.

Clearly, $h_i^1 \subseteq E$ $(GD_i^1 \subseteq GD(E,\triangle))$

$\Rightarrow \mathbf{A} \in E$.

Therefore, it must be that $\neg\mathbf{B}_i \in E$.

If $(GD(E,\triangle) - GD_i^1) = \emptyset$ then

If $\delta \in GD(E,\triangle)$ then $\delta \in GD_i^1$

$\Rightarrow$ CONSEQUENTS$(GD(E,\triangle)) \subseteq h_i^1$

$\Rightarrow$ (W U CONSEQUENTS$(GD(E,\triangle))) \subseteq h_i^1$

$\Rightarrow E \subseteq h_i^1$ (closure, Reiter 1980, Thm 2.5)

$\Rightarrow \neg\mathbf{B}_i \notin E$ (contradiction)

Otherwise $\delta \in D_i^1 \Rightarrow \delta \notin (GD(E,\triangle) - GD_i^1)$

$$\Rightarrow \delta \notin GD(E,\Delta) \vee \delta \in GD_i^1.$$

Thus, if $\delta \in D_i^1$, there are two cases:

i) $\delta \in GD_i^1 \subseteq GD(E,\Delta) \Rightarrow \neg\mathbf{B}_i \notin E$ (contradiction)

ii) $\delta \notin GD(E,\Delta)$

$$\Rightarrow (D_i^1 \cap GD(E,\Delta)) = \emptyset \qquad \text{(given (i))}$$

$$\Rightarrow \{\delta \mid \delta = \frac{\mathbf{C} : \mathbf{F}_1,\ldots,\mathbf{F}_r}{\mathbf{u}} \in D,\ \mathbf{C} \in h_i^1,\ \neg\mathbf{F}_1,\ldots,\neg\mathbf{F}_r \notin h_i^1\} \cap \{\delta \mid \delta = \frac{\mathbf{C} : \mathbf{F}_1,\ldots,\mathbf{F}_r}{\mathbf{u}} \in D,\ \mathbf{C} \in E,\ \neg\mathbf{F}_1,\ldots,\neg\mathbf{F}_r \notin E\} = \emptyset$$

Observe that $\mathbf{C} \in h_i^1 \Rightarrow \mathbf{C} \in E$.

Assume that $D_i^1 \neq \emptyset$

$$\Rightarrow ((\delta = \frac{\mathbf{C} : \mathbf{F}_1,\ldots,\mathbf{F}_r}{\mathbf{u}} \in D \wedge \mathbf{C} \in h_i^1 \wedge \neg\mathbf{F}_1,\ldots,\neg\mathbf{F}_r \notin h_i^1) \Rightarrow \exists k.\ \neg\mathbf{F}_k \in E\ )$$

Consider $\delta' = \dfrac{\mathbf{C} : \mathbf{F}_1,\ldots,\mathbf{F}_r}{\mathbf{u}} \in (ED_{i-1}^1 \cap GD(E,\Delta))$

If there is no such $\delta'$ then, by hypothesis,

$$ED_{i-1}^1 = \emptyset \Rightarrow h_i^1 = h_{i-1}^1$$
$$\Rightarrow ED_i^1 = \emptyset$$

Otherwise:

$$\mathbf{C} \in h_{i-1}^1 \subseteq h_i^1,\ \neg\mathbf{F}_1,\ldots,\neg\mathbf{F}_r \notin E$$

But $h_i^1 \subseteq E \Rightarrow \neg\mathbf{F}_1,\ldots,\neg\mathbf{F}_r \notin h_i^1$.

