ON COMPUTING MINIMAL AND PERFECT MODEL MEMBERSHIP

C. A. JOHNSON
Computer Science Department
University of Keele
Staffordshire, ST5 5BG
England
email : chrisj@cs.keele.ac.uk

Abstract. The computational complexity of a number of problems relating to minimal models of non-Horn deductive databases is considered. In particular, the problem of determining minimal model membership is shown to be NP - complete for non-recursive propositional databases. The structure of minimal models is also examined using the notion of a cyclic tree, and methods of determining minimal model membership, minimality of models and compiling the GCWA are presented. The handling of negative premises is also considered using perfect model semantics, and methods for computing perfect model membership are presented.

Keywords: Deductive databases, indefinite data, generalised closed world assumption, complexity, minimal model, perfect model, cyclic tree.

Deductive databases allow the representation of data and data relationships using logical formulae of the form $P_1 \wedge P_2 \wedge \ldots \wedge P_h \rightarrow P_{h+1} \vee P_{h+2} \vee \ldots \vee P_{h+k}$. Given such a representation, methods for manipulating the database are then required, and one important aspect of such manipulation is the handling of negative data. Since such data is too voluminous to be represented in the database explicitly, it must be inferred via some metarule.

In the case of deductive databases that are restricted to *Horn* rules (in which $k = 1$), Reiter's closed world assumption (CWA) [20] (known as negation as failure in PROLOG [22]) allows the assumption of a negative (ground) atom if the corresponding positive atom cannot be deduced from the database. Alternatively the CWA can be rephrased in model theoretic terms: Horn databases are well known to have a unique minimal model, and the CWA thus reduces to truth evaluation with respect to this minimal model.

One of the disadvantages of Horn databases is their limited expressive power, hence the interest in non-Horn databases [1, 4, 5, 6, 15, 16]. In this case, there may be no unique minimal model, and it is well known that the closed world assumption mentioned above easily yields inconsistencies when applied to such databases [4, p. 177]. Several closed world metarules applicable to non-Horn databases have been suggested (see [1]), one of the main goals (and difficulties) concerning such metarules being to achieve computational feasibility.

The generalised closed world assumption (GCWA) introduced by Minker [16] is one such metarule. The GCWA extends Reiter's CWA by allowing the assumption of a negative (ground) atom given that the corresponding positive atom is not contained in any of the minimal models of the database. It thus relies upon an ability to compute minimal model membership. Query evaluation under the GCWA was studied by Grant and Minker [5] for indefinite relational (as opposed to deductive) databases, but even in this subcase their algorithms turn out to be exponential. Indeed the lack of an efficient method for computing minimal model membership has lead to the study of other (weaker) forms of the GCWA [13, 14, 21] involving a "Horn transformation" of the database.

In this paper we examine a number of problems related to minimal (and perfect) model membership, both from the theoretical standpoint of computational complexity (Sections 2 and 3), and with a view to developing practical methods for solving such problems (Sections 4 - 7).

For recursive databases (at the propositional level), the problem of determining if a given predicate lies in some minimal model is known ([3]) to be $\sum_2^P$ - complete. This suggests that the problem of determining minimal model membership for such databases is not in the class NP (i.e., $\sum_1^P$), and in particular that many well known theorem proving methods are incapable of handling the GCWA. This result is also of interest, since naturally occurring $\sum_k^P$ - complete problems are rare for $k > 1$. (The first order case is discussed in [2].)

Special cases in which the computational complexity is reduced are thus of interest, and in Section 2, we show that the problem of determining minimal model membership is NP - complete for non-recursive databases at the propositional level. We also introduce the witness property that is central to our study of minimal model structure in this and later sections.

A database is said to be *indefinite* iff there is some query having a disjunctive answer of size > 1, i.e., an indefinite answer. Indefiniteness again relates to minimal models, since a database is indefinite iff it has more than one minimal model. In Section 3 we show that the problem of determining whether a non-recursive database is indefinite is NP - complete. We also consider the problem of determining the size of minimal answers to queries. (Again, this problem is related to minimal model membership, since a (ground) atom lies in some minimal answer iff it lies in some minimal model [16].) Such information is useful when considering the termination of query processing methods. We show that this problem is NP - hard for non-recursive databases, and $\sum_2^P$ - complete in the recursive case. We also examine the possibility of using known information about T in order to reduce the complexity of the above-mentioned problems.

In Sections 4 - 6 we examine the structure of minimal models using the witness property and the concept of a cyclic tree. We also use such structural information in order to derive methods for determining minimal model membership, testing models for minimality, and compiling the GCWA (i.e., preprocessing the rules in the database in such a way that testing the GCWA then reduces to manipulation of the extensional part of the database). Compilation of the GCWA is also considered in [7].

Propositional databases are considered in Section 4 (for the non-recursive case) and Section 5 (for the recursive case). The first order level is considered in Section 6, where we place certain constraints on the database in order to guarantee the termination of our cyclic tree constructions.

Finally we turn our attention to databases containing rules of the form

$$P_1 \wedge P_2 \wedge \ldots \wedge P_h \wedge \neg Q_1 \wedge \neg Q_2 \wedge \ldots \wedge \neg Q_r \rightarrow P_{h+1} \vee P_{h+2} \vee \ldots \vee P_{h+k}$$

(in which negative premises are allowed). Such databases are interpreted using the perfect model semantics of [19]. For such databases, a method of computing perfect model membership is necessary for the processing of any query (either positive or negative) as a result of the negative subgoals within the rule bodies. In Section 7 we show that perfect model membership can be characterised in terms of cyclic trees. ([12] combines the methods of Section 7 with the methods of [10] to produce a query answering method for such databases.) It is also shown that for such databases, the handling of both positive and negative data is $\sum_2^P$ - complete. We also consider the special case of databases whose intension is Horn, and show that for such databases, the complexity is reduced to $\sum_1^P$ - completeness.

§1. PRELIMINARIES.

Throughout Sections 1 - 5, $\mathcal{L}$ will denote a propositional language $\mathcal{L} = \{P_1, P_2, \ldots, P_n\}$, where each P_i is a propositional predicate symbol. A *literal* is a predicate symbol P or its negation $\overline{P} = \neg P$. A *rule* C is a formula of the form

$$A_1 \wedge A_2 \wedge \ldots \wedge A_h \rightarrow B_1 \vee B_2 \vee \ldots \vee B_k$$

where each A_i and each B_j is a predicate symbol, $h \geq 0, k > 0$.

The predicates $\{A_1, A_2, \ldots, A_h\}$ are called the *antecedants* of C (antec(C)) and $\{B_1, B_2, \ldots, B_k\}$, the *consequents* of C (conseq(C)). Clearly we may assume that conseq(C) and antec(C) are disjoint. A deductive database is a set of rules, which throughout will be denoted by T. EXT(T) (the *extension* of T) denotes those rules in T whose antecedant is empty, and INT(T) (the *intension* of T) is $T-$ EXT(T).

1.1 Definition. Let C be the rule

$$A_1 \wedge A_2 \wedge \ldots \wedge A_h \rightarrow B_1 \vee B_2 \vee \ldots \vee B_k$$

then a *model* of C is a set $M \subseteq \mathcal{L}$ such that if $\{A_1, A_2, \ldots, A_h\} \subseteq M$, then $\{B_1, B_2, \ldots, B_k\}$ $\cap M \neq \phi$. If M models C, then we write $M \models C$.

M is a model of T iff $M \models C$ for each $C \in T$. A model M of T is said to be a *minimal model* iff no proper subset of M is a model of T.

1.2 Definition [16] (GCWA). Let P be a predicate symbol, then $\overline{P}$ may be assumed from T under the GCWA iff P is not contained in any minimal model of T.

Define MM(T, P) iff T has a minimal model containing P.

1.3 Definition (a). T is said to be *non-recursive* iff there is a *level function* $\ell : \mathcal{L} \rightarrow \mathbb{N}$ such that for each rule

$$A_1 \wedge A_2 \wedge \ldots \wedge A_h \rightarrow B_1 \vee B_2 \vee \ldots \vee B_k$$

in T, $\ell(A_i) < \ell(B_j)$ for each $i \leq h, j \leq k$.

(b). T is said to be *strongly stratified* iff there is a level function ℓ such that for each rule

$$A_1 \wedge A_2 \wedge \ldots \wedge A_h \rightarrow B_1 \vee B_2 \vee \ldots \vee B_k$$

in T, $\ell(A_i) < \ell(B_j) = \ell(B_l)$ for each $i \leq h$ and each $j, l \leq k$.

Thus the database $\{A \to B \vee C, C \to D\}$ is strongly stratified, whereas the database $\{A \to B \vee C, B \to D \vee E, D \to C\}$ is non-recursive, but not strongly stratified.

Stratification is more usually employed ([19]) as a method of handling negative literals within rule bodies. Specifically it insists that positive (negative) literals in the rule body have an ℓ - value $\leq$ ($<$) the ℓ - value of the predicates in the consequent, cf. Definition 7.1.2. Hence, our use of the term strongly stratified.

1.4 Definition. A *valuation* on $\mathcal{L}$ is a set of literals v such that for each predicate $P \in \mathcal{L}$, $|v \cap \{P, \overline{P}\}| = 1$. (Intuitively, P is true in v iff $P \in v$.) Alternatively we can think of v as defining a function $v : \mathcal{L} \to \{0, 1\}$ by $v(P) = 1$ iff $P \in v$.

If Φ is propositional in $\mathcal{L}$, then v *satisfies* Φ (and we write $v \models \Phi$) iff $\bigwedge\{P = v(P) \mid P \in \mathcal{L}\} \Longrightarrow \Phi$.

Φ is *consistent* iff Φ is satisfied by some valuation.

Quite clearly a model $M \subseteq \mathcal{L}$ can be converted into a valuation $v_M = M \cup \{\overline{S} \mid S \in \mathcal{L} - M\}$ (and vice versa) and for any rule C, $M \models C$ iff $v_M \models C$. The reason for the use of models (rather than valuations) in deductive database theory stems from the fact that only positive information is explicitly stored in the database.

1.5 Definition (The polynomial time hierarchy) [24, 25, 27].

A $\sum_k$ formula (in $\mathcal{L}$) is one of the form

$$\exists X_1 \, \forall X_2 \, \exists X_3 \ldots Q_k X_k \, \Phi$$

where each X_i is a string of predicates from $\mathcal{L}$, and Φ is propositional in the predicates in $X_1, X_2, \ldots, X_k$. $Q_k = \forall$ if k is even, else $Q_k = \exists$.

A problem is:

(a) in the class $\sum_k^P$ iff it is polynomially reducible to the problem of deciding upon the truth of $\sum_k$ formulae,

(b) $\sum_k^P$ - *hard* iff the converse reduction is possible, and

(c) $\sum_k^P$ - *complete* iff it is in $\sum_k^P$ and is $\sum_k^P$ - hard.

Similarly we define a $\prod_k$ formula as one of the form

$$\forall X_1 \, \exists X_2 \, \forall X_3 \ldots Q_k X_k \, \Phi$$

where each X_i is a string of predicates, and Φ is propositional in the predicates in $X_1, X_2, \ldots, X_k$. $Q_k = \exists$ if k is even, else $Q_k = \forall$.

Note that the class $\sum_1^P$ is more commonly referred to as NP. Clearly a $\sum_1$ formula represents the satisfiability problem.

It is easy to see that for each $k \geq 0$, $\sum_k^P \cup \prod_k^P \subseteq \sum_{k+1}^P \cap \prod_{k+1}^P$. Although the structure of the polynomial time hierarchy is still an open problem, it is the prevailing belief that this inclusion is proper and that neither $\sum_k^P$ nor $\prod_k^P$ is a subset of the other.

When proving hardness results, we will wish to convert a propositional formula into a deductive database. The following definition provides a mechanism for doing so.

1.6 Definition. Let Φ be a propositional formula in $\mathcal{L} = \{P_1, P_2, \ldots, P_n\}$. By eliminating $\rightarrow$ and $\leftrightarrow$ and using De Morgan's Laws, we may assume that Φ is built up from literals using $\wedge$ and $\vee$ only.

Let $\mathcal{L}' = \mathcal{L} \cup \{P_i' \mid i \leq n\}$ (where P_i' are new predicates not occurring in $\mathcal{L}$), and let Φ' be formed from Φ by replacing each occurrence of $\overline{P}_i$ by P_i'. Suppose that Φ' has subformulae $F_1, F_2, \ldots, F_k$ (where $F_k = \Phi'$). Define $\theta_1, \theta_2, \ldots, \theta_k$ by:

$$\theta_j \text{ is } \begin{cases} P \rightarrow A_j, & \text{if } F_j \text{ is } P; \\ A_i \wedge A_l \rightarrow A_j, & \text{if } F_j \text{ is } F_i \wedge F_l; \\ A_i \vee A_l \rightarrow A_j, & \text{if } F_j \text{ is } F_i \vee F_l. \end{cases}$$

where each A_j is a new predicate not occurring in $\mathcal{L}'$. Notice that the third case is not strictly a rule, but can clearly be converted into two rules $A_i \rightarrow A_j$ and $A_l \rightarrow A_j$.

Let $DD(\Phi) = \bigwedge_{i=1}^k \theta_i$. Notice that the predicate A_k represents Φ/Φ', and we shall write $A_k = \text{Pred}(\Phi)$.

Let Φ be as given in Definition 1.6. The following theorems are trivial using the fact that $DD(\Phi)$ is Horn.

1.7 Theorem. If $M \subseteq \mathcal{L}'$, then there is a unique $M^* \subseteq \{A_1, A_2, \ldots, A_k\}$ such that:

(a) $M \cup M^* \models DD(\Phi)$, and

(b) whenever $M \subseteq N' \subset M \cup M^*$, then $N' \not\models DD(\Phi)$.

It is clear that M^* represents the closure of M under $DD(\Phi)$, which we denote by $cl(\Phi, M)$, that is

$$cl(\Phi, M) = M^* = \{A_j \mid j \leq k, DD(\Phi) \models \bigwedge M \rightarrow A_j\}.$$

1.8 Theorem. If $M_0 \subseteq M_1 \subseteq \mathcal{L}'$, then $cl(\Phi, M_0) \subseteq cl(\Phi, M_1)$.

1.9 Theorem. Let $M' \subseteq \mathcal{L}' \cup \{A_i \mid i \leq k\}$ be such that $M' \models DD(\Phi)$, and $M \subseteq M' \cap \mathcal{L}'$. Then $cl(\Phi, M) \subseteq M'$.

The following theorem follows trivially by induction on the complexity of Φ.

1.10 Theorem. Let v be a valuation on $\mathcal{L}$ and $M = \{P_i \mid P_i \in v\} \cup \{P_i' \mid P_i \notin v\}$. Then $A_k = \mathrm{Pred}(\Phi) \in cl(\Phi, M)$ iff $v \models \Phi$.

§2. MINIMAL MODEL MEMBERSHIP AND COMPLEXITY.

As mentioned in the introduction, the problem of determining minimal model membership (i.e., MM, cf. Definition 1.2) for (propositional) recursive databases is known ([3]) to be $\sum_2^P$ - complete, and hence computationally very hard. Special cases in which the complexity is reduced are thus of interest, and in this section we show that the problem of determining minimal model membership is $\sum_1^P$ - complete for non-recursive databases. Our completeness result also introduces the witness property which is central to the development of weak deduction and cyclic trees in later sections.

2.1 Hardness.

We first show that determining MM is hard. That is, we need to show that for any $\sum_1$ formula Ψ there is a non-recursive deductive database $T(\Psi)$ (whose size is polynomial in that of Ψ) and a predicate X such that Ψ is true iff MM($T(\Psi)$, X).

2.1.1 Theorem ($\sum_1^P$ - hardness). Suppose $\Psi = \exists(P_1, P_2, \ldots, P_m)\,\Phi$, where Φ is a propositional formula in $\{P_1, P_2, \ldots, P_m\} \subseteq \mathcal{L}$. Let

$$T(\Psi) = \{DD(\Phi)\} \cup \{P_i \vee P_i' \mid i \leq m\}$$

then Ψ is true iff MM($T(\Psi)$, A_k) (where $A_k = \text{Pred}(\Phi)$, cf. Section 1.6). Notice that $T(\Psi)$ is non-recursive.

Proof ($\rightarrow$). Suppose $(B_1, B_2, \ldots, B_m) \in \{0,1\}^m$ is such that $\bigwedge\{P_i = B_i \mid i \leq m\} \Longrightarrow \Phi$. Let $M_0 = \{P_i \mid i \leq m, B_i = 1\} \cup \{P_i' \mid i \leq m, B_i = 0\}$, then by Theorems 1.7 and 1.10, $M_0 \cup cl(\Phi, M_0)$ is a minimal model of $T(\Psi)$ containing A_k.
($\leftarrow$). Let M be a minimal model of $T(\Psi)$ containing A_k, then for each $i \leq m, |M \cap \{P_i, P_i'\}| = 1$. For each $i \leq m$, define $B_i \in \{0,1\}$ by $B_i = 1$ iff $P_i \in M$. Let $M_0 = M \cap \mathcal{L}'$.

We claim that $\bigwedge\{P_i = B_i \mid i \leq m\} \Longrightarrow \Phi$. Suppose not, then $A_k \notin M_0 \cup cl(\Phi, M_0) \subseteq M - \{A_k\}$, and since $M_0 \cup cl(\Phi, M_0) \models T(\Psi)$ this contradicts the minimality of M. ∎

2.2 Completeness for non-recursive databases.

In order to prove completeness we need to consider the structure of minimal models. The witness property defined below, is the tool by which we do this, and this property will be used extensively again in later sections. Throughout this section we assume that T is non-recursive (in $\mathcal{L}$).

2.2.1 Definition. Let M be a model of T. A predicate $P \in M$ is *witnessed* in M by a rule

$$A_1 \wedge A_2 \wedge \ldots \wedge A_h \to B_1 \vee B_2 \vee \ldots \vee B_k \vee P$$

in T iff $\{A_1, A_2, \ldots, A_h\} \subseteq M$ and $\{B_1, B_2, \ldots, B_k\} \cap M = \phi$.

Intuitively if P is witnessed by C, then P is forced to be in M by the presence in M of the antecedants of C. If T is non-recursive, then circularity is not possible, i.e., two (or more) predicates cannot 'force' each other into a model where possibly neither is necessary (as in $\{A \to B, B \to A\}$).

2.2.2 Theorem. Suppose that M is a model of T, then M is a minimal model iff $M = \{P \in \mathcal{L} \mid P \text{ is witnessed in } M\}$.

Proof ($\to$). Since $M \models T$ it is clear that $\{P \in \mathcal{L} \mid P \text{ is witnessed in } M\} \subseteq M$.

Suppose $P \in M$ is not witnessed, then it is easy to see that $M - \{P\}$ is a model of T: If

$$A_1 \wedge A_2 \wedge \ldots \wedge A_h \to B_1 \vee B_2 \vee \ldots \vee B_k \vee P$$

is in T with each $A_i \in M$, then since P is not witnessed, $\{B_1, B_2, \ldots, B_k\} \cap M \neq \phi$.
($\leftarrow$). Suppose $N \subseteq M$ is a model of T with $N \neq M$. Pick $P \in M - N$ such that P is minimal with respect to the level function ℓ. If

$$A_1 \wedge A_2 \wedge \ldots \wedge A_h \to B_1 \vee B_2 \vee \ldots \vee B_k \vee P$$

witnesses P in M, then by the minimality of $\ell(P)$, $\{A_1, A_2, \ldots, A_h\} \subseteq N$. However $\{B_1, B_2, \ldots, B_k, P\} \cap N = \phi$ contradicting the fact that N is a model of T. ∎

As indicated in the above proof, a predicate $P \in M$ is witnessed in M by rule C iff $M - \{P\} \not\models C$.

A consequence of Theorem 2.2.2 is that for non-recursive databases, models can be tested for minimality in polynomial time.

2.2.3 Definitions. For each $P \in \mathcal{L}$ and $C \in T$, if $P \in \text{conseq}(C)$, let $Wit(P, C)$ be the formula

$$\bigwedge\{A \mid A \in \text{antec}(C)\} \wedge \bigwedge\{\overline{B} \mid B \in \text{conseq}(C) - \{P\}\}$$

(cf. Definition 2.2.1), and let $\omega(P)$ be the formula

$$P \longleftrightarrow \bigvee\{Wit(P, C) \mid C \in T, P \in \text{conseq}(C)\}$$

(meaning P is true iff P is witnessed). Let $\Phi(T)$ be the propositional formulae

$$T \cup \{\omega(P) \mid P \in \mathcal{L}\}.$$

Notice that the size of $\Phi(T)$ is polynomial in that of T.

2.2.4 Theorem. Let v be a valuation on $\mathcal{L}$ such that $v \models \Phi(T)$, then $M = v \cap \mathcal{L}$ (i.e., those predicates true in v) is a minimal model of T.

Proof. Clearly M is a model of T. We thus need to prove that every predicate in M is witnessed. Let $P \in M$, then since $v \models \omega(P)$, we must have that $v \models Wit(P, C)$ for some $C \in T$. But then C witnesses P in M. ▮

2.2.5 Theorem ($\sum_1^P$ - completeness). MM(T, P) iff $P \wedge \Phi(T)$ is consistent.

Proof ($\rightarrow$). Suppose M is a minimal model of T containing P. Define v by $v = M \cup \{\overline{S} \mid S \in \mathcal{L} - M\}$. It is then easy to see that $v \models P \wedge \Phi(T)$.
($\leftarrow$) follows immediately from Theorem 2.2.4. ▮

T is said to be *indefinite relational* iff $\text{antec}(C) = \emptyset$ for each $C \in T$. For such databases, the question of minimal model membership can be decided in polynomial time. The proof of the following theorem is trivial.

2.2.6 Theorem. Let T be indefinite relational. Then MM(T, P) iff there is a rule $B_1 \vee B_2 \vee \ldots \vee B_k \vee P$ in T such that $(\forall D \in T)(\text{conseq}(D) \not\subseteq \{B_1, B_2, \ldots, B_k\})$.

The results of this section (and [3]) suggest that the GCWA is, at least in the worst case, very hard. It should be pointed out however that the average case behaviour is unknown, and this may also be an important factor in determining whether the use of the GCWA is feasible.

Similar results to those presented in this section may be given for the extended GCWA [6,17].

§3. DEFINITENESS.

Given a database T, a *query* is a set $\mathcal{Q} \subseteq \mathcal{L}$. An *answer* to $\mathcal{Q}$ is a set $\mathcal{P} \subseteq \mathcal{Q}$ such that $T \models \bigvee \mathcal{P}$. Ideally we wish to retrieve only minimal answers, i.e., answers which are not properly subsumed by other answers.

Of course in practice we are interested in answering queries of the form $?Q(\mathbf{x})$ in some first order database. The propositional language corresponding to such query processing is the Herbrand base (of the corresponding first order language), and $\mathcal{Q}$ corresponds to the set of ground instances of $Q(\mathbf{x})$. In particular, $\mathcal{L}$ and $\mathcal{Q}$ will typically be very large, thus one of the major problems which arises in answering $\mathcal{Q}$ (or $?Q(\mathbf{x})$) is to determine how large minimal answers can be. Such knowledge could for instance be used as a means of terminating the querying process [10, 12].

In this section we examine the computational complexity of the problem of determining whether a database has minimal answers of some given size. In particular we show that the problem of determining whether a database is indefinite is polynomial for strongly stratified databases (Section 3.2), but $\sum_1^P$ - complete for non-recursive databases (Section 3.3). We also show that the problem of determining whether a database has a minimal answer (to some query) of size $\geq r$ is $\sum_1^P$ - hard for non-recursive databases (Section 3.3) and $\sum_2^P$ - complete for recursive databases (Section 3.4).

3.1 Preliminaries.

3.1.1 Definition. Let T be a deductive database in $\mathcal{L}$ and $\mathcal{P} \subseteq \mathcal{L}$.

(a) T *satisfies* $\mathcal{P}$ iff $T \models \bigvee \mathcal{P}$.