Therefore $\delta' \in D_i^1 \wedge \neg\mathbf{F}_1,\ldots,\neg\mathbf{F}_r \notin E$

$$\Rightarrow \delta' \in D \wedge \mathbf{C} \in h_i^1 \wedge \neg\mathbf{F}_1, \ldots, \neg\mathbf{F}_r \notin E$$

$$\Rightarrow \delta' \in (D_i^1 \cap GD(E,\Delta)) \qquad \text{(contradiction)}$$

Therefore, $D_i^1 = \emptyset \Rightarrow ED_i^1 = \emptyset$

QED

Given the optimistic nature of these results, one might hope to be able to strengthen them by showing that all sequences of $H_i$'s converge on extensions. Unfortunately, this is not true in general, since it would imply that all default theories have extensions, which is not the case. (For example, the theory:

$$W = \{ \}$$

$$D = \left\{ \frac{:A \wedge \neg B}{A}, \frac{:B \wedge \neg C}{B}, \frac{:C \wedge \neg A}{C} \right\}$$

has no extension.) The weaker result, that all sequences of $H_i$'s converge provided that $\Delta$ has an extension, also fails to hold for general default theories. The default theory:

$$W = \{ A \}$$

$$D = \left\{ \delta_0 = \frac{A : B \wedge \neg D}{B}, \delta_1 = \frac{A : D \wedge \neg C}{D}, \delta_2 = \frac{B : C}{C} \right\}$$

has an extension, $E = Th(\{A,B,C\})$. The sequence of $H_j$'s determined by choosing:

$$GD_i^j = GD_{i-1}^j \cup \{\delta_k\} \quad j > 0, \ k = (i-1) \bmod 3$$

is legal according to Definition 3.1 but never converges.

In general, then, the most that could be hoped for is a tractible means of determining whether a particular sequence of $H_i$'s will not converge. Given the finite nature of D, the only way for a sequence of $H_i$'s not to converge is by the occurrence, somewhere in the sequence, of an $H_n = H_m$, $0 \le m < n-1$ (ie for the sequence to cycle). While theoretically possible, it does not seem desirable to maintain a complete list of the $H_i$'s and compare each new element with all of its predecessors! Experience has shown that the occurrence, at any step in the construction of $H_i$, of a condition where $GD_i^j \not\subseteq D_i^j$ is a reliable indicator that the sequence is about to cycle. Intuitively, this condition indicates that the justifications for a default, $\delta$, which has already been applied, have been denied. Thus, the consequent of $\delta$ will not be in $H_{i+1}$. This information, together with the fact that it is known that there is a sequence of $H_i$'s which converges immediately (ie without looping), might be applied by a depth-first search for an extension to indicate the need for backtracking. It has not been determined, however, whether this rule always holds or would serve only as an heuristic.

The question for which, if any, classes of default theories every sequence of $H_i$'s is guaranteed to converge on an extension (if there is one) for $\Delta$ remains open. Chapter 4 examines this problem further, suggesting some promising candidates.

All of the results presented above apply equally well to closed and open default theories. To see this, it is sufficient to observe that the Herbrand Universe for a finite default theory, $\Delta$, over a finite alphabet with no function symbols, A, must be finite. From this it follows that CLOSED-DEFAULTS($\Delta$) must also be finite. Thus, all of Reiter's results generalizing the notion of extension to arbitrary default theories can also be applied here, finiteness notwithstanding. The reader is referred to Reiter's paper for a detailed description of the machinery involved.

# Chapter 4

## Directions for Future Research

During the course of the research underlying this thesis, a number of promising avenues for research on related topics became apparent. These ranged from extensions of the results presented here to determining the relationships between default logic and the many other existing formalisms. Some of the topics which warrant further exploration are outlined below. Included in the discussion are possible approaches which _may_ prove fruitful in dealing with the problems.

### 4.1 Default Type Hierarchies

Hierarchal Taxonomies, or type hierarchies, are extremely common in AI. The fact that normal IS-A hierarchies are isomorphic to some underlying first-order logic is now widely accepted (Woods 1975, Schubert 1976, Hayes 1977). When exceptions are allowed within the inheritance structure, however, their semantics are no longer so straightforward.

It appears that semi-normal default theories can be used to represent type hierarchies with exceptions. Since the hierarchies are acyclic, the underlying default theories will also be acyclic – although exactly what constitutes an acyclic default theory is not necessarily obvious.

During the course of the research reported herein, several attempts were made to define just this. While a general, formal definition of acyclic default theories has yet to be obtained, a characterization which seems to capture the essential features has been found. As one might expect, given the well-behavedness of normal default theories, the central feature which must be dealt with is in those parts of the justifications which make individual defaults non-normal.

For the restricted class of semi-normal default theories which correspond to type hierarchies with exceptions, all defaults must have the form:

$$\delta = \frac{\mathbf{A} : \mathbf{B} \wedge \mathbf{C}_1 \wedge \ldots \wedge \mathbf{C}_m}{\mathbf{B}}$$

and all formulae in W must be of the form:

"$\mathbf{A} \supset \mathbf{B}$" or "$\mathbf{A}$"

where $\mathbf{A}$, $\mathbf{B}$, and the $\mathbf{C}_i$s are positive or negative literals. The relations $<$ and $\leq$ are defined as follows:

$$\begin{array}{ll} \delta \in D & \Rightarrow \mathbf{A} \leq \mathbf{B}, \quad \neg\mathbf{C}_i < \mathbf{B} \qquad (1 \leq i \leq m) \\ (\mathbf{A} \supset \mathbf{B}) & \Rightarrow \mathbf{A} \leq \mathbf{B} \end{array}$$

together with the usual transitivity relations. If these relations form a partial order on the alphabet, A, (<u>ie</u> $(\forall \mathbf{A} \in A)\ \neg(\mathbf{A} < \mathbf{A})$), the theory is called <u>acyclic</u>. Normal theories, which always have extensions, are all acyclic. Those theories without extensions which have so far been presented (<u>cf</u> [Reiter 1980, 1981]) are cyclic. This suggests that acyclic default theories may always have at least one extension. If this proves to be the case, it should be possible to precisely specify the semantics of any semantic network inference system in terms of a semi-normal default theory. Thus one could determine just what inferences the system was justified in making. Similarly, providing the acyclicity conditions were met, the system could be assured that there was always at least one extension for the theory it was working with. This result would be comforting to those using such a system.

Observation of the conditions which cause the procedure to cycle leads to the conjecture that, for the class of theories described above, the procedure presented in Chapter 3 always converges. This would provide a deterministic procedure for determining the extensions of such theories.

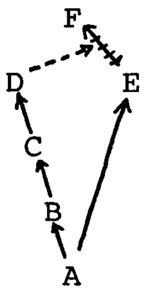
## 4.2 Parallel Inheritance Machines

The work described in this thesis originally derived from an attempt to use default hierarchies to specify the semantics of inheritance and cancellation thereof in NETL, a parallel machine architecture for representing semantic networks. There had been several efforts made [Fahlman et al 1981, Touretzky 1981] to provide a mechanism for determining inheritance using a parallel marker passing algorithm with cost proportional to the "height" of the network. These efforts were plagued by the discovery of peculiar cases which invalidated each newly proposed marker passing algorithm. It was hoped that, by expressing the ad hoc rules of NETL within the formal framework of default logic, the underlying principles might be illuminated, thus paving the way for developing a provably correct inference mechanism.

This work immediately brought encouraging fruit. It was shown that NETL inheritance structures could be expressed as monadic, semi-normal, default theories. The fact that unrestricted inheritance structures caused problems for their parallel marker-passing algorithms was noticed by Fahlman et al, and the so-called "naive" scheme was discarded. Examination of the corresponding default theories revealed the source of the problem: the unrestricted theories could have more than one (possibly orthogonal) extension. NETL's inheritance mechanism had been trying to derive a set of beliefs, in one parallel pass without lookahead, which would all belong to <u>one</u> of the possible extensions. Since extension membership is not a locally determined property, it was apparent that this would be impossible.

NETL's allowable inheritance structures were then restricted to prohibit nodes from which alternate pathways could lead to conflicting conclusions. The effect of these restrictions was to ensure that the corresponding default theory had exactly one extension. Attempts to develop a proof theory for the default logic, so restricted, which would capture the parallelism of NETL met with frustration. It turned out that any parallel proof theory which could correctly deal with cases like that shown in figure 4.1 could not deal with those like figure 4.2, even though when activated for a particular A (<u>eg</u> a), both have a unique extension (namely: $Th(\{A(a), B(a), C(a), D(a)\})$).