(b) $\mathcal{P}$ is said to be *minimally satisfied* by T iff $T \models \bigvee \mathcal{P}$, but $T \not\models \bigvee \mathcal{P}'$ for any $\mathcal{P}' \subset \mathcal{P}$.

(c) A database is said to be *indefinite* iff it has a minimally satisfied set of size > 1, else the database is *definite*.

The following theorems are trivial.

3.1.2 Theorem. Suppose that $T \models \bigvee \mathcal{P}$, then the following are equivalent.

(a) $\mathcal{P}$ is minimally satisfied by T.

(b) For each $P \in \mathcal{P}$ there is a model M of T such that $M \cap \mathcal{P} = \{P\}$.

(c) For each $P \in \mathcal{P}$, $T \not\models \bigvee(\mathcal{P} - \{P\})$.

3.1.3 Theorem [16]. MM(T, P) iff P belongs to some set $\mathcal{P}$ which is minimally satisfied by T.

3.1.4 Theorem. T is indefinite iff T has two or more minimal models.

Proof. ($\rightarrow$) follows trivially from Theorem 3.1.2.
($\leftarrow$). Let $M_0, M_1, \ldots, M_k$ list all minimal models of T, $k \geq 1$. For $i \leq k$ pick $P_i \in M_i - (M_0 \cap M_1)$. Clearly $T \models \bigvee\{P_i \mid i \leq k\}$. Let $\mathcal{P} \subseteq \{P_i \mid 0 \leq i \leq k\}$ be minimally satisfied in T. If T is definite then $|\mathcal{P}| = 1$, say $\mathcal{P} = \{P_{i_0}\}$. However $P_{i_0} \notin M_0 \cap M_1$, contradicting the fact that $T \models P_{i_0}$. ∎

3.1.5 Theorem. T is indefinite iff T has two models M and N such that $M \cap N$ contains no model of T.

3.2 Strongly stratified databases.

The following theorem shows that deciding whether a strongly stratified database is definite can be achieved in polynomial time.

3.2.1 Theorem. Let T be strongly stratified and $M = \{P \in \mathcal{L} \mid \{P\} \in T\}$. Let M^* be the closure of M under $\{C \in T : |\text{conseq}(C)| = 1\}$. Then T is definite iff $M^* \models T$.

Proof ($\rightarrow$). Let N be the unique minimal model of T, then $M^* \subseteq N$. Suppose that $M^* \subset N$, and pick $P \in N - M^*$ such that $\ell(P)$ is minimal.

Let $M' = M^* \cup \{R \mid \ell(R) \geq \ell(P), R \neq P\}$ then $M' \not\supseteq N$. In order to derive a contradiction, we show that $M' \models T$.

Let $C \in T$ be of the form $A_1 \wedge A_2 \wedge \ldots \wedge A_h \rightarrow B_1 \vee B_2 \vee \ldots \vee B_k$ with antec(C) $\subseteq M'$ and conseq(C) $\cap M' = \emptyset$, then by the definition of M', $\ell(B_1) \leq \ell(P)$ and thus antec(C) $\subseteq M^* \subseteq N$.

If $\ell(B_1) = \ell(P)$, then C must be of the form $A_1 \wedge A_2 \wedge \ldots \wedge A_h \rightarrow P$ and thus $P \in M^*$, a contradiction.

Suppose $\ell(B_1) < \ell(P)$. Since $N \models T$ we have that $\text{conseq}(C) \cap N \neq \emptyset$, and thus by the minimality of $\ell(P)$, $\text{conseq}(C) \cap M^* \neq \emptyset$.

($\leftarrow$). Clearly every model of T contains M^*. ∎

3.3 Non-recursive databases.

In this section we show that for non-recursive databases, the problem of deciding upon the size of minimally satisfied sets is $\sum_1^P$ - hard, and the problem of deciding upon indefiniteness is $\sum_1^P$ - complete.

3.3.1 Theorem ($\sum_1^P$ - hardness). Suppose $\Psi = \exists(P_1, P_2, \ldots, P_m)\ \Phi$ where Φ is propositional in $\{P_1, P_2, \ldots, P_m\} \subseteq \mathcal{L}$. Let

$$T_\beta(\Psi) = \{DD(\neg\Phi)\} \cup \{A_k \to Y\} \cup \{Y \vee \bigvee_{\alpha=1}^{\beta} X_\alpha\} \cup \{X_\alpha \to P_i \vee P_i' \mid \alpha \leq \beta, i \leq m\},$$

where $\{X_\alpha\}_{\alpha \leq \beta}$ and Y are new predicates not occurring in $\mathcal{L}' \cup \{A_j \mid j \leq k\}$ and $A_k = \text{Pred}(\neg\Phi)$ (cf. Section 1.6). Notice that $T_\beta(\Psi)$ is non-recursive.

Then the following are equivalent:

(a) Ψ is true.

(b) $T_\beta(\Psi)$ has a minimally satisfied set of size $\beta + 1$.

(c) $T_\beta(\Psi)$ has a minimally satisfied set of size $\geq \beta + 1$.

(d) $T_\beta(\Psi)$ is indefinite.

Proof (a) $\to$ (b). Let v be a valuation on $\mathcal{L}$ satisfying Φ, and $M = \{P_i \mid P_i \in v\} \cup \{P_i' \mid \overline{P}_i \in v\}$. Let $\mathcal{P} = \{Y\} \cup \{X_\alpha \mid \alpha \leq \beta\}$, then clearly $T_\beta(\Psi) \models \bigvee \mathcal{P}$.

Let $M_\alpha = M \cup cl(\neg\Phi, M) \cup \{X_\alpha\}$, then by Theorem 1.10, $A_k \notin M_\alpha$ thus $M_\alpha \models T_\beta(\Psi)$. Also $M_\alpha \cap \mathcal{P} = \{X_\alpha\}$ and $\{Y\} \models T_\beta(\Psi)$, thus by Theorem 3.1.2, $\mathcal{P}$ is minimally satisfied by $T_\beta(\Psi)$.

(b) $\to$ (c) and (c) $\to$ (d) are trivial.

(d) $\to$ (a). Clearly $\{Y\}$ is a minimal model of $T_\beta(\Psi)$. If $T_\beta(\Psi)$ is indefinite, then $T_\beta(\Psi)$ has some model M such that $Y \notin M$. In particular, $A_k \notin M$. But then $X_\alpha \in M$ for some α, and hence $|M \cap \{P_i, P_i'\}| \geq 1$ for each $i \leq m$.

Pick $N \subseteq M$ such that $|N \cap \{P_i, P'_i\}| = 1$ for each $i \leq m$. But then by Theorem 1.9, $cl(\neg\Phi, N) \subseteq M$ and hence $A_k \notin cl(\neg\Phi, N)$. By Theorem 1.10, $\{P_i \mid P_i \in N\} \cup \{\overline{P}_i \mid P'_i \in N\}$ is a valuation satisfying Φ. ∎

We can again use the witness property to show that determining whether a non-recursive database is indefinite is a $\sum_1^P$ - complete problem. Recall (Theorem 2.2.2) that when T is non-recursive, a model M is a minimal model iff every predicate in M is witnessed. In our case, we are interested in determining whether there are two (or more) minimal models. We thus introduce, for each $P \in \mathcal{L}$, two further predicates $W_0(P)$ and $W_1(P)$. Given a rule C of the form

$$A_1 \wedge A_2 \wedge \ldots \wedge A_h \rightarrow B_1 \vee B_2 \vee \ldots \vee B_k,$$

let C^i $(i \in \{0, 1\})$ denote the rule

$$W_i(A_1) \wedge W_i(A_2) \wedge \ldots \wedge W_i(A_h) \rightarrow W_i(B_1) \vee W_i(B_2) \vee \ldots \vee W_i(B_k)$$

If $P \in \text{conseq}(C)$, let $Wit_i(P, C)$ be the formula

$$\bigwedge\{W_i(A) \mid A \in \text{antec}(C)\} \wedge \bigwedge\{\overline{W_i(B)} \mid B \in \text{conseq}(C) - \{P\}\}$$

(cf. Definition 2.2.3), and let $\omega_i(P)$ be the formula

$$W_i(P) \longleftrightarrow \bigvee\{Wit_i(P, C) \mid C \in T, P \in \text{conseq}(C)\}$$

Let $\Phi(T)$ be the propositional formulae

$$\{C^i \mid i \in \{0, 1\}, C \in T\} \cup \{\omega_i(P) \mid i \in \{0, 1\}, P \in \mathcal{L}\}.$$

3.3.2 Theorem ($\sum_1^P$ - completeness). Let T be a non-recursive deductive database in $\mathcal{L}$, then T is indefinite iff $\Phi(T) \wedge \bigvee\{\, W_1(P) \wedge \overline{W_2(P)} \mid P \in \mathcal{L}\}$ is consistent.

The proof is similar to that of Theorem 2.2.5, and we thus leave the details to the reader. The problem of deciding whether T has a minimally satisfied set of size β is considered again in Section 3.5.

3.4 Recursive databases.

In this section we show that for recursive databases the problem of deciding upon the size of minimally satisfied sets is $\sum_2^P$ - complete. As mentioned in the introduction, such results are also of interest since naturally occurring $\sum_k^P$ - complete problems are rare for $k > 1$.

3.4.1 Theorem ($\sum_2^P$ - hardness). Let T be a deductive database in $\mathcal{L}$, $P \in \mathcal{L}$. Let $T^*_\beta = T \cup \{P \rightarrow \bigvee_{\alpha=1}^{\beta} X_\alpha\}$, where $\beta > |\mathcal{L}|$.

Then MM(T, P) iff T^*_β has a minimally satisfied set of size $\geq \beta$.

Proof ($\rightarrow$). By Theorem 3.1.3, we may find a set $\mathcal{P}$ which is minimally satisfied by T such that $P \in \mathcal{P}$. But then $(\mathcal{P} - \{P\}) \cup \{X_\alpha \mid \alpha \leq \beta\}$ is minimally satisfied by T^*_β.
($\leftarrow$). Let $\mathcal{P}$ be minimally satisfied in T^*_β with $|\mathcal{P}| \geq \beta$. Since $\beta > |\mathcal{L}|$, we must have that $\mathcal{P} \cap \{X_\alpha \mid \alpha \leq \beta\} \neq \emptyset$, and thus by Theorem 3.1.3 we may find a minimal model M of T^*_β containing some X_α. But then $M \cap \mathcal{L}$ is clearly a minimal model of T containing P. ▮

We can easily prove $\sum_2^P$ - completeness.

3.4.2 Theorem. T has a minimally satisfied set of size $\geq r$ iff there are r models M_1, M_2, $\ldots$, M_r of T and a set $\mathcal{Q} \subseteq \mathcal{L}$ such that

(a) $T \models \bigvee \mathcal{Q}$, and

(b) for $1 \leq i < j \leq r$, $M_i \cap M_j \cap \mathcal{Q} = \emptyset$.

The proof is trivial using Theorem 3.1.2. Clearly the above set of conditions can be expressed via a $\sum_2$ statement.

3.4.3 Theorem ($\sum_2^P$ - completeness). Let T be given and $\Phi(T)$ be the formula

$$\bigwedge_{i=1}^{r} T^i \wedge \{T \rightarrow \bigvee_{l=1}^{n} (P_l \wedge Q_l)\} \wedge \bigwedge \{Q_l \rightarrow \neg(P_l^i \wedge P_l^j) \mid 1 \leq l \leq n, 1 \leq i < j \leq r\}$$

where T^i is formed from T by replacing each P_l by P_l^i.

Then $\exists(P_1^1, P_2^1, \ldots, P_n^1, P_1^2, P_2^2, \ldots, P_n^r, Q_1, Q_2, \ldots, Q_n)\,\forall(P_1, P_2, \ldots, P_n)\ \Phi(T)$ is true iff T has a minimally satisfied set of size $\geq r$.

Proof. It is easy to check that $\mathcal{Q} = \{P_l \mid Q_l \text{ is true}\}$ and $M_i = \{P_l \mid P_l^i \text{ is true}\}$ $(i \leq r)$ satisfy the conditions of the previous theorem. ∎

3.5 Reducing the complexity.

In this section, we briefly mention that it may be possible to reduce the complexity of minimal model problems, by using what we know to be true (in T) in order to aid the test for minimal model membership.

Specifically, throughout this section we assume that $\mathcal{Q} \subseteq \mathcal{L}$ and that no minimal model of T contains two or more predicates from $\mathcal{Q}$. Such "uniqueness constraints" are common in database theory.

3.5.1 Theorem. Suppose that $T \models \bigvee \mathcal{Q}$.

(a) If $P \in \mathcal{Q}$, then MM$(T,\ P)$ iff $T \not\models \bigvee(\mathcal{Q} - \{P\})$.

(b) The problem of determining MM for elements of $\mathcal{Q}$ is $\sum_1^P$ - complete.

The proof is trivial. In part (b), $\sum_1^P$ - hardness follows as in Theorem 3.3.1. Thus under the stated conditions, the problem of determining minimal model membership is reduced from $\sum_2^P$ to $\sum_1^P$ - completeness.

3.5.2 Theorem. If $T \models \bigvee \mathcal{Q}$, then $\mathcal{P} = \{P \in \mathcal{Q} \mid \text{MM}(T, P)\}$ is minimally satisfied by T and is the only minimally satisfied subset of $\mathcal{Q}$.

The problem of deciding upon the size of the minimally satisfied subset of $\mathcal{Q}$ is equivalent to deciding upon the truth of the conjunct of a $\sum_1$ and a $\prod_1$ formula. In the terminology of [18, 26], it is D^P - complete.

3.5.3 Theorem (D^P - hardness). Let $\Psi_1 = \exists(P_1, P_2, \ldots, P_m)\,\Phi_1$ and $\Psi_2 = \forall(P_{m+1}, P_{m+2}, \ldots, P_r)\,\Phi_2$, where Φ_1 and Φ_2 are propositional in $\{P_1, P_2, \ldots, P_m\} \subseteq$

$\mathcal{L}$ and $\{P_{m+1}, P_{m+2}, \ldots, P_r\} \subseteq \mathcal{L}$ respectively. Clearly we may assume that $\{P_1, P_2, \ldots, P_m\}$ and $\{P_{m+1}, P_{m+2}, \ldots, P_r\}$ are disjoint. Let

$$\begin{aligned} T_\beta \quad = \quad & \{DD(\neg\Phi_1)\} \cup \{DD(\Phi_2)\} \cup \{A_{k_1} \to Y\} \cup \{A'_{k_2} \to Y \vee \bigvee_{\alpha=1}^{\beta} X_\alpha\} \cup \\ & \{X_\alpha \to P_i \vee P'_i \mid \alpha \le \beta, i \le m\} \cup \{P_i \vee P'_i \mid m < i \le r\} \end{aligned}$$

where $\{X_\alpha\}_{\alpha \le \beta}$ and Y are new predicates, $A_{k_1} = \text{Pred}(\neg\Phi_1)$, $A'_{k_2} = \text{Pred}(\Phi_2)$, and $\beta > 2|\mathcal{L}| + k_1 + k_2$. Notice that T_β is non-recursive, and that $\mathcal{Q} = \{X_\alpha \mid \alpha \le \beta\} \cup \{Y\}$ satisfies the conditions given at the beginning of this section. Then the following are equivalent:

(a) $\Psi_1 \wedge \Psi_2$ is true.

(b) T_β has a minimally satisfied set of size $\beta + 1$

(c) $\mathcal{Q}$ is minimally satisfied by T_β.

(d) $\mathcal{Q}$ has a minimally satisfied subset of size > 1.

Proof (a)→(b). We show that $\mathcal{Q}$ is minimally satisfied by T_β. Clearly $\mathcal{Q}$ is satisfied since Ψ_2 is true.

Let v be a valuation on $\{P_1, P_2, \ldots, P_m\}$ which satisfies Φ_1, and $M = \{P_i \mid i \le m, P_i \in v\} \cup \{P'_i \mid i \le m, P_i \notin v\}$. Let $N = \{P_i \mid m < i \le r\}$, then for each $Z \in \mathcal{Q}$, $N \cup cl(\Phi_2, N) \cup M \cup cl(\neg\Phi_1, M) \cup \{Z\}$ is a model of T_β. The result then follows from Theorem 3.1.2.

(b) → (c). Let $\mathcal{R}$ be a minimally satisfied set of size $\beta + 1$. Since $\beta > 2|\mathcal{L}| + k_1 + k_2$, we must have that $\mathcal{R} \cap \{X_\alpha \mid \alpha \le \beta\} \ne \emptyset$, and thus by symmetry $\mathcal{R} \supset \{X_\alpha \mid \alpha \le \beta\}$.

By Theorem 3.1.2 we may find a minimal model M of T_β such that $M \cap \mathcal{R} = \{X_1\}$, whence $(M - \{X_1\}) \cup \{Y\}$ models T_β. But then $\emptyset \ne (((M - \{X_1\}) \cup \{Y\}) \cap \mathcal{R}) \subseteq \{Y\}$, thus $Y \in \mathcal{R}$ and $\mathcal{R} = \mathcal{Q}$.

(c) → (d) is trivial.

(d) → (a). We first show that if $\mathcal{Q}$ is satisfied, then Ψ_2 must be true. Let v be a valuation on $\{P_{m+1}, P_{m+2}, \ldots, P_r\}$ such that $v \models \neg\Phi_2$, and let $M = \{P_i \mid m < i \le r, P_i \in v\} \cup \{P'_i \mid m < i \le r, P_i \notin v\}$, then $M \cup cl(\Phi_2, M)$ models T_β and is disjoint from $\mathcal{Q}$.

If $\mathcal{Q}$ has a minimally satisfied subset of size > 1, then by Theorem 3.1.2, there is a minimal model of T_β which contains some X_α but fails to contain Y, thus proving Ψ_1. ∎

3.5.4 Theorem (D^P - completeness). $\mathcal{Q}$ has a minimally satisfied subset $\mathcal{P} \subseteq \mathcal{Q}$ such that $|\mathcal{P}| \ge r$ iff

(a) $T \models \bigvee \mathcal{Q}$, and

(b) there exists r models of M_1, M_2, $\ldots$, M_r of T such that for each $1 \leq i < j \leq r$, $M_i \cap M_j \cap \mathcal{Q} = \emptyset$.

Quite clearly, (a) can be expressed by a $\prod_1$ statement, and (b) via a $\sum_1$ statement (cf. Theorem 3.4.3).

3.5.5 Corollary (D^P - hardness). The problem of determining whether a non-recursive database has a minimally satisfied set of size β is D^P - hard.

3.6 Open questions.

(1) To what extent can known information about T be used to assist in the computation of the GCWA and other minimal model problems. For instance, when considering the minimally satisfied subsets of some $\mathcal{Q} \subseteq \mathcal{L}$, can other relationships between T and $\mathcal{Q}$ be used to reduce the complexity?

The precise computational complexity of the following are open.

(2) The problem of determining whether a recursive database is indefinite.

(3) The problem of determining whether a non-recursive database has a minimally satisfied set of size $\geq \beta$.

By Theorem 3.3.1, both (2) and (3) above are $\sum_1^P$ - hard.

(4) The problem of determining whether a database has a minimally satisfied set of size β. The problem is D^P - hard (by Corollary 3.3.5) and, as in Theorem 3.4.2, can be shown to be in the class $\sum_2^P$.

§4. COMPUTING MINIMAL MODEL MEMBERSHIP FOR NON-RECURSIVE DATABASES.

In this section we examine the structure of minimal models for non-recursive databases, and develop algorithms for determining minimal model membership. The material of this section also provides insight that is useful when we move on to consider recursive databases in Section 5.

4.1 Minimal model structure.

In order to develop the witness property more fully, we need to extend the definition of MM given in Section 1.2.

4.1.1 Definition. If $\mathcal{P}$ and $\mathcal{O}$ are disjoint subsets of $\mathcal{L}$, then define $\mathrm{MM}(T, \mathcal{P}, \mathcal{O})$ iff T has a minimal model M such that $\mathcal{P} \subseteq M$ and $M \cap \mathcal{O} = \emptyset$.

Quite clearly MM can be regarded as an extension of both the GCWA and the EGCWA [6, 17].

4.1.2 Proposition. P is in some minimal model of T iff $\mathrm{MM}(T, \{P\}, \emptyset)$.

4.1.3 Theorem. Suppose that $\mathcal{P} = \emptyset$, then $\mathrm{MM}(T, \mathcal{P}, \mathcal{O})$ iff $T \not\models \bigvee \mathcal{O}$.

4.1.4 Theorem. Let $\mathcal{P}$ and $\mathcal{O}$ be disjoint subsets of $\mathcal{L}$ and P be an arbitrary predicate in $\mathcal{P}$. Then $\mathrm{MM}(T, \mathcal{P}, \mathcal{O})$ iff there is a rule $C \in T$ such that:

(i) $\mathcal{P} \cap \mathrm{conseq}(C) = \{P\}$,

(ii) $\mathrm{antec}(C) \cap \mathcal{O} = \emptyset$, and

(iii) $\mathrm{MM}(T, (\mathcal{P} - \{P\}) \cup \mathrm{antec}(C), \mathcal{O} \cup (\mathrm{conseq}(C) - \{P\}))$.

Note. Recall that $\mathrm{antec}(C)$ and $\mathrm{conseq}(C)$ are assumed disjoint. Thus if $\mathcal{P}$ and $\mathcal{O}$ are disjoint, and C satisfies conditions (i) and (ii), then $(\mathcal{P} - \{P\}) \cup \mathrm{antec}(C)$ and $\mathcal{O} \cup (\mathrm{conseq}(C) - \{P\})$ are also disjoint.

Proof ($\rightarrow$). Suppose that $\mathrm{MM}(T, \mathcal{P}, \mathcal{O})$ with M the appropriate minimal model of T. Now, $P \in M$ and hence by Theorem 2.2.2, P is witnessed in M (cf. Definition 2.2.1) by some rule $C \in T$, whence $M - \{P\} \not\models C$. It is then easy to check that C satisfies conditions (i), (ii) and (iii).

($\leftarrow$). Let M be a minimal model of T such that $(\mathcal{P} - \{P\}) \cup \mathrm{antec}(C) \subseteq M$ and $M \cap (\mathcal{O} \cup (\mathrm{conseq}(C) - \{P\})) = \emptyset$. Since $M \models C$, we have that $P \in M$, thus $\mathcal{P} \subseteq M$. Hence M satisfies $\mathrm{MM}(T, \mathcal{P}, \mathcal{O})$. ∎

Note that the statement of the above theorem needs careful reading. If $\mathrm{MM}(T, \mathcal{P}, \mathcal{O})$, then for each $P \in \mathcal{P}$ we may find an appropriate rule $C \in T$ (satisfying conditions (i) - (iii)). On the other hand, if there exists *any* $P \in \mathcal{P}$ for which such a $C \in T$ exists, then we may conclude that $\mathrm{MM}(T, \mathcal{P}, \mathcal{O})$ holds.

4.2 Computing minimal model membership.

4.2.1. The above theorems suggest the following algorithm for deciding upon the truth or falsity of $\mathrm{MM}(T, \mathcal{P}, \mathcal{O})$.

```
MINMOD (T, 𝒫, 𝒪)
{  if 𝒫 = ∅ {return(T ⊭ ⋁ 𝒪)};
   pick P ∈ 𝒫;
   for each  C ∈ T
        {if (  𝒫 ∩ conseq(C) = {P}  and  antec(C) ∩ 𝒪 = ∅  )
              {if MINMOD (T, (𝒫 − {P}) ∪ antec(C), 𝒪 ∪ (conseq(C) − {P}))
                        {return(true)}
        }};
   return(false)
}
```

4.2.2 Definition. Let T be non-recursive in $\mathcal{L}$ with level function $\ell : \mathcal{L} \rightarrow \{1, 2, \ldots, n\}$, where $n = |\mathcal{L}|$. Suppose that $\mathcal{P} \subseteq \mathcal{L}$ then define $deg_\ell(\mathcal{P}) = (x_n, x_{n-1}, \ldots, x_1) \in \mathbb{N}^n$ by

$x_i = |\{P \in \mathcal{P} : \ell(P) = i\}|$.