NETL Graph



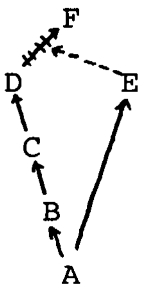
Default Theory

$$W = \{A(a)\}$$

$$D = \left\{ \frac{A(x) \ : \ B(x)}{B(x)}, \ \frac{B(x) \ : \ C(x)}{C(x)}, \ \frac{C(x) \ : \ D(x)}{D(x)}, \ \frac{A(x) \ : \ E(x)}{E(x)}, \ \frac{E(x) \ : \ \neg F(x) \wedge \neg D(x)}{\neg F(x)} \right\}$$

Figure 4.1 — D must be activated before E.

NETL Graph



Default Theory

$$W = \{A(a)\}$$

$$D = \left\{ \frac{A(x) \ : \ B(x)}{B(x)}, \ \frac{B(x) \ : \ C(x)}{C(x)}, \ \frac{C(x) \ : \ D(x)}{D(x)}, \ \frac{A(x) \ : \ E(x)}{E(x)}, \ \frac{E(x) \ : \ \neg F(x) \wedge \neg E(x)}{\neg F(x)} \right\}$$

Figure 4.2 — E must be activated before D.

This is because, in Figure 4.1, the "markers" must reach D by the time they reach E. In Figure 4.2, the opposite is true. Due to the locality of reference in NETL's parallel marker propagation, the required propagation pattern is impossible to determine in parallel. A new version of inheritance in NETL has been developed which disallows such constructs. Touretzky (1981) states that the added restrictions result in a provably correct scheme. If this is so, it will be instructive to determine how the restrictions on well-formedness of NETL inheritance hierarchies will affect the expressive power of the underlying default logic (and hence of NETL). This investigation has not yet been undertaken.

## 4.3 Truth Maintenance

Anyone familiar with the belief revision literature (see [Doyle and London 1980] for a survey) will notice certain similarities between the relaxation process carried out by such systems during "truth maintenance" and the iterations of the procedure presented here. The similarity is all the more compelling when the in- and out-hypotheses in the TMS justifications are viewed as the prerequisites and justifications, respectively, of a default theory.

It may be that a formal understanding of the truth maintenance process can be realized by explicitly detailing this correspondence. The goal of this research would be to show that such systems are essentially converging on extensions for some underlying default theory.

## 4.4 Theoretical Analysis

As has been mentioned, there has been a dearth of formal study of the relationships between the capabilities provided by the various approaches. Due to the level of interest in knowledge representation and reasoning, there has been a proliferation of ideas. Unfortunately, proponents of these ideas have not always provided the degree of rigour for which one might hope. New schemes are conceived without clear demonstrations that they are superior to (or even significantly different from) those already extant.

Rigourous comparisons of approaches would serve not only to demonstrate which are best suited to particular problems, but to point out weaknesses. This study might lead to the abandonment of some and the improvement of others to make them more universally applicable.

It seems clear that much of the work in the field is closely intertwined – perhaps not surprisingly, considering that everyone involved has the same set of problems with which to deal. If this intertwining can be unraveled, it should provide insight into other, more appropriate, ways of attacking the problems.

Approaching the problem from a different angle, there has been a rich tradition of study, dating back to Aristotle, of the properties of the various modalities (cf [Lemmon 1977]). The investigations of McDermott (1982) suggest that there may be ways to apply the classical modal logics to the problems addressed by non-monotonic and default logics. There are likely to be many contributions which modal logic can make to this domain of inquiry.

Default Logic avoids the introduction of modalities into the language. As noted in Chapter 2, this results in advantages and disadvantages. There is, as yet, no characterization of the costs and benefits of such an approach. For example, Default Logic's inability to combine defaults may be remediable without seriously distorting the fabric of the logic. The implications of such alterations invite further investigation.

# Chapter 5

# Conclusion

## 5.1 Evaluation of Results

The following discussion places this work in perspective. There are a variety of criteria which can be applied when determining the significance of these or any other theoretical results. The main dimensions considered will be the power and generality of the method and its applicability to common problems. The intention is provide an intuition about the strengths and shortcomings which have been encountered.

### 5.1.1 Finiteness Restrictions

As has been mentioned, the method presented is applicable only to default theories consisting of a finite set of defaults over a finite alphabet. To the logician, this may seem to be a disquieting shortcoming. In both database theory and common type/IS-A hierarchies, however, the domain of interest usually is (or is treated to be) finite. The case has been made in this thesis and elsewhere (*cf* Reiter 1978, 1978a, 1980, 1981) that default logic provides a suitable formalism for dealing with many of the common problems encountered in these areas. It can be seen that there is a large class of interesting problems to which the method may be applied.

### 5.1.2 Infinite Theories

It may still be of interest to consider whether there is a method for constructing extensions which is not subject to finiteness restrictions.[1] There is at least one: Reiter (1980) has shown that there is a sequence $E_0, E_1, \ldots$ such that $\bigcup_{i=0}^{\infty} E_i = E$ iff there is an extension, E. Furthermore, he has demonstrated that $E = Th(W \cup CONSEQUENTS(GD(E,\Delta)))$, where $GD(E,\Delta)$ are the generating defaults of E from $\Delta$ described in Chapter 3. From this, it is not difficult to show that the sequence of $h_i^j$ obtained as in Definition 3.1, except adding $(GD(E,\Delta) \cap (D_i^j - GD_i^j))$ at each step (at †), yields $\bigcup_{i=0}^{\infty} h_i^1 = E$ $(=H_1)$. The only drawback of this approach is that it is circular. E must be known to determine $GD(E,\Delta)$. This is not, then, a computationally feasible approach.

The existence of one method for arriving at the extensions of an infinite default theory may suggest that there are other, more feasible, approaches. Experience has shown that the immediately apparent or promising methods all fail to achieve the desired result. This appears to be because any non-circular method powerful enough to obtain the result corresponding to

---

[1] Given that extensions are not generally recursively enumerable, this question must be restricted to those theories with recursive, or at least recursively enumerable, extensions.

Lemma 3.3 for the infinite case cannot yield that corresponding to Theorem 3.6, and vice versa. Any procedure which guarantees including every applicable default seems unable to find all extensions. This seems to be a consequence of the interactions between defaults, and of non-monotonicity. Of course, an algorithm could consist of a breadth-first search for "Reiter's Sequence", but it is not clear what value breadth-first search has in a domain with a potentially infinite branching factor.

While this does not constitute a proof, hopefully the reader is left with an impression of the improbability of overcoming these limitations within the confines of any traditional, deterministic model of computation. These pessimistic indications and the wide applicability of the finite method provide little motivation for pursuing the matter further.

### 5.1.3 Computational Considerations

The question of the computational utility of the results is perhaps the hardest to address. Any formalism rooted, as this one is, in first-order logic is bound to be fraught with computationally expensive operations. As the method stands, it must compute the logical closure of each $h_i^j$, of which there are an infinite number. Of course, if the theory is finite, all beyond some finite number will be identical and need not be computed. Computing the closure is still a potentially expensive, or even non-terminating, operation.

It may be possible to heuristically limit this computation, deriving only "relevant" portions of the extension. For example, all tautologies are contained in every extension — but they rarely need to be computed. There may be interesting domains where closure is not necessary. IS-A hierarchies with exceptions immediately suggest themselves, given the ways in which these have traditionally been employed.

Another way to address this problem is to reformulate the procedure slightly. The computation of the logical closure in the construction of $H_0$ and each $h_i^j$ can be avoided if the definition of $D_i^j$ is replaced with:

$$D_i^j = \{\delta \mid \delta = \frac{\mathbf{A} : \mathbf{B}_1, \ldots, \mathbf{B}_m}{\mathbf{w}} \in D,\ h_i^j \vdash \mathbf{A},$$
$$(h_i^j \cup H_{j-1}) \nvdash \neg\mathbf{B}_1, \ldots, \neg\mathbf{B}_m\}$$

Since $\Delta$ must be finite, the references to provability ($\vdash$) and unprovability ($\nvdash$) are all decidable, and the procedure becomes an algorithm. The resulting $H_n$, should the algorithm converge, is no longer an extension, since extensions must be logically closed. $Th(H_n)$ will be an extension. For common applications, however, $H_n$ may be sufficient.