4.2.3 Theorem (Termination). If T is non-recursive, then MINMOD(T, $\mathcal{P}$, $\mathcal{O}$) terminates.

Proof. Let $\prec$ be the lexicographic ordering on $\mathbb{N}^n$. If C is a rule with $P \in \text{conseq}(C) \cap \mathcal{P}$, then $deg_\ell((\mathcal{P} - \{P\}) \cup \text{antec}(C)) \prec deg_\ell(\mathcal{P})$. The result then follows by induction on $deg_\ell(\mathcal{P})$. ∎

4.2.4 Theorem (Correctness). If MINMOD $(T, \mathcal{P}, \mathcal{O})$ terminates, then MM$(T, \mathcal{P}, \mathcal{O})$ iff MINMOD$(T, \mathcal{P}, \mathcal{O})$ returns true.

§5. COMPUTING MINIMAL MODEL MEMBERSHIP FOR RECURSIVE DATABASES.

In Section 5.1 we show that in order to test the minimality of a model in the recursive case, we need to extend the witness property (Definition 2.2.1) to handle subsets of models (rather than individual elements). This then leads us to the notions of weak deduction and cyclic trees, which are shown to characterise minimal model membership. Using cyclic trees, methods are presented for testing model minimality (Section 5.2) and minimal model membership (Section 5.3). In Sections 5.4 and 5.5 we examine restricted forms of cyclic trees that may improve the efficiency of such methods (by reducing the search space). Finally, Section 5.6 shows that cyclic trees may be used as a means of compiling the GCWA.

5.1 Minimal model structure.

For recursive databases, the witness property (as defined in Section 2.2) fails to guarantee the minimality of a model. For instance if $T = \{A \to B, B \to A\}$ and $M = \{A, B\}$, then both A and B are witnessed, but clearly M is not minimal. On the other hand if $T = \{A \to B, B \to A, A \vee B\}$ and $M = \{A, B\}$, then M *is* minimal: the rule $A \vee B$ "witnesses" that $\{A, B\} \cap M \neq \emptyset$, and the "cycle" $A \to B \to A$ guarantees that if $\{A, B\} \cap M \neq \emptyset$, then both A and B must lie in M. Thus we see that in order to guarantee the minimality of a model in the recursive case, it is *subsets* of M that need to be witnessed, rather than just individual elements. The following is analogous to Definition 2.2.1.

5.1.1 Definition. Let $M \models T$ and $\mathcal{P}_0 \subseteq M$, then C *witnesses* $\mathcal{P}_0$ iff $\text{antec}(C) \subseteq M - \mathcal{P}_0$ and $(\text{conseq}(C) - \mathcal{P}_0) \cap M = \emptyset$.

Note that the condition $(\text{conseq}(C) - \mathcal{P}_0) \cap M = \emptyset$ can alternatively be expressed as $\text{conseq}(C) \cap (M - \mathcal{P}_0) = \emptyset$ or as $\text{conseq}(C) \cap M \subseteq \mathcal{P}_0$, and we shall use whichever of these formulations appears to be the most natural at any given instant. Thus we see that C witnesses $\mathcal{P}_0$ iff $M - \mathcal{P}_0 \not\models C$. As in Section 2.2, the presence in M of the predicates from $\text{antec}(C)$ forces *some* predicate from $\mathcal{P}_0$ into M. The following is analogous to Theorem 4.1.4.

5.1.2 Theorem. Let $\mathcal{P}$ and $\mathcal{O}$ be disjoint subsets of $\mathcal{L}$, and $\mathcal{P}_0$ be an arbitrary non-empty subset of $\mathcal{P}$. Then $\mathrm{MM}(T, \mathcal{P}, \mathcal{O})$ iff there is a rule $C \in T$ such that:

(i) $\mathcal{P}_0 \cap \mathrm{conseq}(C) \neq \emptyset$,

(ii) $\mathrm{antec}(C) \cap (\mathcal{O} \cup \mathcal{P}_0) = \emptyset$,

(iii) $(\mathrm{conseq}(C) - \mathcal{P}_0) \cap \mathcal{P} = \emptyset$, and

(iv) $\mathrm{MM}(T, \mathcal{P} \cup \mathrm{antec}(C), \mathcal{O} \cup (\mathrm{conseq}(C) - \mathcal{P}_0))$.

Proof. If M satisfies MM(T, $\mathcal{P}$, $\mathcal{O}$), then $\mathcal{P}_0 \subseteq M$, and by the minimality of M we may find a $C \in T$ such that $M - \mathcal{P}_0 \not\models C$. Thus C witnesses $\mathcal{P}_0$ in M, and it is easy to check that C satisfies conditions (i) - (iv). The converse is trivial. ▮

Quite clearly, a model M of T is minimal iff every subset of M is witnessed. Of course checking that the witness property holds for subsets presents more of a computational challenge, since there are considerably more subsets than elements. Our aim is therefore to find a set $\mathcal{C} \subseteq \{X \mid X \subseteq \mathcal{L}\}$ such that checking the minimality of M amounts to checking the witness property for each element of $\{X \in \mathcal{C} \mid X \subseteq M\}$. Such a computation will be made more efficient by choosing $\mathcal{C}$ in such a way that $\mathcal{C}$ and its elements are small. The above example suggests that the set of "cycles" (suitably defined) is a good candidate for $\mathcal{C}$.

We first present our basic (tree) structure with which to represent the witnessing relationships between rules and subsets of $\mathcal{L}$.

5.1.3 Example. Suppose that T consists of the following rules:

1. $Q_2 \wedge Q_3 \rightarrow P \vee R_1$
2. $P \rightarrow Q_2 \vee R_2$
3. $S_2 \rightarrow Q_3 \vee R_3$
4. $S_3 \rightarrow P \vee Q_2 \vee R_4$
5. $S_2 \vee R_5$
6. $S_1 \rightarrow Q_3 \vee Q_2$
7. $S_3 \vee R_7$
8. $R_1 \rightarrow Q_2$

and we wish to determine whether P lies in some minimal model of T, i.e. whether MM(T, $\{P\}$, $\emptyset$). Using the terminology from Theorem 5.1.2, we set $\mathcal{P} = \{P\}$ and $\mathcal{O} = \emptyset$. We first look for a rule satisfying conditions (i) - (iii) of Theorem 5.1.2 (i.e. which might witness $\mathcal{P}_0 = \{P\}$ in some minimal model), the only options being rules 1 and 4; say we choose rule 1. We will represent this witnessing relationship using a tree structure. Since (we are guessing that) rule 1 witnesses $\{P\}$, we depict this using the tree $\mathcal{T}_1$ (Figure 5.1.3(i)), and Theorem 5.1.2 (condition (iv)) then suggests that we attempt to show that MM(T, $\{P, Q_2, Q_3\}$, $\{R_1\}$). We thus reset $\mathcal{P} = \{P, Q_2, Q_3\}$, $\mathcal{O} = \{R_1\}$ and continue.

We will concentrate on witnessing the leaf nodes within our tree or the cycles within the branches (if such cycles exist). Thus we set $\mathcal{P}_0 = \{Q_2\}$, and re-apply Theorem 5.1.2: In this case the only rule satisfying conditions (i) - (iii) of Theorem 5.1.2 is rule 2, thus yielding the tree $\mathcal{T}_2$ (Figure 5.1.3(i)), and again Theorem 5.1.2 then suggests that we attempt to show that MM(T, $\{P, Q_2, Q_3\}$, $\{R_1, R_2\}$). Thus $\mathcal{P}$ is unchanged, but $\mathcal{O}$ is reset to $\{R_1, R_2\}$.

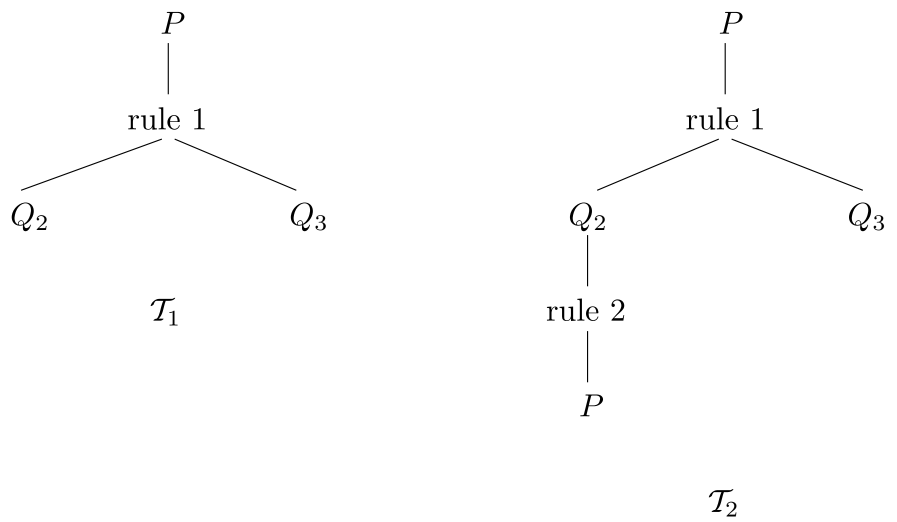

Figure 5.1.3(i).

Notice that the left hand branch of $\mathcal{T}_2$ forms a cycle. We thus reset $\mathcal{P}_0 = \{P, Q_2\}$, and the only rule which satisfies the conditions of Theorem 5.1.2 is rule 4. Rule 7 then terminates the branch (since rule 7 has no antecedants), thus yielding $\mathcal{T}_3$ (Figure 5.1.3(ii)). We thus reset $\mathcal{P} = \{P, Q_2, Q_3, S_3\}$ and $\mathcal{O} = \{R_1, R_2, R_4, R_7\}$ and (by Theorem 5.1.2) attempt to prove MM(T, $\mathcal{P}$, $\mathcal{O}$).

We thus move on to the problem of witnessing $\{Q_3\}$, thus again we set $\mathcal{P}_0 = \{Q_3\}$, and look for a rule satisfying the conditions of Theorem 5.1.2. In this case, rule 3 is the only possibility. Rule 5 is then the only possible witness for S_2, and again this terminates the branch, yielding $\mathcal{T}_4$ (Figure 5.1.3(ii)).

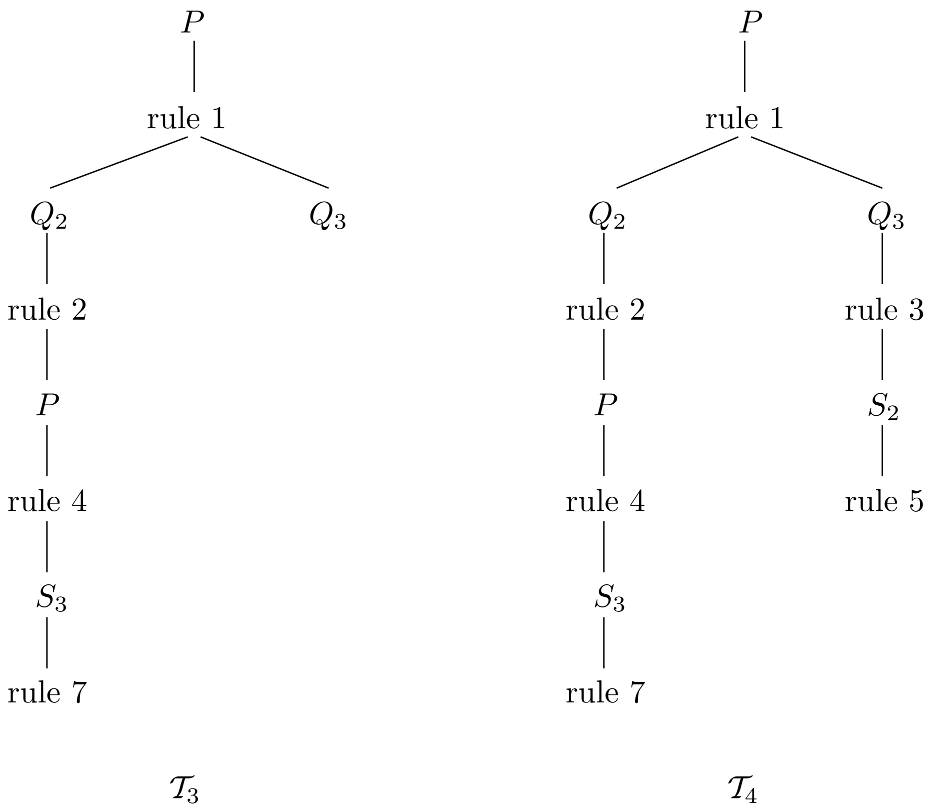

Figure 5.1.3(ii).

Theorem 5.1.2 then suggests that we should try to prove MM(T, $\{P, Q_2, Q_3, S_3, S_2\}$, $\{R_1, R_2, R_4, R_7, R_3, R_5\}$), i.e., that we should search for a minimal model of T containing $\{P, Q_2, Q_3, S_3, S_2\}$ and disjoint from $\{R_1, R_2, R_4, R_7, R_3, R_5\}$.

This appears harder than our initial problem (i.e., MM(T, $\{P\}$, $\emptyset$)). Note however that if M is *any* model of T which is disjoint from $\{R_1, R_2, R_4, R_7, R_3, R_5\}$, then (using the rules in the tree) we may infer that $\{P, Q_2, Q_3, S_3, S_2\} \subseteq M$. In particular, if we can show that $T \not\models R_1 \vee R_2 \vee R_4 \vee R_7 \vee R_3 \vee R_5$, then we may infer MM($T$, $\{P\}$, $\emptyset$).

Notice that at each stage we are making a guess about which rule will witness a particular set. If this guess subsequently turns out to be incorrect, then we would need to "backtrack", and consider other possibilities. In the above example, the only stage at which there was an alternative was during the first step (when we could have chosen rule 4).

The following definition captures most of the features seen in the above example.

5.1.4 Definition. Let $P \in \mathcal{L}$. A *weak deduction tree* for P in T is a tree $\mathcal{T}$ containing two types of nodes, rule nodes and predicate nodes, satisfying the following conditions.

(1) The root node $root(\mathcal{T})$ (at the top of the tree) is a predicate node labelled with P.

(2) Each rule node is labelled with a rule from T, and each predicate node is labelled with a predicate in $\mathcal{L}$. The label of a node N will be denoted by $lab(N)$. (Predicate nodes will be denoted by $N, N', \ldots$ etc, or in the form N_R where R is the label of node N. A rule node whose label is rule C will be denoted by RN_C.)

$$\text{Let } \; Pred(\mathcal{T}) =_{def} \{lab(N) \mid N \text{ is a predicate node in } \mathcal{T}\}.$$

(3) If RN_C is a rule node (labelled with rule C), then for each predicate R in antec(C), RN_C has a child node labelled with R. RN_C has no other child nodes.
If N is a predicate node (other than the root) with $lab(N) = R$, then its parent node is a rule node RN_D with $R \in$ antec(D). A predicate node can have at most one child node. If the child exists, then it must be a rule node satisfying condition (5) below.
(If node1 is the parent of node2, then we write node1>node2, the > relationship being transitive. Thus $root(\mathcal{T}) \geq N$ for each node N in $\mathcal{T}$.)

(4) $\mathcal{T}$ is finite.

(5) Whenever RN_C is a rule node with parent N, then conseq(C) $\cap \{lab(N') \mid N'$ is a predicate node, $N' \geq N\} \neq \emptyset$, and $Pred(\mathcal{T}) \cap ($conseq$(C) - \{lab(N') \mid N'$ is a predicate node, $N' \geq N\}) = \emptyset$.

Note that if RN_C has parent node N, then condition (5) does *not* imply that $lab(N) \in$ conseq(C).

As indicated in Example 5.1.3, the intended relationship between $\mathcal{T}$ and some minimal model M is that $Pred(\mathcal{T}) \subseteq M$. Thus if RN_C is a rule node with parent N, then the motivation behind condition (5) is that C should witness $\{lab(N') \mid N'$ is a predicate node, $N' \geq N\}$ (cf. condition (iii) of Theorem 5.1.2). Condition (ii) of Theorem 5.1.2 (i.e., antec(C) $\cap \{lab(N') \mid N'$ is a predicate node, $N' \geq N\} = \emptyset$) will be included at a later stage (cf. the proof of Theorem 5.1.14 and Definition 5.1.15).

Following the discussion above, we intend to insist that C should in fact witness the cycle within $\{lab(N') \mid N'$ is a predicate node, $N' \geq N\}$, this being defined as follows.

5.1.5 Definition. Let $\mathcal{T}$ be a weak deduction tree. Given a predicate node N define:

(i) $\text{ACT}(N) = \{lab(N') \mid N' \text{ is a predicate node, } N' \geq N\}$,

(ii) $top(N)$ to be the unique predicate node such that $top(N) \geq N$, $lab(top(N)) = lab(N)$, and $\forall N' > top(N)(lab(N') \neq lab(N))$, and

(iii) $\text{CYC}(N) = \{lab(N') \mid N' \text{ is a predicate node, } top(N) \geq N' \geq N\}$.

The terminology $\text{ACT}(N)$ comes from path checking algorithms ([8]) in which the current path/branch being examined is the ACTive path.

Note that it is quite possible to have $\text{CYC}(N) = \{lab(N)\}$ (if for instance $top(N) = N$). Other possible definitions for the notion of cycle are given in Sections 5.4 and 5.5.

If RN_C is a rule node in $\mathcal{T}$ with parent N, then define

$$\mathcal{O}(RN_C) = \text{conseq}(C) - \text{ACT}(N),$$

and

$$\mathcal{O}(\mathcal{T}) = \bigcup\{\mathcal{O}(RN_C) \mid RN_C \text{ is a rule node in } \mathcal{T}\}.$$

Thus condition (5) of Definition 5.1.4 implies that $Pred(\mathcal{T}) \cap \mathcal{O}(\mathcal{T}) = \emptyset$. Again the motivation is that if M is some minimal model with $Pred(\mathcal{T}) \subseteq M$, and if RN_C has parent N with C witnessing $\text{ACT}(N)$ in M, then we must have that $Pred(\mathcal{T}) \cap \mathcal{O}(RN_C) \subseteq M \cap \mathcal{O}(RN_C) = \emptyset$ (cf. Definition 5.1.1).

5.1.6 Definition. Let $\mathcal{T}$ be a weak deduction tree, then $\mathcal{T}$ is said to be:

(i) *unfactored* iff each leaf node is a rule node (which must, by condition (3) of Definition 5.1.4, be labelled with a rule in $\text{EXT}(T) = \{C \in T \mid \text{antec}(C) = \emptyset\}$ (cf. §1)), and

(ii) a *cyclic* tree iff for each rule node RN_C in $\mathcal{T}$, if RN_C has parent N then $\text{conseq}(C) \cap \text{ACT}(N) \subseteq \text{CYC}(N)$.

Thus in a cyclic tree, $\text{conseq}(C)$ - $\text{ACT}(N)$ = $\text{conseq}(C)$ - $\text{CYC}(N)$ is disjoint from $Pred(\mathcal{T})$, with the intention that C should witness $\text{CYC}(N)$.

Aside. Whilst factorisation is not discussed at length in the current paper, the term "unfactored" is used here to maintain consistency with [10, 11]. An unfactored tree is one in which every branch has been maximally extended. When generating weak deduction trees we will, in the interest of computational efficiency, wish to prune redundant branches from such trees. One of the most common methods used in such pruning is factorisation [8, 10, 15], hence the terminology.

5.1.7 Theorem. Suppose that $P \in \mathcal{L}$ and $\mathcal{T}$ is an unfactored weak deduction tree for P in T. Then $T \models P \vee \bigvee \mathcal{O}(\mathcal{T})$.

Proof. Suppose that $M \models T$ with $M \cap (\{P\} \cup \mathcal{O}(\mathcal{T})) = \emptyset$. Let N_0 be the root node of $\mathcal{T}$, then we inductively construct an (infinite) branch $N_0 > \mathrm{RN}_{C_0} > N_1 > \mathrm{RN}_{C_1} > N_2 > \ldots$ through $\mathcal{T}$ such that for each $i \geq 0$:

(i) $lab(N_i) \notin M$, and

(ii) N_i is the parent of RN_{C_i} which in turn is the parent of N_{i+1}.

Suppose that for each $i \leq j$, $lab(N_i) \notin M$. Since $\mathcal{T}$ is unfactored, N_j is not a leaf, and thus RN_{C_j} exists with

$$\mathrm{conseq}(C_j) = \mathcal{O}(RN_{C_j}) \cup (\mathrm{conseq}(C_j) - \mathcal{O}(RN_{C_j})) \subseteq \mathcal{O}(\mathcal{T}) \cup \mathrm{ACT}(N_j).$$

Now $\mathcal{O}(\mathcal{T}) \cap M = \emptyset$ by hypothesis, and $\mathrm{ACT}(N_j) \cap M = \emptyset$ by condition (i). Thus $\mathrm{conseq}(C_j) \cap M = \emptyset$. Since $M \models T$, we may find a child node N_{j+1} of RN_{C_j} such that $lab(N_{j+1}) \notin M$.

Finally, the existence of an infinite branch through $\mathcal{T}$ contradicts condition (4) of Definition 5.1.4. ▮

5.1.8 Corollary. Suppose that $P \in \mathcal{L}$ and $\mathcal{T}$ is an unfactored weak deduction tree for P in T with $T \not\models \bigvee \mathcal{O}(\mathcal{T})$, say $M \models T \wedge \neg \bigvee \mathcal{O}(\mathcal{T})$. If $M' \subseteq M$ is a minimal model of T, then $P \in M'$.

Thus the existence of such a tree witnesses that P belongs to some minimal model of T, and hence we make the following definition.

5.1.9 Definition. Let $P \in \mathcal{L}$, then an unfactored weak deduction tree $\mathcal{T}$ for P in T *witnesses* P iff $T \not\models \bigvee \mathcal{O}(\mathcal{T})$.

In the case of cyclic trees, we have the following stronger result.

5.1.10 Theorem. Suppose that $\mathcal{T}$ is an unfactored cyclic tree in T. If $M \models T \wedge \neg \bigvee \mathcal{O}(\mathcal{T})$, and $M' \subseteq M$ is a minimal model of T, then $Pred(\mathcal{T}) \subseteq M'$.

Proof. Suppose that N is a predicate node in $\mathcal{T}$ such that $lab(N) \notin M'$. Clearly we may assume that for each predicate node $N' > N$, $lab(N') \in M'$. ....(*).

As in the proof of Theorem 5.1.7, we may construct an (infinite) branch $N = N_0 > \mathrm{RN}_{C_0} > N_1 > \mathrm{RN}_{C_1} > N_2 > \ldots$ through $\mathcal{T}$ such that for each $i \geq 0$:

(i) $lab(N_i) \notin M'$, and

(ii) N_i is the parent of RN_{C_i} which in turn is the parent of N_{i+1}.

Since the tree is cyclic,

$$\mathrm{conseq}(C_j) = \mathcal{O}(RN_{C_j}) \cup (\mathrm{conseq}(C_j) - \mathcal{O}(RN_{C_j})) \subseteq \mathcal{O}(\mathcal{T}) \cup \mathrm{CYC}(N_j),$$

and by conditions (i) and (*), $\mathrm{CYC}(N_j) \subseteq \{lab(N_i) \mid i \leq j\} \subseteq \mathcal{L} - M'$. Since $M \cap \mathcal{O}(\mathcal{T}) = \emptyset$, we have that $\mathrm{conseq}(C_j) \cap M' = \emptyset$. Hence, as in the proof of Theorem 5.1.7, we may find a child node N_{j+1} of RN_{C_j} with $lab(N_{j+1}) \notin M'$. ∎

Theorem 5.1.10 can be rephrased in the manner of Theorem 5.1.7, i.e., $T \models Q \vee \bigvee \mathcal{O}(\mathcal{T})$ for each $Q \in Pred(\mathcal{T})$. Converse results are given in Theorems 5.1.13 and 5.1.14.

As regards the problem of deciding upon the truth or falsity of $T \models \bigvee \mathcal{O}(\mathcal{T})$, appropriate methods are given in [10] (using the notion of a deduction tree) and [11] (using the related notion of a cover).

5.1.11 Definition [11]. Let $\mathcal{Q} \subseteq \mathcal{L}$, then a *cover* of $\mathcal{Q}$ in $\mathrm{INT}(T)$ is a set $\mathcal{C}$ such that $\mathcal{Q} \subseteq \mathcal{C} \subseteq \mathcal{L}$ and

$$\forall C \in \mathrm{INT}(T)\, (\mathrm{conseq}(C) \subseteq \mathcal{C} \implies \mathrm{antec}(C) \cap \mathcal{C} \neq \emptyset).$$

5.1.12 Theorem. Let $\mathcal{T}$ be an unfactored cyclic tree in T, then $T \not\models \bigvee \mathcal{O}(\mathcal{T})$ iff there is a cover $\mathcal{C}$ of $\mathcal{O}(\mathcal{T})$ in $\mathrm{INT}(T)$ such that $\mathcal{C} \cap Pred(\mathcal{T}) = \emptyset$ and $\mathrm{EXT}(T) \not\models \bigvee \mathcal{C}$.

Proof ($\rightarrow$). Suppose that $M \models T$ with $M \cap \mathcal{O}(\mathcal{T}) = \emptyset$, then $\mathcal{C} = \mathcal{L} - M$ is a cover of $\mathcal{O}(\mathcal{T})$ in $\mathrm{INT}(T)$ such that $M \cap \mathcal{C} = \emptyset$. By Theorem 5.1.10, $Pred(\mathcal{T}) \subseteq M$ and clearly $M \models \mathrm{EXT}(T) \wedge \neg \bigvee \mathcal{C}$.

($\leftarrow$). Let $\mathcal{C}$ be a cover of $\mathcal{O}(\mathcal{T})$ in INT(T) such that EXT(T) $\not\models \bigvee \mathcal{C}$. Then $\mathcal{L} - \mathcal{C} \models T \wedge \neg \bigvee \mathcal{O}(\mathcal{T})$. ∎

The condition restricting covers to those disjoint from $\mathit{Pred}(\mathcal{T})$ can be seen as a means of simplifying (or pruning) the search for a model of $T \wedge \neg \bigvee \mathcal{O}(\mathcal{T})$.

5.1.13 Theorem. If P belongs to some minimal model of T, then there is an unfactored weak deduction tree $\mathcal{T}$ for P in T such that:

(a) $\mathcal{T}$ witnesses P, and

(b) If N and N' are predicate nodes in $\mathcal{T}$ with $N > N'$, then $\mathit{lab}(N) \neq \mathit{lab}(N')$.

Thus in particular, each branch has $\leq |\mathcal{L}|$ predicate nodes. The proof of the above theorem is similar to that of Theorem 5.1.14 below.

5.1.14 Theorem. If P belongs to some minimal model of T, then there is an unfactored cyclic tree for P in T such that:

(a) $\mathcal{T}$ witnesses P, and

(b) the number of predicate nodes along each branch does not exceed $|\mathcal{L}| * (|\mathcal{L}| + 1)/2$.

Proof. Let M be a minimal model of T containing P. We construct a witnessing tree $\mathcal{T}$ such that $\mathit{Pred}(\mathcal{T}) \subseteq M$ and $\mathcal{O}(\mathcal{T}) \subseteq \mathcal{L} - M$. Initially we set $\mathcal{T}$ to consist of the single predicate node N_P.

Suppose that we have a predicate leaf node N, then CYC(N) $\subseteq \mathit{Pred}(\mathcal{T}) \subseteq M$ and CYC(N) $\neq \emptyset$. By the minimality of M, $M -$ CYC(N) $\not\models T$, and thus we may find a rule $C \in T$ such that antec(C) $\subseteq M -$ CYC(N) and conseq(C) $\cap$ (M $-$ CYC(N)) $= \emptyset$. (i.e., C witnesses CYC(N) in M.) Since $M \models T$, conseq(C) $\cap$ CYC(N) $\neq \emptyset$, and since ACT(N) $\subseteq M$, we have that

$$\text{conseq}(C) \cap \text{ACT}(N) \subseteq \text{conseq}(C) \cap (\text{CYC}(N) \cup (M - \text{CYC}(N))) \subseteq \text{CYC}(N).$$

Also conseq(C) $-$ ACT(N) $\subseteq$ conseq(C) $-$ CYC(N) $\subseteq \mathcal{L} - M \subseteq \mathcal{L} - \mathit{Pred}(\mathcal{T})$. Hence we may append the rule node RN_C as the child of N.

Finally we need to show that the above construction yields a finite tree. The crucial condition is that whenever RN_C has parent N, then for each child node N' of RN_C we have $\mathit{lab}(N') \notin$ CYC(N).

Suppose that $root(\mathcal{T}) = N_1 > RN_{C_1} > N_2 > RN_{C_2} > \ldots > N_i > RN_{C_i} > \ldots$ is a branch through $\mathcal{T}$, and let $i_1 < i_2 < \ldots < i_r$ be such that for each i, $lab(N_i) \notin \{lab(N_k) \mid k < i\}$ iff $i \in \{i_j \mid j \leq r\}$. Thus the set $\{i_j \mid j \leq r\}$ indicates where new predicates are added to the branch. Clearly $i_1 = 1$ and $r \leq |\mathcal{L}|$.

Pick $j < r$, then we claim that $lab(N_{i_j})$, $lab(N_{i_j+1})$, ..., $lab(N_{i_{j+1}-1})$ are all distinct. Suppose not, and

$$l_0 = \min\{l \mid i_j < l \leq i_{j+1} - 1, lab(N_l) \in \{lab(N_k) \mid i_j \leq k < l\}\}$$

then $l_0 > i_j$. By the definition of i_j and i_{j+1}, $top(N_{l_0-1}) \geq N_{i_j}$ and thus $lab(N_{l_0}) \in \mathrm{CYC}(N_{l_0-1})$; a contradiction. This then proves that $i_{j+1} \leq i_j + j$.

Similarly we may show that the predicates in $lab(N_{i_r})$, $lab(N_{i_r+1})$, $lab(N_{i_r+2})$, $\ldots$ are all distinct, and thus that the branch has length $\leq i_r + r - 1$.

By induction, $i_r \leq 1 + 1 + 2 + \ldots + (r-1)$, and thus the branch has length $\leq (1 + 1 + 2 + 3 + 4 + ... + (r-1)) + r - 1 \leq |\mathcal{L}| * (|\mathcal{L}| + 1)/2$. ∎

Notice that if $\mathcal{T}$ is a cyclic tree for a non-recursive database, and $N > N'$, then $\ell(lab(N)) > \ell(lab(N'))$, thus each branch must have length $\leq |\{\ell(P) \mid P \in \mathcal{L}\}|$ (i.e. less than the number of levels in the stratification). This again suggests that computing the GCWA is easier in the non-recursive case (cf. Section 2.1).

The property guaranteeing termination in the above proof will prove to be important in our later discussions, and we thus make the following definitions.

5.1.15 Definition. Let $\mathcal{T}$ be a weak deduction tree and $N > RN > N'$ where N is the parent of RN which in turn is the parent of N'. Then N' is:

(a) *redundant* iff $lab(N') \in \mathrm{ACT}(N)$, i.e., $\exists N'' > N'\ (lab(N'') = lab(N'))$, and

(b) *strongly redundant* iff $lab(N') \in \mathrm{CYC}(N)$.

5.2 Testing models for minimality.

The results of the preceding section allow us to deduce the following theorem, which in turn provides a method of testing models for minimality.

5.2.1 Theorem. Let M be a model of T, then M is minimal iff there is a sequence $(\mathcal{T}_i \mid 0 \leq i \leq n)$ of cyclic trees in T such that:

(a) $\bigcup_{i=0}^{n} Pred(\mathcal{T}_i) = M \subseteq \mathcal{L} - \bigcup_{i=0}^{n} \mathcal{O}(\mathcal{T}_i)$, and

(b) if N is a predicate leaf node in $\mathcal{T}_i$, then $lab(N) \in \bigcup_{j<i} Pred(\mathcal{T}_j)$.

Proof $(\rightarrow)$. If M is minimal and $P \in M$, then by Theorem 5.1.14, there is an unfactored cyclic tree $\mathcal{T}_P$ for P in T such that $Pred(\mathcal{T}_P) \subseteq M \subseteq \mathcal{L} - \mathcal{O}(\mathcal{T}_P)$.

$(\leftarrow)$. Let $M' \subset M$ be a minimal model of T, and let $i_0 = \min\{i \leq n \mid Pred(\mathcal{T}_i) \not\subseteq M'\}$.

Pick a predicate node N in $\mathcal{T}_{i_0}$ such that $lab(N) \notin M'$ and for each $N' > N$, $lab(N') \in M'$. We construct a sequence $N = N_0 > \mathrm{RN}_{C_0} > N_1 > \mathrm{RN}_{C_1} > N_2 > \ldots$ through $\mathcal{T}_{i_0}$ such that for each $k \geq 0$:

(i) $lab(N_k) \notin M'$, and

(ii) N_k is the parent of RN_{C_k} which in turn is the parent of N_{k+1}.

Note that each such N_k cannot be a leaf, since otherwise $lab(N_k) \in Pred(\mathcal{T}_j) - M'$ for some $j < i_0$, thus contradicting the minimality of i_0. Hence we may find RN_{C_k} and N_{k+1} as in the proof of Theorem 5.1.10. This then contradicts the finiteness of $\mathcal{T}_{i_0}$. ∎

Note that condition (b) represents a pruning of the tree $\mathcal{T}_i$. Another method of pruning is presented in Section 5.4 below. Thus to test for minimality we need to build the cyclic trees given in the above theorem. The definition below identifies the basic mechanism for constructing weak deduction trees.

5.2.2 Definition. Let $\mathcal{T}$ be a weak deduction tree in $\mathcal{L}$, N a predicate leaf node in $\mathcal{T}$ and C a rule in T such that:

(i) $\mathrm{conseq}(C) \cap \mathrm{ACT}(N) \neq \emptyset$, and

(ii) $Pred(\mathcal{T}) \cup \mathrm{antec}(C)$ and $\mathcal{O}(\mathcal{T}) \cup (\mathrm{conseq}(C) - \mathrm{ACT}(N))$ are disjoint.

Then EXTEND($\mathcal{T}$, N, C) is the tree formed by appending the node RN_C to T as a child of N and appending the antecedants of C as child nodes of RN_C.

Notice that $Pred(\mathrm{EXTEND}(\mathcal{T}, N, C)) = Pred(\mathcal{T}) \cup \mathrm{antec}(C)$, and $\mathcal{O}(\mathrm{EXTEND}(\mathcal{T}, N, C)) = \mathcal{O}(\mathcal{T}) \cup (\mathrm{conseq}(C) - \mathrm{ACT}(N))$, hence the reason for insisting that these two sets be disjoint.

Note also that if $\mathcal{T}$ is cyclic and $\mathrm{conseq}(C) \cap \mathrm{ACT}(N) \subseteq \mathrm{CYC}(N)$ then EXTEND($\mathcal{T}$,

N, C) is cyclic.

5.2.3. The following algorithm attempts to construct a sequence of trees satisfying the conditions of Theorem 5.2.1. At each stage we attempt to construct the next cyclic tree $\mathcal{T}_i$. $\mathcal{P}$ will represent $\bigcup_{j<i} Pred(\mathcal{T}_j)$, and we thus need to extend $\mathcal{T}_i$ until $\{lab(N) \mid N \text{ is a leaf in } \mathcal{T}_i\} \subseteq \mathcal{P}$. If at some stage, no rule can be found to extend a predicate leaf node N, then we may conclude that CYC(N) is not witnessed, and hence that $M -$ CYC(N) is a model, i.e., M is not minimal. We leave the proof of correctness to the reader.

```
MIN (T, M)
{    P = ∅;
     while P ⊂ M
          {    pick P ∈ M − P;
               T = {N_P};
               while {lab(N) | N is a predicate leaf node in T} ⊈ P
                    {    pick a predicate leaf node N ∈ T (lab(N) ∉ P);
                         pick C ∈ T (antec(C) ⊆ M − CYC(N) and
                                        conseq(C) ∩ (M − CYC(N)) = ∅);
                         if no such C exists {return(false)};
                         T = EXTEND(T, N, C)
                    };
               P = P ∪ Pred(T)
          };
     return(true)
}
```

5.3 Computing minimal model membership.

5.3.1. As in Section 4, we may use the results of Section 5.1 to compute minimal model membership. The following algorithm generates unfactored cyclic trees $\mathcal{T}$ for P in T, and then tests whether $T \not\models \bigvee \mathcal{O}(\mathcal{T})$.

MINMOD (T, $\mathcal{T}$)
{ *if* $\mathcal{T}$ is unfactored {*return*($T \not\models \bigvee \mathcal{O}(\mathcal{T})$)};
pick a predicate leaf node N in $\mathcal{T}$;
for each $C \in T$
{*if* ($\emptyset \neq$ conseq(C) $\cap$ ACT(N) $\subseteq$ CYC(N) *and*
$Pred(\mathcal{T}) \cup$ antec(C) *and* $\mathcal{O}(\mathcal{T}) \cup$ (conseq(C) $-$ ACT(N)) are disjoint *and*
antec(C) $\cap$ CYC(N) $= \emptyset$)
{ *if* MINMOD (T, EXTEND($\mathcal{T}, N, C$)) {*return*(*true*)}}
};
return(*false*)
}

The algorithm presented above suggests that at each stage we need to maintain all of the tree which has been constructed to-date. This simplifies the presentation of the algorithm, but would obviously be computationally expensive. At each stage the information that is really needed is the current branch, information about branches yet to be examined and the current values for $Pred(\mathcal{T})$ and $\mathcal{O}(\mathcal{T})$. Methods such as those found in [10] can be used to construct (traverse) the tree branch by branch in a depth first, left - to - right fashion using two stack mechanisms and a variant of Bibel's connection method [8].

5.3.2 Theorem (Termination). For any $\mathcal{T}$, MINMOD (T, $\mathcal{T}$) terminates.

Proof. As in the proof of Theorem 5.1.14, the finiteness of each branch is guaranteed by the fact that no predicate node is strongly redundant (cf. Definition 5.1.15). ∎

5.3.3 Theorem (Correctness). Let $P \in \mathcal{L}$, and $\mathcal{T} = \{N_P\}$. Then P belongs to some minimal model of T iff MINMOD (T, $\mathcal{T}$) returns true.

5.3.4 Parallel execution. As mentioned earlier, the methods of [10] can be employed to check whether $T \models \bigvee \mathcal{O}(\mathcal{T})$ using the notion of a deduction tree. Specifically $T \models \bigvee \mathcal{O}(\mathcal{T})$ iff there is an (unfactored) deduction tree for $\mathcal{O}(\mathcal{T})$ in T.

Moreover this method has the desirable property that if $T \not\models \bigvee \mathcal{O}_1$, $\mathcal{D}_1$ is a partial deduction tree for $\mathcal{O}_1$, and $T \models \bigvee \mathcal{O}_2$ where $\mathcal{O}_2 \supseteq \mathcal{O}_1$, then $\mathcal{D}_1$ may be extended to an unfactored deduction tree for $\mathcal{O}_2$.

Notice also that in our construction, the set $\mathcal{O}(\mathcal{T})$ grows monotonically as $\mathcal{T}$ is constructed. It is thus possible to check the status of $T \models \bigvee \mathcal{O}(\mathcal{T})$ in parallel with construction of $\mathcal{T}$.

At any given stage we maintain a partial deduction tree $\mathcal{D}$ for $\mathcal{O}(\mathcal{T})$ in T: as $\mathcal{O}(\mathcal{T})$ grows, so does $\mathcal{D}$. If $\mathcal{D}$ becomes unfactored, then $T \models \bigvee \mathcal{O}(\mathcal{T})$, and we can abort our tree construction since the tree being generated cannot be a witness tree.

5.4 Cyclic vs. non-cyclic trees.

The minimal model membership problem can thus be solved by attempting to construct an unfactored weak deduction tree. This raises the question as to whether it is preferable (from a computational viewpoint) to construct a cyclic or a non-cyclic tree.

When constructing a non-cyclic tree, Theorem 5.1.13 indicates that we may extend a predicate leaf node N via any rule C such that:

(i) $\emptyset \neq \text{conseq}(C) \cap \text{ACT}(N)$,

(ii) $\mathcal{P}red(\mathcal{T}) \cap\ (\text{conseq}(C) - \text{ACT}(N)) = \emptyset$, and

(iii) $\text{antec}(C) \cap (\mathcal{O}(\mathcal{T}) \cup \text{ACT}(N)) = \emptyset$.

Thus N is extended by a rule which will witness $\text{ACT}(N)$, but which will not introduce any redundant predicate node (cf. Definition 5.1.15). When constructing a cyclic tree the corresponding conditions are aimed at witnessing $\text{CYC}(N)$, and are:

(iv) $\emptyset \neq \text{conseq}(C) \cap \text{ACT}(N) \subseteq \text{CYC}(N)$,

(v) $\mathcal{P}red(\mathcal{T}) \cap\ (\text{conseq}(C) - \text{ACT}(N)) = \emptyset$, and

(vi) $\text{antec}(C) \cap (\mathcal{O}(\mathcal{T}) \cup \text{CYC}(N)) = \emptyset$.

Clearly condition (iv) provides a stronger constraint than does (i) (this being a good thing, since it limits the search space), whereas condition (vi) is weaker than (iii). As indicated in Theorems 5.1.13 and 5.1.14, a result of this is that whilst constructing a cyclic tree we may be forced to admit pairs of nodes $N > N'$ where $lab(N) = lab(N')$, thus resulting in redundant predicate nodes and hence longer branches. We can however show that such redundant nodes do not need to induce further branching, and thus in particular do not increase the cost of our tree search. Specifically we will show that if a node N' is redundant, then its sibling nodes need not be extended (i.e., can be ignored). Note however that if two siblings are redundant, then we cannot ignore them both by virtue of the redundancy of the other. Condition (b) of Definition 5.4.2 below arbitrarily resolves this issue.

5.4.1 Definition. Let $\mathcal{T}$ be a weak deduction tree.

(a) If N is a predicate node in $\mathcal{T}$, then

$$\text{CYC}^*(N) = \{lab(N') \mid N' \text{ is a non redundant predicate node, } top(N) \geq N' \geq N\}.$$

(b) $\mathcal{O}^*(\mathcal{T}) = \bigcup\{\mathcal{O}(RN) \mid RN \text{ has no redundant child }\}$.

5.4.2 Definition. A cyclic tree $\mathcal{T}$ is said be a *pruned* cyclic tree iff:

(a) every predicate leaf node has a non-leaf redundant sibling,

(b) every redundant leaf node has a non-leaf redundant right sibling, and

(c) whenever RN_C is a rule node with parent N, then $\text{conseq}(C) \cap \text{ACT}(N) \subseteq \text{CYC}^*(N)$.

When generating cyclic trees, we extended a predicate leaf node N via a rule node RN_C such that $\text{antec}(C) \cap \text{CYC}(N) = \emptyset$.

In the terminology of Definition 5.1.15, no predicate node was allowed to be strongly redundant. This in turn guaranteed the finiteness of the construction. The following lemma and theorem show that a corresponding result holds for pruned trees.

5.4.3 Lemma. Let $\mathcal{T}$ be a cyclic tree in which $N > RN_C > N'$, where N is the parent of RN_C which in turn is the parent of N'. If N' is redundant and $lab(N') \notin \text{CYC}^*(N)$, then $top(N') > top(N)$.

Proof. Suppose that $top(N') \leq top(N)$. Since N' is redundant, $top(N') \geq N$. However, $top(N')$ is itself not redundant, and hence $lab(N') = lab(top(N')) \in \text{CYC}^*(N)$. ∎

As in Section 5.1, minimal model membership may be characterised by the existence of a witnessing pruned cyclic tree. The following are analogous to Theorems 5.1.14 and 5.1.7 respectively.

5.4.4 Theorem. If P belongs to some minimal model of T, then there is a pruned cyclic tree $\mathcal{T}$ for P in T such that:

(i) $T \not\models \bigvee \mathcal{O}(\mathcal{T})$ (i.e., $\mathcal{T}$ witnesses P), and

(ii) any branch through $\mathcal{T}$ has $\leq |\mathcal{L}| * (|\mathcal{L}| + 1)/2$ predicate nodes.

Proof. Let M be a minimal model of T containing P. We construct a reduced pruned tree $\mathcal{T}$ such that $Pred(\mathcal{T}) \subseteq M \subseteq \mathcal{L} - \mathcal{O}(\mathcal{T})$. As in the proof of Theorem 5.1.14, we extend predicate nodes N (which cannot be leaf nodes by the conditions of Definition 5.4.2) by a rule node RN_C such that $\text{antec}(C) \subseteq M - \text{CYC}^*(N)$ and $\text{conseq}(C) \cap (M - \text{CYC}^*(N)) = \emptyset$.

Suppose that $root(\mathcal{T}) = N_1 > RN_{C_1} > N_2 > RN_{C_2} > \ldots > N_i > RN_{C_i} > \ldots$ is a branch through $\mathcal{T}$, and (as in the proof of Theorem 5.1.14) let $i_1 < i_2 < \ldots < i_r$ be such that for each i, $lab(N_i) \notin \{lab(N_k) \mid k < i\}$ iff $i \in \{i_j \mid j \leq r\}$. Suppose that N_{i+1} is redundant, then by the preceding lemma, $top(N_{i+1}) > top(N_i)$. Thus, as in the proof of Theorem 5.1.14, for each $j < r$, $i_{j+1} \leq i_j + j$, and the branch has length $\leq i_r + r - 1$. ▮

5.4.5 Theorem. If $\mathcal{T}$ is a pruned cyclic tree for P in T, then $T \models P \vee \bigvee \mathcal{O}^*(\mathcal{T})$.

Proof. Let $M \models T \wedge \neg \bigvee \mathcal{O}^*(\mathcal{T})$ and suppose that $P \notin M$. As in the proof of Theorem 5.1.7, we may construct an (infinite) branch $root(\mathcal{T}) = N_0 > \text{RN}_{C_0} > N_1 > \text{RN}_{C_1} > N_2 > \ldots$ through $\mathcal{T}$ such that for each $i \geq 0$:

(i) $lab(N_i) \notin M$,

(ii) N_i is not a leaf, and

(iii) N_i is the parent of RN_{C_i} which in turn is the parent of N_{i+1}.

Suppose we are given N_i ($i \leq j$). By condition (ii), RN_{C_j} exists.

If RN_{C_j} has a redundant child, then it has a non-leaf redundant child N''. Moreover, by condition (i) we must have that $lab(N'') \notin M$, and thus we may set $N_{j+1} = N''$.

If RN_{C_j} has no redundant child then $\text{conseq}(C_j) \subseteq \mathcal{O}(RN_{C_j}) \cup (\text{conseq}(C_j) \cap \text{ACT}(N_j)) \subseteq \mathcal{O}^*(\mathcal{T}) \cup \text{ACT}(N_j) \subseteq \mathcal{L} - M$ (by (i) and the choice of M), and thus there is some child N'' of RN_{C_j} such that $lab(N'') \notin M'$. Since RN_{C_j} has no redundant child, N'' cannot be a leaf, and thus we may set $N_{j+1} = N''$. ▮

Pruned cyclic trees can be generated by a modified version of the algorithm presented in Section 5.3.1.

The reader will note that a weak deduction tree $\mathcal{T}$ for P actually forms a deduction tree ([10]) for $\{P\} \cup \mathcal{O}(\mathcal{T})$. This then enables us to employ other pruning techniques (in particular factorisation) discussed in [10], including similar pruning of $\mathcal{O}(\mathcal{T})$ to a smaller $\mathcal{O}^*(\mathcal{T})$ as seen in the above theorem.