Even with this reformulation, consistency tests are expensive and indispensible (Reiter 1980). There is a great deal of work being done on ways to circumvent the problems presented both by the combinatorial explosion and the inherent undecidability of the underlying systems. These include theorem provers which constrain the explosion (cf C-Ordered Linear Resolution [Reiter 1971], SL Resolution [Kowalski and Kuehner 1971]), and various heuristic approaches such as resource limited computation (*cf* KRL [Bobrow and Winograd 1977, Winograd 1981]). It is beyond the scope of this thesis to enter into a discussion of such methods. It remains to be seen whether techniques will be found which can make default reasoning in general, and these results in particular, computationally attractive.

Finally, the fact that Theorem 3.6 shows that extensions can always be arrived at in a single iteration suggests that the procedure is too elaborate and could be simplified. Not every sequence converges immediately, however. The procedure outlined is general enough to recover from non-optimal choices of which defaults to apply. It was also designed to be robust enough to withstand the "discovery" of new consequences of a theory – the type of effect which might be expected were the required computations heuristically limited in some fashion.

## 5.2 Summary

A survey of formalisms for non-monotonic reasoning has been presented, providing a sketch of the "state of the art" in the field. Reiter's logic for default reasoning has been discussed in detail. A procedure for determining the extensions of general finite default theories has been demonstrated.

The potential impact of the procedure on some of the other research in the field has been explored, indicating some promising areas for future research. Grounds for cautious optimism about the tractibility of default theories capable of representing a wide variety of common situations have been provided. This optimism may not turn out to be well-founded, but it is hoped that this thesis may provide the groundwork for further development in this area.

## Bibliography


Bobrow, D.G. and Winograd, T. (1977), "An Overview of KRL-0, a Knowledge Representation Language", Cognitive Science 1(1).

Clark, K.L. (1978), "Negation as Failure", in Logic and Databases, H. Gallaire and J. Minker (eds.), Plenum Press, New York.

Codd, E.F. (1972), "Relational Completeness of Data Base Sublanguages", in R. Rustin (ed.), Data Base Systems, Prentice-Hall.

Davis, M. (1980), "The Mathematics of Non-Monotonic Reasoning", Artificial Intelligence 13, North-Holland.

Doyle, J. (1979), A Truth Maintenance System, AI Memo 521, MIT Artificial Intelligence Lab, Cambridge, Mass.

Doyle, J. and London, P. (1980), A Selected Descriptor-Indexed Bibliography to the Literature on Belief Revision, AI Memo 568, MIT, Cambridge, Mass.

Fahlman, S.E. (1979), NETL: A System for Representing and Using Real-World Knowledge, MIT Press, Cambridge, Mass.

Fahlman, S.E., Touretzky, D.S., and von Roggen, W. (1981), "Cancellation in a Parallel Semantic Network", Proc. Seventh International Joint Conference on Artificial Intelligence.

Grice, H.P. (1975), "Logic and Conversation", in Syntax and Semantics, Vol 3: Speech Acts, P. Cole and J.L. Morgan (eds.), Academic Press.

Hayes, P.J. (1973), "The Frame Problem and Related Problems in Artificial Intelligence", in Artificial and Human Thinking, A. Elithorn and D. Jones (eds.), Jossey-Bass Inc., San Francisco.

Hayes, P.J. (1977), "In Defense of Logic", Proc. Fifth International Joint Conference on Artificial Intelligence.

Hewitt, C. (1972), Description and Theoretical Analysis (Using Schemata) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot, AI Memo 251, MIT Project MAC, Cambridge, Mass.

Hughes, G.E. and Cresswel, M.J. (1968), An Introduction to Modal Logic, Methuen and Co. Ltd., London.

Kowalski, R.A. and Kuehner, D. (1971), "Linear Resolution with Selection Function", Artificial Intelligence 2, North-Holland.

Kramosil, I. (1975), "A Note on Deduction Rules with Negative Premises", Proc. Fourth International Joint Conference on Artificial Intelligence.

Lemmon, E.J. (1977), An Introduction to Modal Logic, with Dana Scott, K. Segerberg (ed.), American Philosophical Quarterly, Monograph No. 11, Oxford.

McAllester, D.A. (1978), A Three-Valued Truth Maintenance System, AI Memo 473, MIT, Cambridge, Mass.