The above results and remarks indicate that cyclic trees appear to be preferable to non-cyclic.

The following example shows that Theorem 5.4.5 cannot be extended to prove that $Pred(\mathcal{T}) \subseteq M$.

5.4.6 Example. Let T contain the rules:

1. $Q \rightarrow P$ 2. $P \wedge R \rightarrow Q$ 3. $P \vee Q$

then a pruned tree for P in T is depicted in Figure 5.4.6, yet the only minimal model of T is $\{P\}$.

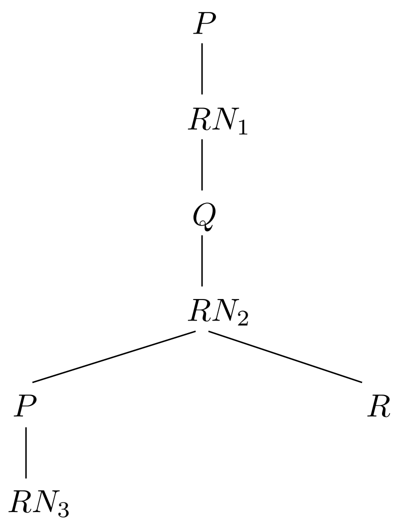

Figure 5.4.6.

5.5 Further pruning.

The removal of redundant nodes from cycles has the effect of reducing the size of the cycle, and as indicated in the previous section, this is advantageous since it limits the search space. We can extend this process, further reducing the size of the cycle as follows.

5.5.1 Definition. Let $\mathcal{T}$ be a weak deduction tree and N a predicate node in $\mathcal{T}$.

(i) Let $top^*(N)$ denote the unique node such that $\exists N' \geq N\,(top^*(N) = top(N'))$ and $\forall N'' \geq N\,(top(N'') \geq top^*(N))$.

(ii) $\text{CYC}^+(N) = \{lab(M) \mid top^*(N) \geq M \geq N, top(N) \geq top(M)\}$.

5.5.2 Lemma. Let $\mathcal{T}$ be a weak deduction tree and N a predicate node in $\mathcal{T}$. Then:

(a) $top^*(N)$ is the unique non-redundant node, $top^*(N) \geq N$ such that each node in $\{N' \mid top^*(N) > N' \geq N\}$ is redundant.

(b) $N \leq top^*(N) \leq top(N)$.

(c) $\text{CYC}^+(N) \subseteq \text{CYC}^*(N)$ (cf. Definition 5.4.1).

Proof. Parts (a) and (b) are trivial.

For part (c), suppose that $top^*(N) \geq M \geq N$ and $top(N) \geq top(M)$. Thus $top(N) \geq top(M) \geq N$, and since $top(M)$ is not redundant, $lab(M) = lab(top(M)) \in \text{CYC}^*(N)$. ∎

As mentioned above, the intention is that $\text{CYC}^+(N)$ should be smaller than $\text{CYC}^*(N)$, and indeed it is easy to construct examples in which $\text{CYC}^+(N) \subset \text{CYC}^*(N)$. The following proposition shows that top^* is easily computed.

5.5.3 Proposition. Let $\mathcal{T}$ be a weak deduction tree, and $N > RN > N'$ be nodes in $\mathcal{T}$ where N is the parent of RN which in turn is the parent of N'. Then

$$top^*(N') = \begin{cases} top^*(N), & \text{if } N' \text{ is redundant;} \\ N', & \text{otherwise.} \end{cases}$$

5.5.4 Definition. A pruned tree $\mathcal{T}$ is said to be a *reduced* pruned tree iff whenever RN_C is a rule node with parent N, then $\text{conseq}(C) \cap \text{ACT}(N) \subseteq \text{CYC}^+(N)$.

5.5.5 Theorem. If P belongs to some minimal model of T, then there is a reduced pruned tree $\mathcal{T}$ for P in T such that $T \not\models \bigvee \mathcal{O}(\mathcal{T})$.

The construction proceeds as in the proof of Theorem 5.4.4 employing CYC$^+$ rather than CYC*. The interested reader may show (in a similar fashion to that seen in Theorem 5.1.14) that the construction yields a tree in which each branch has at most $|\mathcal{L}|^3$ predicate nodes.

5.6 Compiling the GCWA.

In research into deductive databases it is usual to categorise predicates as extensional or intensional. Thus in this section we assume that $\mathcal{L}$ is the disjoint union of $\text{EXT}(\mathcal{L})$ (the *extensional* predicates) and $\text{INT}(\mathcal{L})$ (the *intensional* predicates) such that:

(a) if $C \in \text{EXT}(T)$ then $\text{conseq}(C) \subseteq \text{EXT}(\mathcal{L})$, and

(b) if $C \in \text{INT}(T)$ then $\text{conseq}(C) \subseteq \text{INT}(\mathcal{L})$.

Recall that $\text{EXT}(T) = \{C \in T \mid \text{antec}(C) = \emptyset\}$, and that $\text{INT}(T) = T - \text{EXT}(T)$. Notice also that for $P \in \text{EXT}(\mathcal{L})$, P belongs to some minimal model of T iff P belongs to some minimal model of $\text{EXT}(T)$, thus we may concentrate on the problem of intensional predicates.

5.6.1 Proposition.

(a) If $\mathcal{T}$ is a weak deduction tree in $\text{INT}(T)$, then each leaf node is a predicate node.

(b) If $\mathcal{T}$ is a cyclic tree in T and $lab(N) \in \text{EXT}(\mathcal{L})$, then either N is a leaf, or has a child node RN_C which is labelled with a rule $C \in \text{EXT}(T)$ (and hence is itself a leaf).

(c) If $\mathcal{T}$ is a cyclic tree in $\text{INT}(T)$, then for each predicate node N ($lab(N) \in \text{EXT}(\mathcal{L}) \Longrightarrow N$ is a leaf), and $\mathcal{O}(\mathcal{T}) \subseteq \text{INT}(\mathcal{L})$.

Proof (b). We proceed by induction. Suppose that $lab(N) \in \text{EXT}(\mathcal{L})$ and the result holds for all predicate nodes $N' > N$. Note that this implies that $\forall N' > N(lab(N') \in \text{INT}(\mathcal{L}))$, and hence that $\text{CYC}(N) = \{lab(N)\}$.

Suppose that $C \in \text{INT}(T)$, then $\emptyset \neq \text{conseq}(C) \cap \text{ACT}(N) \subseteq \text{INT}(\mathcal{L}) \cap \text{CYC}(N)$, a contradiction. ∎

5.6.2 Theorem. Suppose that $P \in \mathrm{INT}(\mathcal{L})$ and $\mathcal{T}$ is a cyclic tree for P in $\mathrm{INT}(T)$ such that each leaf node N is a predicate node with $lab(N) \in \mathrm{EXT}(\mathcal{L})$. If $\mathrm{EXT}(T)$ has a minimal model M containing $\{lab(N) \mid N \text{ is a leaf in } \mathcal{T}\}$ such that $\mathrm{INT}(T) \not\models \bigwedge M \to \bigvee \mathcal{O}(\mathcal{T})$, then P belongs to some minimal model of T.

Proof. Suppose that M' is a model of $\mathrm{INT}(T)$ such that $M' \supseteq M$ and $M' \cap \mathcal{O}(\mathcal{T}) = \emptyset$. Quite clearly $M \cup (M' \cap \mathrm{INT}(\mathcal{L})) \models T$.

Pick $M'' \subseteq M \cup (M' \cap \mathrm{INT}(\mathcal{L}))$ such that M'' is a minimal model of T, then $M'' \cap \mathrm{EXT}(\mathcal{L})$ is a model of $\mathrm{EXT}(T)$ and thus by the minimality of M we have $M'' \cap \mathrm{EXT}(\mathcal{L}) = M$. We show that $Pred(\mathcal{T}) \subseteq M''$.

As in the proof of Theorem 5.1.10, if N if a predicate node such that $lab(N) \notin M''$ and $\forall N'' > N(lab(N'') \in M'')$, then we may find a branch $N = N_0 > RN_{C_0} > N_1 > RN_{C_1} > \ldots > N_i > RN_{C_i} > \ldots$ through N such that for each i, $lab(N_i) \notin M''$. But then the branch terminates in some leaf node N_j, whence $lab(N_j) \in M \subseteq M''$; a contradiction. ∎

Similarly we can prove the converse.

5.6.3 Theorem. Suppose that P belongs to some minimal model of T, then there is a cyclic tree $\mathcal{T}$ for P in $\mathrm{INT}(T)$ such that:

(a) each leaf node N is a predicate node with $lab(N) \in \mathrm{EXT}(\mathcal{L})$, and

(b) $\mathrm{EXT}(T)$ has a minimal model M containing $\{lab(N) \mid N \text{ is a leaf in } \mathcal{T}\}$ such that $\mathrm{INT}(T) \not\models \bigwedge M \to \bigvee \mathcal{O}(\mathcal{T})$.

Proof. Let M' be a minimal model of T containing P, then by Theorem 5.1.14 there is an unfactored cyclic tree $\mathcal{T}'$ for P in T such that $Pred(\mathcal{T}') \subseteq M' \subseteq \mathcal{L} - \mathcal{O}(\mathcal{T}')$.

Let $\mathcal{T}$ be the tree formed by removing all leaf nodes from $\mathcal{T}$. Note that if RN_C is a leaf in $\mathcal{T}'$ with parent N, then C is extensional and $\mathrm{EXT}(\mathcal{L}) \cap \mathrm{ACT}(N) \supseteq \mathrm{conseq}(C) \cap \mathrm{ACT}(N) \neq \emptyset$. However, by Proposition 5.6.1, the only place in which an extensional predicate can appear on $\mathrm{ACT}(N)$ is at N itself. Thus $\mathcal{T}$ satisfies condition (a).

Let $M = M' \cap \mathrm{EXT}(\mathcal{L})$, then M satisfies condition (b). ∎

Thus the above theorems suggests that to compute the GCWA we should:

(i) compute cyclic trees $\mathcal{T}$ in INT(T) satisfying condition (a) of Theorem 5.6.3,

(ii) compute minimal models M of EXT(T) containing $\{lab(N) \mid N \text{ is a leaf in } \mathcal{T}\}$, and then

(iii) check whether INT(T) $\models \bigwedge M \rightarrow \bigvee \mathcal{O}(\mathcal{T})$.

Methods for handling step (iii) are discussed in [10] and [11]. Computing the EGCWA for EXT(T) (step (ii)) can be handled via the top down method given in [9, Section 5]. The following theorem indicates that steps (ii) and (iii) can be handled in the converse order, again illustrating the idea (cf. Section 5.3.4) of interleaving the construction of $\mathcal{T}$ with the processing of $\mathcal{O}(\mathcal{T})$.

5.6.4 Theorem. P belongs to some minimal model of T iff there is a cyclic tree, $\mathcal{T}$ for P in INT(T) such that:

(a) each leaf node is a predicate node with $lab(N) \in$ EXT($\mathcal{L}$),

(b) each predicate node is not strongly redundant, and

(c) $\mathcal{O}(\mathcal{T})$ has a cover $\mathcal{C}$ in INT(T) such that (i) $\mathcal{C} \cap Pred(\mathcal{T}) = \emptyset$, and (ii) MM(EXT($T$), $\{lab(N) \mid N \text{ is a leaf in } \mathcal{T}\}$, $\mathcal{C} \cap$ EXT($\mathcal{L}$)) (cf. Definition 4.1.1).

Note that condition (ii) is included merely to simplify Example 5.6.5 below.

Proof ($\rightarrow$). Let M be a minimal model of T containing P, then as in the proof of the previous theorem, we may find a cyclic tree in INT(T) satisfying conditions (a) and (b) such that $Pred(\mathcal{T}) \subseteq M \subseteq \mathcal{L} - \mathcal{O}(\mathcal{T})$.

But then $\mathcal{C} = \mathcal{L} - M$ is a cover of $\mathcal{O}(\mathcal{T})$ such that $\mathcal{C} \cap Pred(\mathcal{T}) = \emptyset$ and clearly $M \cap$ EXT($\mathcal{L}$) satisfies MM(EXT(T), $\{lab(N) \mid N \text{ is a leaf in } \mathcal{T}\}$, $\mathcal{C} \cap$ EXT($\mathcal{L}$)).

($\leftarrow$). Let $M \subseteq$ EXT($\mathcal{L}$) be a minimal model of EXT(T) satisfying MM(EXT(T), $\{lab(N) \mid N \text{ is a leaf in } \mathcal{T}\}$, $\mathcal{C} \cap$ EXT($\mathcal{L}$)), then $M \cup (\text{INT}(\mathcal{L}) - \mathcal{C}) \models T \wedge \bigwedge M \wedge \neg \bigvee \mathcal{O}(\mathcal{T})$. The result then follows from Theorem 5.6.2. ∎

The usual assumptions made about EXT(T) and INT(T) are that EXT(T) is large and volatile, whereas INT(T) is somewhat smaller and static. This assumption suggests the idea of compiling (pre-processing) INT(T) so that at query processing time we simply need to manipulate the current extension [7]. (Of course a modification to the intension would require re-compilation).

Theorem 5.6.4 gives us a means of compiling the GCWA, since we may (using the methods of Section 5.3.1) generate the $(\{lab(N) \mid N \text{ is a leaf in } \mathcal{T}\}, \mathcal{C} \cap \text{EXT}(\mathcal{L}))$ - pairs satisfying conditions (a), (b) and (c)(i) of Theorem 5.6.4 *without* reference to EXT(T). Processing the query $?\neg P$, then amounts to searching for such a pair satisfying condition (c)(ii) of Theorem 5.6.4, which in particular involves a manipulation of the current extension (only).

The above theorem addresses the problem of deciding MM(T, $\{P\}$, $\emptyset$). Methods for processing (or compiling) MM(T, $\mathcal{P}$, $\mathcal{O}$) are given in Section 7.4.

We may say that a cover $\mathcal{C}$ of $\mathcal{O}(\mathcal{T})$ is *minimal* iff it properly contains no cover of $\mathcal{O}(\mathcal{T})$. Quite clearly the above theorem remains valid if we restrict attention to minimal covers. This restriction then simplifies the following example.

5.6.5 Example. Suppose that $\text{EXT}(\mathcal{L}) = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$, and INT($T$) consists of the following rules:

1. $Q_1 \rightarrow P$	2. $Q_2 \wedge Q_3 \rightarrow P \vee R_1$	3. $P \rightarrow Q_2 \vee R_4$
4. $S_1 \rightarrow P \vee Q_2$	5. $S_2 \rightarrow Q_2 \vee Q_3$	6. $S_2 \rightarrow Q_3 \vee R_2$
7. $S_6 \rightarrow Q_2$	8. $Q_2 \wedge S_4 \rightarrow R_1 \vee R_2 \vee R_4$	9. $Q_2 \wedge S_5 \rightarrow R_1 \vee R_2$

then the only cyclic trees for P in INT(T) satisfying conditions (a) and (b) of Theorem 5.6.4 are depicted in Figures 5.6.5(i) and 5.6.5(ii) with $\mathcal{O}(\mathcal{T}_1) = \mathcal{O}(\mathcal{T}_2) = \{R_1, R_2, R_4\}$, $\mathcal{O}(\mathcal{T}_3) = \{Q_2\}$ and $\mathcal{O}(\mathcal{T}_4) = \{R_1, R_2\}$.

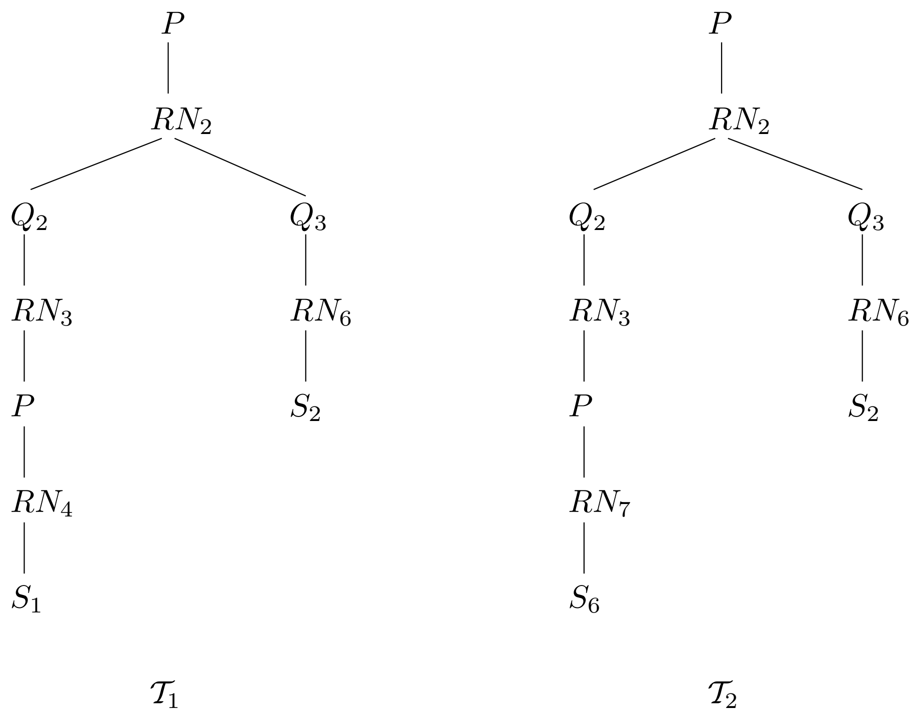

Figure 5.6.5(i).

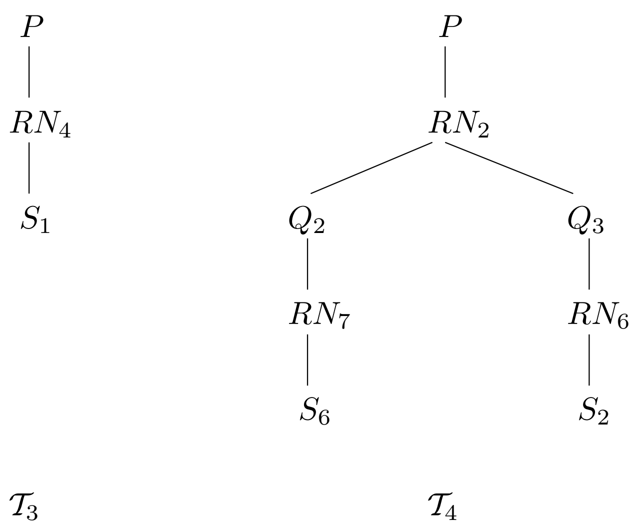

Figure 5.6.5(ii).

Since $Pred(\mathcal{T}_2) = Pred(\mathcal{T}_4)$ and $\mathcal{O}(\mathcal{T}_2) \supseteq \mathcal{O}(\mathcal{T}_4)$, we may ignore $\mathcal{T}_2$ (since a cover satisfying condition (c) of Theorem 5.6.4 for $\mathcal{T}_2$ satisfies the same condition for $\mathcal{T}_4$).

We now look for minimal covers satisfying condition (c)(i) of Theorem 5.6.4. Any cover of $\mathcal{O}(\mathcal{T}_1)$ must (by virtue of rule 8) contain Q_2 or S_4. Since $Q_2 \in \mathit{Pred}(\mathcal{T}_1)$, the only minimal cover of $\mathcal{O}(\mathcal{T}_1)$ satisfying condition (c)(i) is $\{R_1, R_2, R_4, S_4, S_5\}$. By the same argument, the only minimal cover of $\mathcal{O}(\mathcal{T}_4)$ which satisfies (c)(i) is $\{R_1, R_2, S_5\}$. For $\mathcal{O}(\mathcal{T}_3)$, the only minimal cover is $\{Q_2, S_6\}$.

Suppose now that at a given instant, $\mathrm{EXT}(T) = \{S_1, S_2 \vee S_3, S_3 \vee S_4, S_7 \vee S_5, S_6\}$.

Considering $\mathcal{T}_1$, any minimal model of EXT(T) containing $\{S_1, S_2\}$ cannot contain S_3, and thus must contain S_4. Hence MM(EXT(T), $\{S_1, S_2\}$, $\{S_4, S_5\}$) fails.

For $\mathcal{T}_3$, any model of EXT(T) contains S_6, and thus again the MM condition fails.

For $\mathcal{T}_4$, $\{S_1, S_2, S_4, S_6, S_7\}$ is a minimal model of EXT(T), and thus MM(EXT(T), $\{S_2, S_6\}$, $\{S_5\}$) holds. In particular, P belongs to some minimal model of T.

The assumption given at the beginning of this section may be regarded as a (very) weak form of stratification. We can extend this idea as follows.

5.6.6 Definition. T is said to be *partially stratified* iff there is a set $\mathrm{STRAT}(\mathcal{L}) \subseteq \mathcal{L}$ and a function $\ell : \mathrm{STRAT}(\mathcal{L}) \to \mathbb{N}$ such that:

(a) $\mathrm{EXT}(\mathcal{L}) \subseteq \mathrm{STRAT}(\mathcal{L})$,

(b) $\ell(P) = 0$ for each $P \in \mathrm{EXT}(\mathcal{L})$, and

(c) whenever $C \in T$ is such that $\mathrm{conseq}(C) \cap \mathrm{STRAT}(\mathcal{L}) \neq \emptyset$, then $\mathrm{conseq}(C) \cup \mathrm{antec}(C) \subseteq \mathrm{STRAT}(\mathcal{L})$ and $\ell(P) < \ell(Q)$ whenever $P \in \mathrm{antec}(C)$ and $Q \in \mathrm{conseq}(C)$.

Notice that in such a database, $\mathrm{STRAT}(T) = \{C \in T \mid \mathrm{conseq}(C) \cap \mathrm{STRAT}(\mathcal{L}) \neq \emptyset\}$ is non-recursive. Similar results to Theorems 5.6.2 - 5.6.4 can be obtained for partially stratified databases (employing the methods of Section 4 to compute minimal models of STRAT(T)). We leave the details to the reader.

§6. THE FIRST ORDER LEVEL.

In this section we will consider how the results of the previous two sections can be extended to the first order level. In particular, given a first order database T and a positive atom $P(\mathbf{t})$, we wish to be able to test whether some ground instance of $P(\mathbf{t})$ is contained in some minimal model of T.

It is not our purpose here to give an exhaustive treatment, but to illustrate the issues via the consideration of one particular class of databases.

6.1 Terminology.

A first order function free language has the form

$$\mathcal{L} = \{P_1, P_2, \ldots, P_n, c_1, c_2, \ldots, c_m\}$$

where $\{P_1, P_2, \ldots, P_n\}$ is the set of predicate symbols and $\{c_1, c_2, \ldots, c_m\}$ is the set of constant symbols ($n, m > 0$). We implicitly assume the existence of countably many variables $x_1, x_2, \ldots$ for the construction of formulae in $\mathcal{L}$. Again, predicates in $\mathcal{L}$ are either extensional or intensional.

A *term* in $\mathcal{L}$ is a variable or a constant, and a *positive atom* is a formula of the form $P(\mathbf{t})$ where P is a predicate symbol and $\mathbf{t}$ is a sequence of terms of length arity(P). $\mathcal{H}$ (the *Herbrand base*) consists of those positive atoms which are ground, i.e., contain no variables.

6.1.1 Definition. A *rule* in $\mathcal{L}$ is a formula C of the form

$$A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow B_1 \vee B_2 \vee \ldots \vee B_m$$

where:

(i) Each A_i and each B_j is a positive atom, $m > 0$. If $n > 0$, then each predicate in conseq(C) is intensional, else if $n = 0$, then each predicate in conseq(C) is extensional.

(ii) Rules are range restricted, meaning that every variable that appears in conseq(C) also appears in antec(C).

We shall assume that the atoms in antec(C) are ordered, thus antec(C) forms a list $(A_1, A_2, \ldots, A_n)$. More importantly, the child nodes of rule nodes in cyclic trees will be ordered, left to right according to this ordering. INT(T) again consists of a set of rules with non-empty antecedant, and EXT(T) consists of (disjunctive) facts of the form

$$B_1 \vee B_2 \vee \ldots \vee B_m$$

where each B_i is an extensional positive atom. By condition (ii), each such B_i is ground.

Given a deductive database T in $\mathcal{L}$, let $gr(T)$ denote the set of ground instances of rules from T. A *model* of T is a set $M \subseteq \mathcal{H}$ such that $M \models gr(T)$.

6.2 Minimal model structure.

6.2.1 Definition. Let $\mathcal{P}$ and $\mathcal{O}$ be disjoint sets of positive atoms. Define MM(T, $\mathcal{P}$, $\mathcal{O}$) iff there is a ground instance $(\mathcal{P} \cup \mathcal{O})\theta$ such that some minimal model of T contains $\mathcal{P}\theta$ and is disjoint from $\mathcal{O}\theta$.

Quite clearly MM can be regarded as a first order extension of Definition 4.1.1. The following is analogous to Theorem 4.1.4.

6.2.2 Theorem. Let $\mathcal{P}$ and $\mathcal{O}$ be disjoint sets of positive atoms and $\mathcal{P}_0$ be an arbitrary non-empty subset of $\mathcal{P}$. Then MM(T, $\mathcal{P}$, $\mathcal{O}$) iff there is a rule $C \in T$ and a substitution μ such that:

(i) $\text{conseq}(C)\mu \cap \mathcal{P}_0\mu \neq \emptyset$,

(ii) $\text{antec}(C)\mu \cap \mathcal{P}_0\mu = \emptyset$,

(iii) $\mathcal{P}\mu \cup \text{antec}(C)\mu$ and $\mathcal{O}\mu \cup (\text{conseq}(C)\mu - \mathcal{P}_0\mu)$ are disjoint, and

(iv) MM(T, $\mathcal{P}\mu \cup \text{antec}(C)\mu$, $\mathcal{O}\mu \cup (\text{conseq}(C)\mu - \mathcal{P}_0\mu)$).

Proof ($\rightarrow$). Let M be a minimal model of T such that $\mathcal{P}\theta \subseteq M$ and $M \cap \mathcal{O}\theta = \emptyset$. Since M is minimal, we may find a ground instance $C\psi$ of a rule $C \in T$ such that $\text{antec}(C)\psi \subseteq M - \mathcal{P}_0\theta$ and $\text{conseq}(C)\psi \cap (M - \mathcal{P}_0\theta) = \emptyset$.

By renaming if necessary we may assume that C and $\mathcal{P} \cup \mathcal{O}$ contain no variables in common, and thus we may take the union of θ and ψ; say $\mu = \theta \cup \psi$.

Since $M \models T$, condition (i) holds. Conditions (ii) - (iv) hold trivially.

($\leftarrow$). If M is a minimal model of T satisfying (iv), say $\mathcal{P}\mu\delta \cup \text{antec}(C)\mu\delta \subseteq M$ and $M \cap (\mathcal{O}\mu\delta \cup (\text{conseq}(C)\mu - \mathcal{P}_0\mu)\delta) = \emptyset$, then clearly M satisfies MM(T, $\mathcal{P}$, $\mathcal{O}$). ∎

6.3 Weak deduction trees.

6.3.1 Definition. A *weak deduction tree* for $\mathcal{L}$ satisfies the conditions of Definition 5.1.4, with the following exceptions:

(a) Predicate nodes are labelled with (not necessarily ground) positive atoms.

(b) Rule nodes are labelled with instances of rules in T. If $RN_{C\phi}$ has parent N, then

$$\text{conseq}(C)\phi \cap \text{ACT}(N) \neq \emptyset,$$

$$\mathcal{O}(RN_{C\phi}) =_{def} \text{conseq}(C)\phi - \text{ACT}(N),$$

and

$$\mathcal{O}(RN_{C\phi}) \cap Pred(\mathcal{T}) = \emptyset.$$

(c) The child nodes of a rule node are ordered left to right using the ordering of antec(C).

Note that by $\text{conseq}(C)\phi \cap \text{ACT}(N) \neq \emptyset$, we mean that some atom (which may or may not contain variables) must lie in both conseq$(C)\phi$ and ACT(N). It is not sufficient for conseq$(C)\phi$ to contain an atom that is unifiable with some atom in ACT(N). A similar comment can be made concerning the other definitions in part (b).

6.3.2 Definition. Let $\mathcal{T}$ be a weak deduction tree. If N is a predicate node in $\mathcal{T}$, we will write CYC$(\mathcal{T}, N)$ to denote CYC(N) computed in $\mathcal{T}$ (cf. Definition 5.1.5), $lab(\mathcal{T}, N)$ to denote the label of N in $\mathcal{T}$, etc.

$\mathcal{T}$ is *cyclic* iff whenever $RN_{C\phi}$ is a rule node with parent N, then $\text{conseq}(C)\phi \cap \text{ACT}(\mathcal{T}, N) \subseteq \text{CYC}(\mathcal{T}, N)$.

If θ is a substitution, then $\mathcal{T}\theta$ is formed from $\mathcal{T}$ by apply θ to all labels in $\mathcal{T}$. Note that $\text{CYC}(\mathcal{T}\theta, N) \supseteq \text{CYC}(\mathcal{T}, N)\theta$.

It is easy to check that if $Pred(\mathcal{T}\theta) \cap \mathcal{O}(\mathcal{T}\theta) = \emptyset$, then $\mathcal{T}\theta$ is also a weak deduction tree. This condition does not however guarantee that the cyclic property is preserved. For instance if $\mathcal{T}$ is the cyclic tree depicted in Figure 6.3.2, then clearly $\mathcal{T}\theta$ is not cyclic. To guarantee that the cyclic property is preserved, we need to insist upon a slightly stronger condition.

$R(x)$

$|$

RN_C

$|$

$P(x)$ where D is the rule

$|$

RN_D $Q(x, y) \to P(x) \vee R(y)$, and

$|$

$Q(x, y)$ $\theta = \{x \to y\}$

Figure 6.3.2.

6.3.3 Proposition. Let $\mathcal{T}$ be a weak deduction tree and θ a substitution such that $Pred(\mathcal{T})\theta \cap \mathcal{O}(\mathcal{T})\theta = \emptyset$. Then:

(a) for each rule node $RN_{C\phi}$ in $\mathcal{T}$, $\mathcal{O}(\mathcal{T}\theta, RN_{C\phi\theta}) = \mathcal{O}(\mathcal{T}, RN_{C\phi})\theta$,

(b) $\mathcal{O}(\mathcal{T}\theta) = \mathcal{O}(\mathcal{T})\theta$, and

(c) If $\mathcal{T}$ is cyclic, then so is $\mathcal{T}\theta$.

Proof (a). $\mathcal{O}(\mathcal{T}\theta, RN_{C\phi\theta}) = \text{conseq}(C)\phi\theta - \text{ACT}(\mathcal{T}\theta, N) = \text{conseq}(C)\phi\theta - \text{ACT}(\mathcal{T}, N)\theta \subseteq (\text{conseq}(C)\phi - \text{ACT}(\mathcal{T}, N))\theta = \mathcal{O}(\mathcal{T}, RN_{C\phi})\theta$.

Suppose $A \in \text{conseq}(C)$ with $A\phi \in \mathcal{O}(\mathcal{T}, RN_{C\phi}) = \text{conseq}(C)\phi - \text{ACT}(\mathcal{T}, N)$. If $A\phi\theta \notin \text{conseq}(C)\phi\theta - \text{ACT}(\mathcal{T}\theta, N)$, then $A\phi\theta \in Pred(\mathcal{T}\theta) = Pred(\mathcal{T})\theta$, thus contradicting the fact that $Pred(\mathcal{T})\theta \cap \mathcal{O}(\mathcal{T}, RN_{C\phi})\theta = \emptyset$.

(b). This follows trivially from part (a).

(c). Let $RN_{C\phi}$ be a rule node in $\mathcal{T}$ and suppose that $A \in \text{conseq}(C)$ with $A\phi\theta \in \text{ACT}(\mathcal{T}\theta, N) \subseteq Pred(\mathcal{T})\theta$, then $A\phi\theta \notin \mathcal{O}(\mathcal{T})\theta$. In particular $A\phi \notin \mathcal{O}(\mathcal{T})$, thus since $\mathcal{T}$ is cyclic, $A\phi \in \text{CYC}(\mathcal{T}, N)$. Hence $A\phi\theta \in \text{CYC}(\mathcal{T}, N)\theta \subseteq \text{CYC}(\mathcal{T}\theta, N)$. ∎

This and Theorem 6.2.2 indicate how we should extend weak deduction trees (cf. Definition 5.2.2).

6.3.4 Definition. Let $\mathcal{T}$ be a weak deduction tree, N a predicate leaf node in $\mathcal{T}$, C a rule in T, and μ a substitution such that:

(i) $\text{conseq}(C)\mu \cap \text{ACT}(\mathcal{T}, N)\mu \neq \emptyset$,

(ii) $\mathit{Pred}(\mathcal{T})\mu \cup \text{antec}(C)\mu$ and $\mathcal{O}(\mathcal{T})\mu \cup (\text{conseq}(C)\mu - \text{ACT}(\mathcal{T}, N)\mu)$ are disjoint.

Then EXTEND($\mathcal{T}$, N, C, μ) is the tree formed by appending the node RN_C to T as a child of N (and appending the antecedants of C as child nodes of RN_C), and finally applying the substitution μ to all labels in the tree.

Notice that $\text{EXTEND}(\mathcal{T}, N, C, \mu) = \text{EXTEND}(\mathcal{T}\mu, N, C\mu, \emptyset)$. Also, if $\mathcal{T}$ is cyclic and $\text{conseq}(C)\mu \cap \text{ACT}(\mathcal{T}\mu, N) \subseteq \text{CYC}(\mathcal{T}\mu, N)$ then $\text{EXTEND}(\mathcal{T}, N, C, \mu)$ is also cyclic, by Proposition 6.3.3 (c). Clearly

$$\mathit{Pred}(\text{EXTEND}(\mathcal{T}, N, C, \mu)) = \mathit{Pred}(\mathcal{T})\mu \cup \text{antec}(C)\mu$$

and since $\mathit{Pred}(\mathcal{T})\mu \cap \mathcal{O}(\mathcal{T})\mu = \emptyset$, Proposition 6.3.3 (b) gives

$$\mathcal{O}(\text{EXTEND}(\mathcal{T}, N, C, \mu)) = \mathcal{O}(\mathcal{T})\mu \cup (\text{conseq}(C)\mu - \text{ACT}(\mathcal{T}, N)\mu),$$

hence the reason for condition (ii).

As in Section 5, our algorithm for determining minimal model membership constructs an unfactored cyclic tree. The main problem at the first order level is to ensure the finiteness of each branch generated in this construction. At the propositional level, this was achieved (cf. the proof of Theorem 5.1.14) by insisting that no predicate node is strongly redundant (cf. Definition 5.1.15), i.e., whenever a predicate node N is extended via a rule node RN_C, then $\text{antec}(C) \cap \text{CYC}(N) = \emptyset$.

At the first order level, we have the added complication that this cycle may change as a result of the application of subsequent substitutions. The constraints to be placed on T in the next section, and the left to right construction of $\mathcal{T}$ in Section 6.5 alleviate this problem.

6.4 Stratification.

In this and the following section we restrict our attention to a particular class of database (defined in Section 6.4.1 below). This section then details the basic properties of such databases in relation to cyclic trees. Section 6.5 contains the details of our cyclic tree construction, and also shows that this construction is terminating for databases in the given class.

6.4.1. For the remainder of Section 6, let $\ell : \{P_1, P_2, \ldots, P_n\} \to \mathbb{N}$ be such that $\ell(P) = 0$ iff P is extensional.

For each positive atom $P(\mathbf{t})$, $\ell(P(\mathbf{t})) =_{def} \ell(P)$, and for each $C \in T$, $\ell(C) =_{def} \min\{\ell(P) \mid P \in \text{conseq}(C)\}$.

We assume that for each $C \in T$:

(i) $\ell(R) \leq \ell(C)$ for each $R \in \text{antec}(C)$, and

(ii) for each variable x in C there is an $R(\mathbf{t}) \in \text{antec}(C)$ with $x \in \mathbf{t}$ and $\ell(R) < \ell(C)$.

We will also assume that rules are written in the form

$$A_1 \wedge A_2 \wedge \ldots \wedge A_n \to B_1 \vee B_2 \vee \ldots \vee B_m$$

where $\ell(A_1) \leq \ell(A_2) \leq \ldots \leq \ell(A_n)$. i.e., the antecedants are ordered according to ℓ, and more importantly, that child nodes of rule nodes are ordered (left to right) in the same manner. These assumptions on T have the following consequences.

6.4.2 Proposition. Let N and N' be predicate nodes in a cyclic tree with $N' > RN_C > N$, where N' is the parent of RN_C which in turn is the parent of N. Then $\ell(lab(N')) \geq \ell(C) \geq \ell(lab(N))$.

Proof. Suppose that the proposition holds along $\text{ACT}(N')$, then $\text{CYC}(N') \subseteq \{N'' \mid N'' \geq N', \ell(lab(N'')) = \ell(lab(N'))\}$. Moreover, since $\text{conseq}(C) \cap \text{CYC}(N') \neq \emptyset$, we must have that $\ell(C) \leq \ell(lab(N'))$. $\ell(lab(N)) \leq \ell(C)$ follows from condition (i) of Section 6.4.1. ▮

Thus in a cyclic tree, $N' > N \implies \ell(lab(N')) \geq \ell(lab(N))$.

6.4.3 Definition. Let $\mathcal{T}$ be a weak deduction tree, then a predicate node N is ℓ - *decreasing* iff $\ell(lab(N')) > \ell(lab(N))$ whenever $N' > N$.

By Proposition 6.4.2, in a cyclic tree we can tell whether N is ℓ - decreasing from the ℓ - value of the grandparent node. Moreover an ℓ - decreasing node cannot be redundant.

6.4.4 Proposition. Suppose that $\mathcal{T}$ is a weak deduction tree and N is an ℓ - decreasing predicate node, then $\mathrm{CYC}(\mathcal{T}\theta, N) = \{lab(\mathcal{T}\theta, N)\} = \mathrm{CYC}(\mathcal{T}, N)\theta$ for any instance $\mathcal{T}\theta$ of $\mathcal{T}$.

The following is an immediate consequence of condition (ii) of Section 6.4.1.

6.4.5 Lemma. Suppose that $RN_{C\theta}$ is a rule node, and $lab(M)$ is ground for each child node M of $RN_{C\theta}$ such that $\ell(lab(M)) < \ell(C)$. Then $\mathrm{conseq}(C)\theta$ and $\{lab(M') \mid M'$ is a child of $RN_{C\theta}\}$ are ground.

6.4.6 Definition. Let N be a predicate node in $\mathcal{T}$, then (cf. Figure 6.4.6)

$$below(\mathcal{T}, N) = \{N' \mid N' \text{ is a predicate node in } \mathcal{T}, N \geq N'\}, \text{ and}$$

$$left(\mathcal{T}, N) = \bigcup\{below(\mathcal{T}, N') \mid N' \text{ has a right sibling } N'' \geq N\}.$$

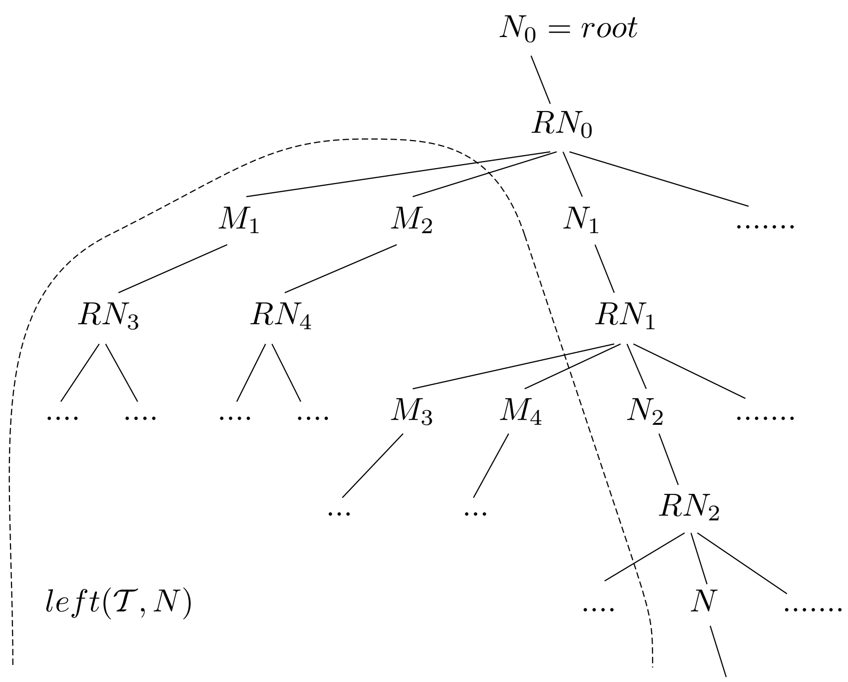

Figure 6.4.6.

6.4.7 Theorem. Let $\mathcal{T}$ be a cyclic tree and N a predicate node in $\mathcal{T}$ with child $RN_{C\theta}$ such that:

(a) $left(\mathcal{T}, N)$ contains no predicate leaf node, and

(b) for each child node M of $RN_{C\theta}$, if $\ell(lab(M)) < \ell(C)$ then $below(\mathcal{T}, M)$ contains no predicate leaf node.

Then $\{lab(N') \mid N' \geq N, \ell(lab(N')) = \ell(lab(N))\}$, $\mathcal{O}(RN_{C\theta})$, $\{lab(M') \mid M'$ is a child of $RN_{C\theta}\}$ and $left(\mathcal{T}, N)$ are all ground.

Proof. We proceed by induction. Suppose that the theorem holds for all nodes in $left(\mathcal{T}, N)$ and $below(\mathcal{T}, N) - \{N\}$. But then conditions (a) and (b) hold for all predicate nodes in $left(\mathcal{T}, N) \cup \{lab(M') \mid M'$ is a child of $RN_{C\theta}, \ell(lab(M')) < \ell(C)\}$, and hence by inductive hypothesis:

(i) $left(\mathcal{T}, N)$ is ground, and

(ii) for each child M of $RN_{C\theta}$, if $\ell(lab(M)) < \ell(C)$, then $lab(M)$ is ground.

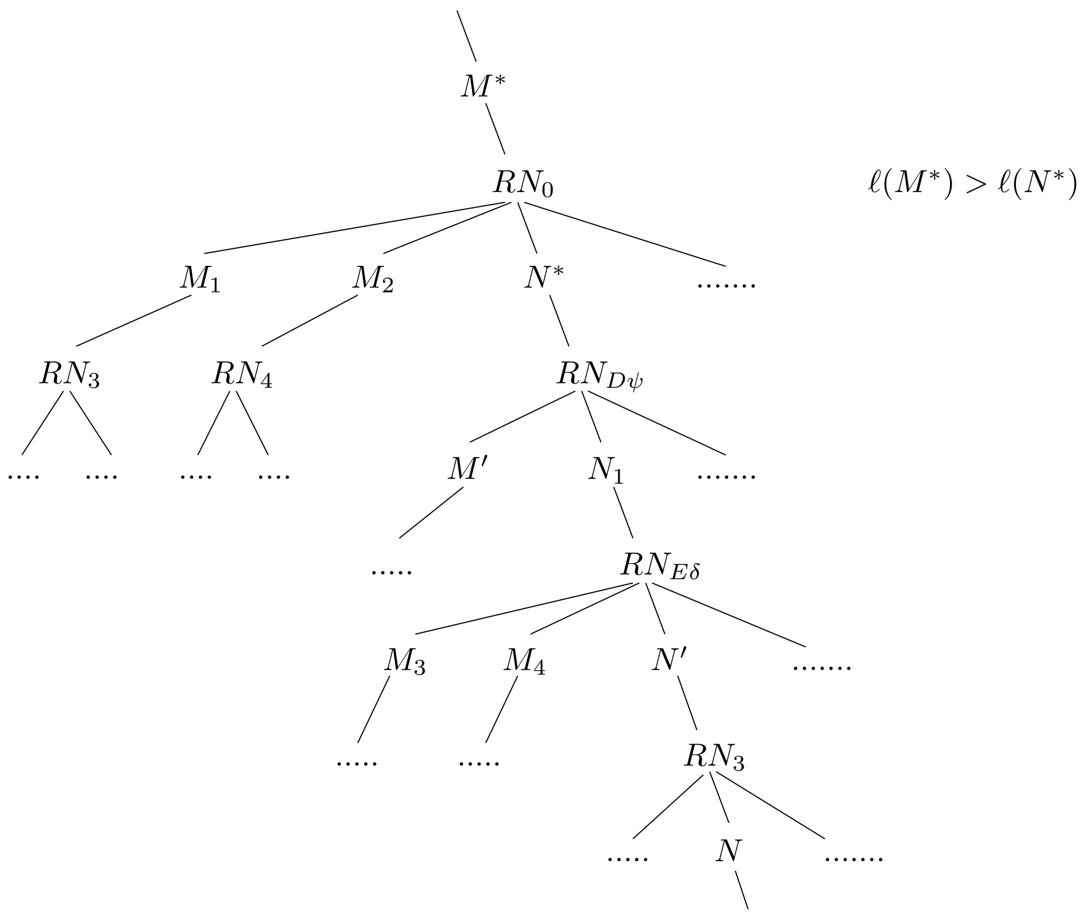

Figure 6.4.7.

By Lemma 6.4.5, $\{lab(M') \mid M'$ is a child of $RN_{C\theta}\}$ and conseq$(C)\theta$ are ground. This also shows that $\mathcal{O}(RN_{C\theta})$ is ground since $\mathcal{O}(RN_{C\theta}) \subseteq$ conseq$(C)\theta$.

Let $N^* \geq N$ be such that $\ell(lab(N^*)) = \ell(lab(N))$ and N^* is ℓ - decreasing. (i.e., N^* is the top-most node on ACT(N) such that $\ell(lab(N^*)) = \ell(lab(N))$.)

We first show that $lab(N^*)$ is ground. By Proposition 6.4.4, CYC$(\mathcal{T}, N^*) = \{lab(N^*)\}$, hence if $RN_{D\psi}$ is the child of N^*, then $lab(N^*) \in$ conseq$(D)\psi$. It thus suffices to show that conseq$(D)\psi$ is ground.

Let M' be a child of $RN_{D\psi}$ such that $\ell(lab(M')) < \ell(D)$. If $N^* > N$, then by Proposition 6.4.2 and the assumptions made in Section 6.4.1 concerning the ordering of child nodes, $M' \in left(\mathcal{T}, N)$ (cf. Figure 6.4.7). Hence by (i), $lab(M')$ is ground. If $N^* = N$, then $lab(M')$ is ground by (ii). Thus conseq$(D)\psi$ is ground by Lemma 6.4.5.

Suppose that $N^* > N' \geq N$, then N' is not ℓ - decreasing. If $RN_{E\delta}$ is the parent of N', then by Proposition 6.4.2, $\ell(E) = \ell(lab(N'))$, and thus $\{M \mid M$ is a child of $RN_{E\delta}, \ell(lab(M)) < \ell(E)\} \subseteq left(\mathcal{T}, N)$ (cf. Figure 6.4.7) and hence by (i) is ground. Thus by Lemma 6.4.5, $lab(N')$ is ground. ∎

6.4.8 Corollary. Let $\mathcal{T}$ be a cyclic tree and M' a predicate node such that $left(\mathcal{T}, M')$ does not contain any predicate leaf node.

(a) If M' is not ℓ - decreasing, then $\{lab(\mathcal{T}, N') \mid N' \geq M', \ell(lab(N')) = \ell(lab(M'))\}$ is ground.