McAllester, D.A. (1980), An Outlook On Truth Maintenance, AI Memo 551, MIT, Cambridge, Mass.

McCarthy, J. (1977), "Epistemological Problems of Artificial Intelligence", Proc. Fifth International Joint Conference on Artificial Intelligence.

McCarthy, J. (1980), "Circumscription – A Form of Non-Monotonic Reasoning", Artificial Intelligence 13, North-Holland.

McCarthy, J. and Hayes, P.J. (1969), "Some Philosophical Problems from the Standpoint of Artificial Intelligence", in Machine Intelligence 4, B. Meltzer and D. Michie (eds.), Edinburgh University Press, Edinburgh.

McDermott, D. and Doyle, J. (1980), "Non-Monotonic Logic I", Artificial Intelligence 13, North-Holland.

McDermott, D. (1982), "Non-Monotonic Logic II", J.ACM 29(1).

Mendelson, E. (1964), Introduction to Mathematical Logic, Van Nostrand Reinhold, New York.

Minsky, M. (1975), "A Framework for Representing Knowledge", in The Psychology of Computer Vision, P. Winston (ed.), McGraw-Hill, New York.

Raphael, B. (1971), "The Frame Problem in Problem-Solving", in Artificial Intelligence and Heuristic Programming, N.V. Findler and B. Meltzer (eds.), Edinburgh University Press, Edinburgh.

Reiter, R. (1978), "On Closed World Data Bases", in Logic and Databases, H. Gallaire and J. Minker (eds.), Plenum Press, New York.

Reiter, R. (1978a), "On Reasoning By Default", Proc. Second Symp. on Theoretical Issues in Natural Language Processing, Urbanna, Illinois.

Reiter, R. (1980), "A Logic For Default Reasoning", Artificial Intelligence (13), North-Holland.

Reiter, R. and Criscuolo, G. (1981), "On Interacting Defaults", Proc. Seventh International Joint Conference on Artificial Intelligence.

Reiter, R. (1981a), "Towards a Logical Reconstruction of Relational Database Theory", Manuscript.

Roberts, R.B. and Goldstein, I. (1977), The FRL Manual, AI Memo 409, MIT, Cambridge, Mass.

Robinson, J.A. (1965), "A Machine Oriented Logic Based on the Resolution Principle", J.ACM 12.

Roussel, P. (1975), PROLOG, Manuel de Reference et d'Utilisation, Group d'Intelligence Artificielle, U.E.R. de Marseille, France.

Sandewall, E. (1972), "An Approach to the Frame Problem and Its Implementation", in Machine Intelligence 7, B. Meltzer and D. Michie (eds.), Edinburgh University Press, Edinburgh.

Schubert, L.K. (1976), "Extending the Expressive Power of Semantic Networks", Artificial Intelligence 7(2), North-Holland.

Touretzky, D. (1981), personal communication.

Winograd, T. (1980), "Extended Inference Modes in Reasoning", Artificial Intelligence 13, North-Holland.

Woods, W.A. (1975), "What's in a Link?", Representation and Understanding, Academic Press.

## APPENDIX A

### Dictionary of Symbols

| Symbol | Interpretation |
|---|---|
| $\cap$ | Set intersection |
| $\cup$ | Set union |
| $-$ | Set difference |
| $\emptyset$ | The empty set |
| $\{ \}$ | The empty set |
| $\subset$ | Proper subset |
| $\not\subset$ | Not a proper subset |
| $\subseteq$ | Subset |
| $\not\subseteq$ | Not a subset |
| $\in$ | Is an element of |
| $\notin$ | Is not an element of |
| $\mid$ | Set qualification (read "such that") |
| iff | If and only if |
| $=>$ | It follows that |
| $\supset$ | Logical Implication |
| $\wedge$ | Logical And |
| $\vee$ | Logical Or |
| $\neg$ | Logical Not |
| $\exists$ | Existential quantifier |
| $\forall$ | Universal quantifier |

| | |
|---|---|
| $\cdot$ | Preceeding quantifiers bind to the end of the sentence or first enclosing right parenthesis |
| $\vdash$ | Provability relation |
| $\nvdash$ | Non-provability |