For any substitution θ:

(b) M' is strongly redundant in $\mathcal{T}\theta$ iff it is strongly redundant in $\mathcal{T}$, and

(c) CYC$(\mathcal{T}\theta, M') =$ CYC$(\mathcal{T}, M')\theta$.

Proof (a). Since M' is not ℓ - decreasing, it cannot be the root node, and thus applying the previous theorem to the grandparent of M' yields the result.

(b) If M' is strongly redundant in $\mathcal{T}\theta$, then M' is not ℓ - decreasing in which case the result follows from part (a). The converse is trivial.

(c) If M' is ℓ - decreasing, then CYC$(\mathcal{T}\psi, M') = \{lab(\mathcal{T}, M')\psi\}$ for any ψ. The case when M' is not ℓ - decreasing is covered in part (a). ∎

6.5 Computing minimal model membership.

In order to test whether $\text{MM}(T, \{P(\mathbf{t})\}, \emptyset)$, we need to generate an unfactored cyclic tree for (some instance of) $P(\mathbf{t})$ in $gr(T)$. As indicated in earlier sections, this construction will proceed in a left to right manner, and termination will be guaranteed by ensuring that no predicate node is strongly redundant (at any stage in the construction). We thus extend Definition 6.3.4 as follows.

6.5.1 Definition. Let $\mathcal{T}$ be a weak deduction tree and N a predicate leaf node in $\mathcal{T}$. The extension step $\text{EXTEND}(\mathcal{T}, N, C, \mu)$ is *valid* iff:

(i) $\emptyset \neq \text{conseq}(C)\mu \cap \text{ACT}(\mathcal{T}\mu, N) \subseteq \text{CYC}(\mathcal{T}\mu, N)$,

(ii) $Pred(\mathcal{T})\mu \cup \text{antec}(C)\mu$ and $\mathcal{O}(\mathcal{T})\mu \cup (\text{conseq}(C)\mu - \text{ACT}(\mathcal{T}\mu, N))$ are disjoint,

(iii) no predicate node in $\text{EXTEND}(\mathcal{T}, N, C, \mu)$ is strongly redundant, and

(iv) there is no predicate leaf node in $left(\mathcal{T}, N)$ (i.e., the construction proceeds in a depth first, left to right manner).

In order to keep the computation as general as possible, we would normally expect that μ should be a most general unifier of some subset of $\text{conseq}(C)$ with some subset of $\text{ACT}(\mathcal{T}, N)$.

By the remarks following Definition 6.3.4, if $\mathcal{T}$ is cyclic, then so is any valid extension of $\mathcal{T}$. The following proposition summarises the other main properties of a sequence of valid extensions.

6.5.2 Proposition. Let $(\mathcal{T}_i \mid 0 \leq i \leq m)$ be a sequence of cyclic trees such that $\mathcal{T}_0$ consists of a single predicate node $N_{P(\mathbf{t})}$, and each $\mathcal{T}_{i+1}$ is a valid extension of $\mathcal{T}_i$ $(0 \leq i < m)$. Suppose that $\mathcal{T}_m$ is not unfactored, and N is the unique predicate leaf node in $\mathcal{T}_m$ satisfying condition (iv) of Definition 6.5.1. Then:

(a) $left(\mathcal{T}_m, N)$ is ground.

(b) Each predicate node in $\{M' \mid \exists N_0 \geq N, M' \text{ is a right sibling of } N_0\}$ is a leaf.

(c) If $\mathcal{T}_{m+1} = \text{EXTEND}(\mathcal{T}_m, N, C, \mu)$ satisfies conditions (i) and (ii) of Definition 6.5.1, and no node in $\{M' \mid M' \text{ is a child of } RN_{C\mu}, \ell(lab(M')) = \ell(C)\} \cup \{M' \mid \exists N_0 \geq N, M'$ is a right sibling of $N_0\}$ is strongly redundant in $\mathcal{T}_{m+1}$, then $\mathcal{T}_{m+1}$ is a valid extension of $\mathcal{T}_m$.

Proof. Part (a) follows immediately from Theorem 6.4.7, and part (b) is obvious from the left to right construction.

(c). If $\mathcal{T}_{m+1}$ contains a strongly redundant predicate node M', then M' must be on ACT(N) or in $left(\mathcal{T}_{m+1}, N) = left(\mathcal{T}_m, N)$. However, since each $\mathcal{T}_{i+1}$ is a valid extension of $\mathcal{T}_i$, M' cannot be strongly redundant in $\mathcal{T}_m$.

Since M' is strongly redundant in $\mathcal{T}_{m+1}$ it must also be strongly redundant in $\mathcal{T}_m\mu$, thus contradicting Corollary 6.4.8 (b) (since $left(\mathcal{T}_m, M')$ contains no predicate leaf node). ∎

6.5.3 Theorem (Termination). There is no infinite sequence of cyclic trees $(\mathcal{T}_i \mid i \geq 0)$ such that $\mathcal{T}_0$ consists of a single predicate node, and for each $i \geq 0$, $\mathcal{T}_{i+1}$ is a valid extension of $\mathcal{T}_i$.

Proof. Suppose that each $\mathcal{T}_{i+1}$ = EXTEND(T_i, N_i, C_i, μ_i). Clearly if the series of extensions is infinite, then it must construct an infinite branch. Moreover, by Proposition 6.4.2, the ℓ values are monotonic decreasing along such a branch, thus we may find a sequence $r_0 < r_1 < r_2 < \ldots$ such that for each $j \geq 0$:

(i) $\ell(N_{r_j}) = \ell(N_{r_0})$, and

(ii) N_{r_j} is the grandparent of $N_{r_{j+1}}$ in $\mathcal{T}_{r_{j+1}}$.

For each $j \geq 0$, $\{lab(\mathcal{T}_{r_{j+1}}, M') \mid M' \geq N_{r_{j+1}}, \ell(lab(\mathcal{T}_{r_{j+1}}, M')) = \ell(lab(\mathcal{T}_{r_{j+1}}, N_{r_{j+1}}))\}$ is ground by Corollary 6.4.8(a) and condition (iv) of Definition 6.5.1.

But then since $lab(N_{r_{j+1}})$ is not strongly redundant in $\mathcal{T}_{r_{j+1}}$, $N_{r_{j+1}} \notin$ CYC($\mathcal{T}_{r_{j+1}}, N_{r_j}$), and as in the proof of Theorem 5.1.14, this guarantees the finiteness of the branch. ∎

6.5.4 Theorem (Correctness). Let $(\mathcal{T}_i \mid 0 \leq i \leq n)$ be such that:

(i) $\mathcal{T}_0$ consists of a single predicate node,

(ii) for each $i < n$, $\mathcal{T}_{i+1}$ is a valid extension of $\mathcal{T}_i$, and

(iii) $\mathcal{T}_n$ contains no predicate leaf node.

Then $\mathcal{T}_n$ is a ground unfactored cyclic tree.

Proof. Each $\mathcal{T}_i$ is cyclic by the remark following Definition 6.5.1. The fact that $\mathcal{T}_n$ is ground follows from Theorem 6.4.7. ∎

6.5.5 Theorem (Completeness). Suppose that $P(\mathbf{t})$ is a positive atom such that MM(T, $\{P(\mathbf{t})\}$, $\emptyset$). Then there is a sequence $(\mathcal{T}_i \mid 0 \leq i \leq n)$ satisfying the conditions of Theorem 6.5.4 such that $\mathcal{T}_0$ consists of the single predicate node $N_{P(\mathbf{t})}$ and $T \not\models \bigvee \mathcal{O}(\mathcal{T}_n)$.

Proof. Let M be a minimal model of T containing some ground instance $P(\mathbf{t}\theta)$ of $P(\mathbf{t})$, and (as in the proof of Theorem 5.1.14), let $\mathcal{T}$ be an unfactored cyclic tree for $P(\mathbf{t}\theta)$ in $gr(T)$ such that $Pred(\mathcal{T}) \subseteq M \subseteq \mathcal{H}-\mathcal{O}(\mathcal{T})$ and $\mathcal{T}$ contains no strongly redundant predicate node.

We inductively construct sequences $(\mathcal{T}_i \mid 0 \leq i \leq n)$ and $(\theta_i \mid 0 \leq i \leq n)$ such that:

(i) $(\mathcal{T}_i \mid 0 \leq i \leq n)$ satisfies the conditions of Theorem 6.5.4, $\mathcal{T}_{i+1} =$ EXTEND($\mathcal{T}_i$, N_i, C_i, μ_i),

(ii) $\mathcal{T}_i\theta_i$ is an initial segment of $\mathcal{T}$, and

(iii) $\mathcal{O}(\mathcal{T}_i)\theta_i \subseteq \mathcal{O}(\mathcal{T})$.

Note that $\mathcal{T}_i$ can contain no strongly redundant predicate node (cf. condition (iii) of Definition 6.5.1), else the same node would be strongly redundant in $\mathcal{T}$ by condition (ii) above.

Clearly $\mathcal{T}_0$ is as given in the statement of the Theorem, and $\theta_0 = \theta$. Suppose that we are given $\mathcal{T}_i$ and θ_i. If $\mathcal{T}_i$ has no predicate leaf node, then $n = i$ (and $\mathcal{T}_n = \mathcal{T}$), else let N_i be the (unique) predicate leaf node such that $left(\mathcal{T}_i, N_i)$ contains no predicate leaf node (cf. condition (iv) of Definition 6.5.1).

Let $RN_{C_i\psi_i}$ be the child node of N_i in $\mathcal{T}$. By renaming the variables in C_i if necessary, we may assume that C_i and $\mathcal{T}_i$ have no variables in common and so we may write $\delta_i = \theta_i \cup \psi_i$. Let $\mathcal{C} = \{B \in \text{conseq}(C_i) \mid B\psi_i \in \text{ACT}(\mathcal{T}, N_i)\} = \{B \in \text{conseq}(C_i) \mid B\psi_i \in \text{CYC}(\mathcal{T}, N_i)\}$, then $\mathcal{C} \neq \emptyset$.

Case 1: N_i is ℓ - decreasing.

$\text{CYC}(\mathcal{T}, N_i) = \{lab(\mathcal{T}, N_i)\}$, and thus $\mathcal{C}\delta_i = \mathcal{C}\psi_i = \{lab(\mathcal{T}, N_i)\} = \{lab(\mathcal{T}_i, N_i)\theta_i\} = \{lab(\mathcal{T}_i, N_i)\delta_i\}$. Let μ_i be the most general unifier of $lab(\mathcal{T}_i, N_i)$ and $\mathcal{C}$, then δ_i extends μ_i, say $\delta_i = \mu_i\eta$.

Case 2: N_i is not ℓ - decreasing.

By Corollary 6.4.8(a), $lab(\mathcal{T}_i, N')$ is ground for each $N' \geq N_i$ such that $\ell(lab(N')) = \ell(lab(N_i))$. Let μ_i be equal to ψ_i restricted to those variables which occur in $\mathcal{C}$, then μ_i is the most general unifier of $\mathcal{C}$ with $\{B\psi_i \mid B \in \mathcal{C}\}$ which in turn is a subset of CYC($\mathcal{T}, N_i$) = CYC($\mathcal{T}_i, N_i$). Again δ_i extends μ_i, say $\delta_i = \mu_i\eta$.

In either case we can set $\theta_{i+1} = \eta$ whence $\mathcal{T}_{i+1}\theta_{i+1}$ = EXTEND($\mathcal{T}_i, N_i, C_i, \mu_i)\theta_{i+1}$ = EXTEND($\mathcal{T}_i\mu_i, N_i, C_i\mu_i, \emptyset)\theta_{i+1}$ = EXTEND($\mathcal{T}_i\theta_i, N_i, C_i\psi_i, \emptyset$) which in turn is an initial segment of $\mathcal{T}$ by our choice of $C_i\psi_i$.

We thus set about showing that $\mathcal{T}_{i+1}$ is a valid extension of $\mathcal{T}_i$ and that $\mathcal{O}(\mathcal{T}_{i+1})\theta_{i+1} \subseteq \mathcal{O}(\mathcal{T})$.

As mentioned earlier, since $\mathcal{T}_{i+1}\theta_{i+1}$ is an initial segment, $\mathcal{T}_{i+1}$ can contain no strongly redundant predicate node (cf. condition (iii) of Definition 6.5.1.)

Claim (a): $\emptyset \neq$ conseq$(C_i)\mu_i \cap$ ACT$(\mathcal{T}_i\mu_i, N_i) \subseteq$ CYC$(\mathcal{T}_i\mu_i, N_i)$ (cf. condition (i) of Definition 6.5.1).

The fact that conseq$(C_i)\mu_i \cap$ ACT$(\mathcal{T}_i\mu_i, N_i) \neq \emptyset$ follows from the choice of μ_i. If $B\mu_i \in$ conseq$(C_i)\mu_i \cap$ ACT$(\mathcal{T}_i\mu_i, N_i)$ then $B\mu_i\theta_{i+1} \in$ conseq$(C_i)\psi_i \cap$ ACT$(\mathcal{T}, N_i)$, hence $B \in \mathcal{C}$; thus $B\mu_i \in$ CYC$(\mathcal{T}_i\mu_i, N_i)$, again by our choice of μ_i.

Claim (b): $\mathcal{O}(\mathcal{T}_{i+1}, RN_{C_i\mu_i})\theta_{i+1} = \mathcal{O}(\mathcal{T}, RN_{C_i\psi_i})$.

If $B \in$ conseq(C_i) with $B\psi_i = B\mu_i\theta_{i+1} \in$ ACT$(\mathcal{T}, N_i)$ then $B \in \mathcal{C}$ and hence by the choice of μ_i, $B\mu_i \in$ ACT$(\mathcal{T}_i\mu_i, N_i)$. Thus $\mathcal{O}(\mathcal{T}_{i+1}, RN_{C_i\mu_i})\theta_{i+1}$ = (conseq$(C_i)\mu_i$ − ACT$(\mathcal{T}_i\mu_i,\ N_i))\theta_{i+1} \subseteq$ conseq$(C_i)\psi_i$ − ACT$(\mathcal{T}, N_i) = \mathcal{O}(\mathcal{T}, RN_{C_i\psi_i})$.

Conversely, conseq$(C_i)\psi_i$− ACT$(\mathcal{T}, N_i)$ = conseq$(C_i)\mu_i\theta_{i+1}$− ACT$(\mathcal{T}_i\mu_i, N_i)\theta_{i+1} \subseteq$ (conseq$(C_i)\mu_i$ − ACT$(\mathcal{T}_i\mu_i, N_i))\theta_{i+1}$.

Claim (c): $\mathit{Pred}(\mathcal{T}_i)\mu_i \cap \mathcal{O}(\mathcal{T}_i)\mu_i = \emptyset$.

If $B \in \mathcal{O}(\mathcal{T}_i)$, then $B\mu_i\theta_{i+1} = B\theta_i \in \mathcal{O}(\mathcal{T}_i)\theta_i \subseteq \mathcal{O}(\mathcal{T})$ by condition (iii). Thus if $B\mu_i \in \mathit{Pred}(\mathcal{T}_i)\mu_i$, then $B\mu_i\theta_{i+1} \in \mathit{Pred}(\mathcal{T}_i)\mu_i\theta_{i+1} = \mathit{Pred}(\mathcal{T}_i)\theta_i \subseteq \mathit{Pred}(\mathcal{T})$, a contradiction.

Claim (d): $\mathcal{O}(\mathcal{T}_{i+1})\theta_{i+1} \subseteq \mathcal{O}(\mathcal{T})$.

$\mathcal{O}(\mathcal{T}_{i+1})\theta_{i+1} = \mathcal{O}(\mathcal{T}_i\mu_i)\theta_{i+1} \cup \mathcal{O}(\mathcal{T}_{i+1}, RN_{C_i\mu_i})\theta_{i+1}$.

By Claim (c) and Proposition 6.3.3(b), $\mathcal{O}(\mathcal{T}_i\mu_i)\theta_{i+1} = \mathcal{O}(\mathcal{T}_i)\mu_i\theta_{i+1} = \mathcal{O}(\mathcal{T}_i)\theta_i \subseteq \mathcal{O}(\mathcal{T})$ by condition (iii).

In addition, $\mathcal{O}(\mathcal{T}_{i+1}, RN_{C_i\mu_i})\theta_{i+1} = \mathcal{O}(\mathcal{T}, RN_{C_i\psi_i}) \subseteq \mathcal{O}(\mathcal{T})$ by Claim (b).

Claim (e): $\mathit{Pred}(\mathcal{T}_i)\mu_i \cup$ antec$(C_i)\mu_i$ and $\mathcal{O}(\mathcal{T}_i)\mu_i \cup$ (conseq$(C_i)\mu_i$ − ACT$(\mathcal{T}_i\mu_i, N_i)$) are disjoint (cf. condition (ii) of Definition 6.5.1).

Notice that the first of these sets is $\mathit{Pred}(\mathcal{T}_{i+1})$ and (since $\mathcal{O}(\mathcal{T}_i\mu_i) = \mathcal{O}(\mathcal{T}_i)\mu_i$) the second is equal to $\mathcal{O}(\mathcal{T}_{i+1})$. The claim then follows from the fact that $\mathit{Pred}(\mathcal{T}_{i+1})\theta_{i+1} \subseteq$

$Pred(\mathcal{T}) \subseteq \mathcal{H} - \mathcal{O}(\mathcal{T}) \subseteq \mathcal{H} - \mathcal{O}(\mathcal{T}_{i+1})\theta_{i+1}$ (by Claim (d)). ∎

The above results show that generating unfactored trees via valid extensions provides a method of testing $\mathrm{MM}(T, \{P(\mathbf{t})\}, \emptyset)$, which we summarise as the following corollary.

6.5.6 Corollary. $\mathrm{MM}(T, \{P(\mathbf{t})\}, \emptyset)$ iff there is a sequence of cyclic trees $(\mathcal{T}_i \mid i \leq n)$ such that

(i) $\mathcal{T}_0 = \{N_{P(\mathbf{t})}\}$,

(ii) each $\mathcal{T}_{i+1}$ is a valid extension of $\mathcal{T}_i$,

(iii) $\mathcal{T}_n$ is unfactored, and

(iv) $T \not\models \bigvee \mathcal{O}(\mathcal{T}_n)$.

§7. PERFECT MODELS.

In this section, we conclude by showing that the methods of the previous sections can be extended to cover databases in which rules may contain negated premises. Such premises are treated using the perfect model semantics of Przymusinski [19]. For the sake of simplicity we return to the propositional level.

7.1 Preliminaries.

Throughout we assume the existence of a function $\ell : \mathcal{L} \to \{0, 1, \ldots, n-1\}$ (where $n = |\mathcal{L}|$) such that for each $P \in \mathcal{L}$, $\ell(P) = 0$ iff $P \in \mathrm{EXT}(\mathcal{L})$.

7.1.1 Definition. If $\mathcal{P} \subseteq \mathcal{L}$ and $\alpha \leq n-1$, then we define $\mathcal{P}_\alpha = \{P \in \mathcal{P} \mid \ell(P) = \alpha\}$, $\mathcal{P}_{\leq\alpha} = \{P \in \mathcal{P} \mid \ell(P) \leq \alpha\}$ and $\mathcal{P}_{<\alpha} = \{P \in \mathcal{P} \mid \ell(P) < \alpha\}$, etc.

7.1.2 Definition. A *rule* C is a logical formula of the form

$$A_1 \wedge A_2 \wedge \ldots \wedge A_h \wedge \overline{A_{h+1}} \wedge \overline{A_{h+2}} \wedge \ldots \wedge \overline{A_{h+r}} \to B_1 \vee B_2 \vee \ldots \vee B_k$$

such that:

(i) each A_i and each B_j is a predicate in $\mathcal{L}$, $k > 0$,
(ii) for each $j \leq k$, $\ell(B_j) = \ell(B_1)$,
(iii) for each $i \leq h$, $\ell(A_i) \leq \ell(B_1)$,
(iv) for $1 \leq i \leq r$, $\ell(A_{h+i}) < \ell(B_1)$,
(v) if $h + r > 0$, then each B_j is intensional, and
(vi) if $h = r = 0$, then each B_j is extensional.

For each such rule, $\mathrm{antec}(C) = \{A_1, A_2, \ldots, A_h\}$, $\mathcal{N}(C) = \{A_{h+1}, A_{h+2}, \ldots, A_{h+r}\}$, $\overline{\mathcal{N}}(C) = \{\overline{A_{h+1}}, \overline{A_{h+2}}, \ldots, \overline{A_{h+r}}\}$, and $\mathrm{conseq}(C) = \{B_1, B_2, \ldots, B_k\}$. Let $T_\alpha = \{C \in T \mid \mathrm{conseq}(C) \subseteq \mathcal{L}_\alpha\}$, etc., (where $\mathcal{L}_\alpha = \{P \in \mathcal{L} \mid \ell(P) = \alpha\}$, cf. Definition 7.1.1).

Let C^* be the rule $A_1 \wedge A_2 \wedge \ldots \wedge A_h \to B_1 \vee B_2 \vee \ldots \vee B_k$. $T^* = \{C^* \mid C \in T\}$.

7.1.3 Definition. If $M \subseteq \mathcal{L}$, then M *satisfies* the above rule iff

$$\{A_1, A_2, \ldots, A_h\} \subseteq M \implies M \cap \{A_{h+1}, A_{h+2}, \ldots, A_{h+r}, B_1, B_2, \ldots, B_k\} \neq \emptyset.$$

$M \models T$ iff $M \models C$ for each $C \in T$.

7.1.4 Proposition. $M \models T$ iff for each $\alpha \leq n-1$, $M_{\leq\alpha} \models T_{\leq\alpha}$.

7.2 Perfect models.

The level function ℓ is intended to indicate priorities amongst the predicates. In particular, that the minimizing of predicates of low ℓ - value should take priority. This naturally leads to the definition of a perfect model below.

7.2.1 Definition [19].
(a) Let M' and M be models of T, then M' is *preferable* to M iff $M' \neq M$ and whenever $P \in M' - M$, there is a $Q \in M - M'$ such that $\ell(P) > \ell(Q)$.
(b) A model M of T is *perfect* iff there is no model of T which is preferable to M.

7.2.2 Theorem. A model M of T is perfect iff whenever $\alpha \leq n-1$ and $M' \subset M_\alpha$, then $M' \cup M_{<\alpha} \not\models T_\alpha$.

Proof ($\rightarrow$). Suppose that $M \subseteq \mathcal{L}$ is a perfect model of T and that $M' \subset M_\alpha$ with $M' \cup M_{<\alpha} \models T_\alpha$. Let $M'' = M' \cup M_{<\alpha} \cup \mathcal{L}_{>\alpha}$, then clearly $M'' \models T$, and M'' is preferable to M.
($\leftarrow$). Suppose that $M'' \models T$ is preferable to M, then $M - M'' \neq \emptyset$.

Let $\alpha = \min\{\gamma \mid M_\gamma - M''_\gamma \neq \emptyset\}$, then since M'' is preferable to M we have that $M''_{\leq\alpha} \subseteq M_{\leq\alpha}$, hence $M''_{<\alpha} = M_{<\alpha}$ and $M''_\alpha \subset M_\alpha$. Moreover $M''_\alpha \cup M_{<\alpha} = M''_{\leq\alpha} \models T_\alpha$, contradicting our hypothesis. ∎

7.2.3 Proposition [19]. Any perfect model of T is minimal. If $\mathcal{N}(C) = \emptyset$ for each $C \in T$, then the converse holds.

7.2.4 Definition. Let $\mathcal{P}$ and $\mathcal{O}$ be disjoint subsets of $\mathcal{L}$, then define PM(T, $\mathcal{P}$, $\mathcal{O}$) iff there is a perfect model of T containing $\mathcal{P}$ but disjoint from $\mathcal{O}$.

In this section we are working with respect to the semantics defined by perfect models. We thus make the following definition, and in particular note that PM allows us to handle both positive and negative data.

7.2.5 Definition. Given $\mathcal{Q} \subseteq \mathcal{L}$, then:

(a) $\bigvee \mathcal{Q}$ may be *inferred* from T iff $\bigvee \mathcal{Q}$ is true in every perfect model of T, i.e., $\neg\,\mathrm{PM}(T, \emptyset, \mathcal{Q})$, and

(b) $\bigvee\{\overline{R} \mid R \in \mathcal{Q}\}$ may be *assumed* from T iff no perfect model of T contains $\mathcal{Q}$, i.e., $\neg\,\mathrm{PM}(T, \mathcal{Q}, \emptyset)$.

7.2.6 Lemma. Let $\alpha \leq n-1$.

(a) If M is a perfect model of T, then $M_{\leq\alpha}$ is a perfect model of $T_{\leq\alpha}$.

(b) If $M \subseteq \mathcal{L}_{\leq\alpha}$ is a perfect model of $T_{\leq\alpha}$, then M may be extended to a perfect model M' of T such that $M'_{\leq\alpha} = M$.

7.2.7 Corollary. Let $\alpha \leq n-1$ and $\mathcal{P}$ and $\mathcal{O}$ be disjoint subsets of $\mathcal{L}_{\leq\alpha}$, then $\mathrm{PM}(T, \mathcal{P}, \mathcal{O})$ iff $\mathrm{PM}(T_{\leq\alpha}, \mathcal{P}, \mathcal{O})$.

7.3 Covers and trees.

In this section and the next we show that perfect model membership can be characterised in terms of cyclic trees.

7.3.1 Definition. Let $0 < \alpha \leq n-1$ and $\mathcal{O} \subseteq \mathcal{L}_\alpha$, then a *cover* of $\mathcal{O}$ in T_α is a non-complementary set of literals $\mathcal{C}$ such that $\mathcal{O} \subseteq \mathcal{C} \subseteq \mathcal{L}_{\leq\alpha} \cup \{\overline{P} \mid P \in \mathcal{L}_{<\alpha}\}$ and

$$\forall C \in T_\alpha\,(\mathrm{conseq}(C) \subseteq \mathcal{C} \implies (\mathrm{antec}(C) \cup \overline{\mathcal{N}}(C)) \cap \mathcal{C} \neq \emptyset).$$

Note that the requirement $\mathcal{C} \subseteq \mathcal{L}_{\leq\alpha} \cup \{\overline{P} \mid P \in \mathcal{L}_{<\alpha}\}$ follows from conditions (iii) and (iv) of Definition 7.1.2.

7.3.2 Lemma. If $0 < \alpha \leq n-1$, $\mathcal{O} \subseteq \mathcal{L}_\alpha$ and $M \models T_{\leq\alpha} \wedge \neg \bigvee \mathcal{O}$, then we may find a cover $\mathcal{C}$ of $\mathcal{O}$ in T_α such that $\{P \in \mathcal{L} \mid P \in \mathcal{C}\} \cap M = \emptyset$ and $\{P \in \mathcal{L} \mid \overline{P} \in \mathcal{C}\} \subseteq M$, i.e., $M \models \neg \bigvee \mathcal{C}$.

Proof. Let $\mathcal{C} = (\mathcal{L}_{\leq\alpha} - M_{\leq\alpha}) \cup \{\overline{P} \mid P \in M_{<\alpha}\}$. ∎

7.3.3 Definition. Let $\alpha \leq n-1$ and $\mathcal{T}$ be a weak deduction tree in T^*_α (cf. Definition 7.1.2), then define

$$\mathcal{N}(\mathcal{T}) = \bigcup\{\mathcal{N}(C) \mid C^* \text{ labels some rule node in } \mathcal{T}\}.$$

Notice that $\mathcal{N}(\mathcal{T}) \subseteq \mathcal{L}_{<\alpha}$, and that $\mathcal{O}(\mathcal{T}) \subseteq \mathcal{L}_\alpha$ by the conditions of Definition 7.1.2.

The following is analogous to Propositions 6.4.2 and 5.6.1(c).

7.3.4 Lemma. Let $0 < \alpha \leq n-1$ and $\mathcal{T}$ be a cyclic tree in T^*_α. Then:

(i) If $N > N'$, then $\alpha = \ell(lab(N)) \geq \ell(lab(N'))$.

(ii) If $\ell(lab(N)) = \alpha$, then $\mathrm{CYC}(N) \subseteq \mathcal{L}_\alpha$.

(iii) Every leaf node in $\mathcal{T}$ is a predicate node. If $\ell(lab(N)) < \alpha$, then N is a leaf.

7.3.5 Lemma. Let $0 < \alpha \leq n-1$, $P \in \mathcal{L}_\alpha$ and $M \subseteq \mathcal{L}_{\leq\alpha}$ be a perfect model of $T_{\leq\alpha}$ containing P. Then there is a cyclic tree $\mathcal{T}$ for P in T^*_α such that:

(i) $Pred(\mathcal{T}) \subseteq M \subseteq \mathcal{L} - \mathcal{O}(\mathcal{T})$,

(ii) $M \cap \mathcal{N}(\mathcal{T}) = \emptyset$, and

(iii) for each leaf node $N \in \mathcal{T}$, $\ell(lab(N)) < \alpha$.

Proof. The construction is similar to that seen in the proof of Theorem 5.1.14. If N is a predicate leaf node with $\ell(lab(N)) = \alpha$, then by Proposition 7.3.4, $\mathrm{CYC}(N) \subseteq M_\alpha$. By Theorem 7.2.2, $M - \mathrm{CYC}(N) \not\models T_\alpha$, thus we may find a rule $C \in T_\alpha$ such that $\mathrm{antec}(C) \subseteq M - \mathrm{CYC}(N)$, $\mathcal{N}(C) \cap (M - \mathrm{CYC}(N)) = \emptyset$, and $\mathrm{conseq}(C) \cap (M - \mathrm{CYC}(N)) = \emptyset$. Since $\mathcal{N}(C) \subseteq \mathcal{L}_{<\alpha}$ we have $\mathcal{N}(C) \cap \mathrm{CYC}(N) = \emptyset$, and in particular $\mathcal{N}(C) \cap M = \emptyset$. Thus we may extend N via the rule node RN_C. Termination follows as in the proof of Theorem 5.1.14. ∎

7.4 Computing PM.

7.4.1 Theorem. If $\mathcal{P}$ and $\mathcal{O}$ are disjoint subsets of $\mathcal{L}_0 = \mathrm{EXT}(\mathcal{L})$, then $\mathrm{PM}(T_0, \mathcal{P}, \mathcal{O})$ iff $\mathrm{MM}(T_0, \mathcal{P}, \mathcal{O})$.

The theorem is an immediate consequence of Proposition 7.2.3. As mentioned earlier, the methods of [9, Section 5] can be used to determine $\mathrm{MM}(T_0, \mathcal{P}, \mathcal{O})$.

7.4.2 Theorem. Let $0 < \alpha \leq n-1$ and $\mathcal{P}$ and $\mathcal{O}$ be disjoint subsets of $\mathcal{L}_{\leq\alpha}$, then $\mathrm{PM}(T_{\leq\alpha}, \mathcal{P}, \mathcal{O})$ iff there is a sequence $(\mathcal{T}_i \mid 0 \leq i \leq m)$ of cyclic trees in T^*_α such that:

(a) $\mathcal{P}_\alpha \subseteq \bigcup_{i=0}^m \mathit{Pred}(\mathcal{T}_i)$,

(b) if N is a predicate leaf node in $\mathcal{T}_i$, then $\mathit{lab}(N) \in \mathcal{L}_{<\alpha} \cup \bigcup_{j<i} \mathit{Pred}(\mathcal{T}_j)$, and

(c) there is a cover $\mathcal{C}$ of $\mathcal{O}_\alpha \cup \bigcup_{i=0}^m \mathcal{O}(\mathcal{T}_i)$ in T_α such that $\mathcal{C} \cap \bigcup_{i=0}^m \mathit{Pred}(\mathcal{T}_i) = \emptyset$ and $\mathrm{PM}(T_{<\alpha}, \mathcal{P}', \mathcal{O}')$ where

$$\mathcal{O}' = \mathcal{O}_{<\alpha} \cup \{P \in \mathcal{L}_{<\alpha} \mid P \in \mathcal{C}\} \cup \bigcup_{i=0}^m \mathcal{N}(\mathcal{T}_i), \quad \text{and}$$

$$\mathcal{P}' = \mathcal{P}_{<\alpha} \cup \bigcup_{i=0}^m \{\mathit{lab}(N) \mid N \text{ is a leaf in } \mathcal{T}_i \text{ with } \ell(\mathit{lab}(N)) < \alpha\} \cup \{P \in \mathcal{L}_{<\alpha} \mid \overline{P} \in \mathcal{C}\}.$$

Proof ($\rightarrow$). Let $M \subseteq \mathcal{L}_{\leq\alpha}$ be a perfect model of $T_{\leq\alpha}$ satisfying $\mathrm{PM}(T_{\leq\alpha}, \mathcal{P}, \mathcal{O})$. For each $P \in \mathcal{P}_\alpha$, let $\mathcal{T}_P$ be a cyclic tree in T^*_α satisfying the conditions of Lemma 7.3.5.

Since M and $\mathcal{O}_\alpha \cup \bigcup\{\mathcal{O}(\mathcal{T}_P) \mid P \in \mathcal{P}_\alpha\}$ are disjoint, we may (by Lemma 7.3.2) find a cover $\mathcal{C}$ of $\mathcal{O}_\alpha \cup \bigcup\{\mathcal{O}(\mathcal{T}_P) \mid P \in \mathcal{P}_\alpha\}$ in T_α such that $\{P \in \mathcal{L} \mid P \in \mathcal{C}\} \cap M = \emptyset$, and $\{P \in \mathcal{L}_{<\alpha} \mid \overline{P} \in \mathcal{C}\} \subseteq M$. This then proves conditions (a), (b) and (c).

($\leftarrow$). Let $(\mathcal{T}_i \mid 0 \leq i \leq m)$ be a sequence of cyclic trees satisfying conditions (a), (b) and (c). Let $M \subseteq \mathcal{L}_{<\alpha}$ be a perfect model of $T_{<\alpha}$ containing $\mathcal{P}'$ and disjoint from $\mathcal{O}'$.

Let $M^* = M \cup (\mathcal{L}_\alpha - \{P \in \mathcal{L}_\alpha \mid P \in \mathcal{C}\})$, where $\mathcal{C}$ is given in condition (c). Notice that $M^* \cap (\mathcal{O}(\mathcal{T}_i) \cup \mathcal{N}(\mathcal{T}_i)) = \emptyset$ for each $i \leq m$.

We claim that $M^* \models T_\alpha$. Suppose that $C \in T_\alpha$ with $\mathrm{conseq}(C) \cap M^* = \emptyset$, then $\mathrm{conseq}(C) \subseteq \mathcal{C}$, and hence since $\mathcal{C}$ is a cover, either $\mathrm{antec}(C) \cap \mathcal{C} \neq \emptyset$ or $\overline{\mathcal{N}}(C) \cap \mathcal{C} \neq \emptyset$.

If $P \in \mathrm{antec}(C) \cap \mathcal{C}_\alpha$, then $P \notin M^*$. If $P \in \mathrm{antec}(C) \cap \mathcal{C}_{<\alpha}$, then $P \in \mathcal{O}'$, hence $P \notin M^*$. If $\overline{P} \in \overline{\mathcal{N}}(C) \cap \mathcal{C}$, then $P \in \mathcal{P}' \subseteq M$. Thus $M^* \models C$.

Hence we may find an $M' \subseteq \mathcal{L}_\alpha - \{P \in \mathcal{L}_\alpha \mid P \in \mathcal{C}\}$ such that $M \cup M' \models T_\alpha$ and $\forall M'' \subset M'(M \cup M'' \not\models T_\alpha)$.

We claim that $M \cup M'$ satisfies PM($T_{\leq\alpha}, \mathcal{P}, \mathcal{O}$). Firstly it is clear that $M \cup M'$ is a perfect model of $T_{\leq\alpha}$, and that $M \cup M'$ is disjoint from $\mathcal{O}$. Also $\mathcal{P}_{<\alpha} \subseteq \mathcal{P}' \subseteq M$, thus to show that $\mathcal{P} \subseteq M \cup M'$, it suffices, by condition (a) to show that $\bigcup_{i=0}^{m} Pred(\mathcal{T}_i) \subseteq M \cup M'$. Suppose not, and let $i_0 = \min\{i \leq m \mid Pred(\mathcal{T}_i) \not\subseteq M \cup M'\}$.

Pick a predicate node $N \in \mathcal{T}_{i_0}$ such that $lab(N) \notin M \cup M'$ and $\forall N'' > N\,(lab(N'') \in M \cup M')$. As in the proof of Theorem 5.6.2, we may construct a sequence $N = N_0 > RN_{C_0} > N_1 > RN_{C_1} > \ldots$ such that for each k, $lab(N_k) \notin M \cup M'$. Suppose that N_k is given, and has child RN_{C_k}. Then

$$\text{conseq}(C_k) \subseteq (\text{conseq}(C_k) - \text{CYC}(N_k)) \cup \text{CYC}(N_k) \subseteq \mathcal{O}(\mathcal{T}_{i_0}) \cup \text{CYC}(N_k).$$

Moreover $M \cup M' \subseteq M^* \subseteq \mathcal{L} - (\mathcal{N}(\mathcal{T}_{i_0}) \cup \mathcal{O}(\mathcal{T}_{i_0}))$ and $\text{CYC}(N_k) \subseteq \{lab(N_i) \mid i \leq k\} \subseteq \mathcal{L} - (M \cup M')$. Thus we may find some antecedant of C_k which is not in $M \cup M'$.

But then the branch terminates in some node N_j such that $\ell(lab(N_j)) < \alpha$, and hence $lab(N_j) \in \mathcal{P}' \subseteq M$; a contradiction. ∎

The above results are extended further in [12], where a first order query answering mechanism is presented for databases under the perfect model semantics. This mechanism uses both cyclic trees and deduction trees [10].

7.5 The complexity of PM.

When working with the semantics defined by minimal models, the problem of deciding whether $T \not\models P$ (i.e., whether T has a (minimal) model which does not contain P) is $\sum_1^P$ - complete, whereas the corresponding problem for negative data (i.e., whether T has a minimal model containing P) is more difficult, i.e., $\sum_2^P$ - complete.

When we employ the perfect model semantics, the handling of both positive and negative data have the same complexity.

Firstly it is clear that PM($T, \{P\}, \emptyset$) is $\sum_2^P$ - hard from Proposition 7.2.3 and the results of [3] (see Section 2). We can prove a similar hardness result for the handling of positive data.

7.5.1 Theorem ($\sum_2^P$ - hardness). Let T be a deductive database in $\mathcal{L}$ and $P \in \mathcal{L}$, then T has a perfect model containing P iff $T \cup \{\overline{P} \to X\}$ has a perfect model which does not contain X, where X is a predicate not in $\mathcal{L}$.

It is clear from Theorem 7.2.2 that the existence of a perfect model (satisfying PM(T, $\mathcal{P}$, $\mathcal{O}$)) can be expressed via a $\sum_2$ statement, and hence the problem of determining PM is $\sum_2^P$ - complete.

7.6 Horn databases.

In this section we consider the special case of databases in which the intension is Horn. That is, for each rule C in INT(T) $= T - T_0$, $|\text{conseq}(C)| = 1$. Such databases are of importance in view of their relationship to logic programming.

7.6.1 Theorem. If $0 < \alpha \leq n-1$ and $M \subseteq \mathcal{L}_{<\alpha}$ is a perfect model of $T_{<\alpha}$, then there is a unique set $M' \subseteq \mathcal{L}_\alpha$ such that $M \cup M'$ is a perfect model of $T_{\leq\alpha}$.

The proof is trivial, since M' is the closure of M under the Horn rules in T_α. We denote this closure by $cl(T_\alpha, M) = \{P \in \mathcal{L}_\alpha \mid T_\alpha \models \bigwedge\{R \mid R \in M\} \wedge \bigwedge\{\overline{R} \mid R \in \mathcal{L}_{<\alpha} - M\} \to P\}$.

7.6.2 Theorem. A set $M \subseteq \mathcal{L}$ is a perfect model of T iff M_0 is a minimal model of EXT(T) $= T_0$, and for each $0 < \alpha \leq n-1$, $M_\alpha = cl(T_\alpha, M_{<\alpha})$.

7.6.3 Definition. Let $M \subseteq \mathcal{L}_0 = \text{EXT}(\mathcal{L})$ and $0 < \alpha \leq n-1$, then define $cl^*(T_{\leq\alpha}, M) \subseteq \mathcal{L}_{\leq\alpha}$ by

$$cl^*(T_{\leq\alpha}, M)_\beta = cl(T_\beta, cl^*(T_{\leq\alpha}, M)_{<\beta})$$

for each $0 < \beta \leq \alpha$. In the case when $\alpha = n-1$, we write $cl^*(T, M)$ for $cl^*(T_{\leq\alpha}, M)$.

7.6.4 Corollary. Let $M \subseteq \mathcal{L}_0$ be a minimal model of T_0 and $0 < \alpha \leq n-1$, then $cl^*(T_{\leq\alpha}, M)$ is the unique perfect model of $T_{\leq\alpha}$ extending M.

Notice that if $\mathcal{T}$ is a weak deduction tree in $T^*_{>0}$, then $\mathcal{O}(\mathcal{T}) = \emptyset$.

7.6.5 Theorem. Let $\mathcal{P}$ and $\mathcal{O}$ be disjoint subsets of $\mathcal{L}_{\leq\alpha}$, then PM($T_{\leq\alpha}$, $\mathcal{P}$, $\mathcal{O}$) iff there is a sequence $(\mathcal{T}_i \mid 0 \leq i \leq m)$ of cyclic trees in $\bigcup_{\beta=1}^{\alpha} T^*_\beta$ such that:

(a) $\mathcal{P}_{>0} \subseteq \bigcup_{i=0}^{m} Pred(\mathcal{T}_i) \subseteq \mathcal{L} - \bigcup_{i=0}^{m} \mathcal{N}(\mathcal{T}_i)$,

(b) if N is a predicate leaf node in $\mathcal{T}_i$, then $lab(N) \in \mathcal{L}_0 \cup \bigcup_{j<i} Pred(\mathcal{T}_j)$, and

(c) there is a minimal model M of T_0 containing $\bigcup_{i=0}^{m}\{lab(N) \mid N$ is a leaf in $\mathcal{T}_i, lab(N) \in \mathcal{L}_0\} \cup \mathcal{P}_0$ and such that $\mathcal{O} \cup \bigcup_{i=0}^{m} \mathcal{N}(\mathcal{T}_i)$ is disjoint from $cl^*(T_{\leq\alpha}, M)$.

Proof ($\rightarrow$). Let $M' \subseteq \mathcal{L}_{\leq\alpha}$ be a perfect model of $T_{\leq\alpha}$ containing $\mathcal{P}$ and disjoint from $\mathcal{O}$. For each $P \in \mathcal{P}_{>0}$, we may (as in the proof of Theorem 7.3.5) construct a cyclic tree $\mathcal{T}_P$ for P in $\bigcup_{\beta=1}^{\alpha} T^*_\beta$ such that $Pred(\mathcal{T}_P) \subseteq M' \subseteq \mathcal{L} - \mathcal{N}(\mathcal{T}_P)$ and in which each leaf node is a predicate node N with $lab(N) \in \mathcal{L}_0$. But then $M = M' \cap \mathcal{L}_0$ is a minimal model of T_0 with $cl^*(T_{\leq\alpha}, M) = M'$, and hence condition (c) is satisfied.
($\leftarrow$). If M is a minimal model of T_0 satisfying condition (c), then (as in the proof of Theorem 7.4.2) it is easy to show that $cl^*(T_{\leq\alpha}, M)$ contains $\bigcup_{i=0}^{m} Pred(\mathcal{T}_i)$, and hence satisfies PM($T_{\leq\alpha}$, $\mathcal{P}$, $\mathcal{O}$). ∎

Finally we show that for Horn databases, PM is $\sum_1^P$ - complete.

7.6.6 Theorem ($\sum_1^P$ - hardness). Let $\Psi = \exists(P_1, P_2, \ldots, P_m)\,\Phi$, where Φ is propositional in $\{P_1, P_2, \ldots, P_m\} \subseteq \mathcal{L}$. Let

$$T(\Psi) = \{DD(\Phi)\} \cup \{P_i \vee P'_i \mid i \leq m\} \cup \{\overline{A_k} \rightarrow Y\} \cup \{\overline{Y} \rightarrow X\},$$

where A_k = Pred(Φ) and X and Y are predicates not occurring in $\mathcal{L}' \cup \{A_1, A_2, \ldots, A_k\}$ (cf. Section 1.6). Define $\ell(P) = 0$ iff $P \in \{P_i \mid i \leq m\} \cup \{P'_i \mid i \leq m\}$.

Then Ψ is true iff X is contained in some perfect model of $T(\Psi)$.

Proof ($\rightarrow$). Let v be a valuation on $\{P_i \mid i \leq m\}$ satisfying Φ, and $M = \{P_i \mid i \leq m, P_i \in v\} \cup \{P'_i \mid i \leq m, \overline{P}_i \in v\}$. Let $M' = cl^*(T(\Psi), M)$, then by Theorem 1.10, $A_k \in M'$, thus $Y \notin M'$ and $X \in M'$.
($\leftarrow$). Let M be a minimal model of $T(\Psi)_0 = \{P_i \vee P'_i \mid i \leq m\}$, then $|M \cap \{P_i, P'_i\}| = 1$ for each $i \leq m$. If $X \in cl^*(T(\Psi), M)$, then $Y \notin cl^*(T(\Psi), M)$, and thus $A_k \in cl^*(T(\Psi), M)$.

By Theorem 1.10, $\{P_i \mid i \leq m, P_i \in M\} \cup \{\overline{P}_i \mid i \leq m, P'_i \in M\}$ is a valuation satisfying Φ. ∎

7.6.7 Definition. Let $0 < \alpha \leq n-1$ and $M \subseteq \mathcal{L}$, then a predicate $P \in M_\alpha$ is (r, α) - *witnessed* in M (where $r \geq 0$) iff there is a rule $C \in T_\alpha$ such that $\text{conseq}(C) = \{P\}$, $\text{antec}(C) \subseteq M$, $\mathcal{N}(C) \cap M = \emptyset$ and each $R \in \text{antec}(C) \cap \mathcal{L}_\alpha$ is (s, α) - witnessed in M for some $s < r$.

7.6.8 Theorem. A model $M \subseteq \mathcal{L}$ of T is a perfect model iff M_0 is a minimal model of T_0 and for each $0 < \alpha \leq n-1$, each $P \in M_\alpha$ is (r, α) - witnessed in M for some $r \leq |\mathcal{L}_\alpha|$.

Proof $(\rightarrow)$. For each $r \leq |\mathcal{L}_\alpha|$ let $M_\alpha^r = \{P \in M_\alpha \mid P \text{ is } (r, \alpha) - \text{witnessed}\}$, then we may find an $r \leq |\mathcal{L}_\alpha|$ such that $M_\alpha^r \subseteq \bigcup_{s<r} M_\alpha^s$. It is then easy to see that $M_{<\alpha} \cup \bigcup_{s<r} M_\alpha^s \models T_\alpha$. The result then follows from Theorem 7.2.2.

$(\leftarrow)$. Suppose that $M' \subset M_\alpha$ and $M_{<\alpha} \cup M' \models T_\alpha$. Pick $P \in M_\alpha - M'$ such that P is (r, α) - witnessed and $\{Q \in M_\alpha \mid \exists s < r(Q \text{ is } (s, \alpha) - \text{ witnessed})\} \subseteq M'$.

If C is a rule in T_α which (r, α) - witnesses P, then $\text{conseq}(C) = \{P\}$, $\text{antec}(C) \subseteq M_{<\alpha} \cup M'$ and $\mathcal{N}(C) \cap (M_{<\alpha} \cup M') = \emptyset$. Hence, $P \in M'$. ∎

The $\sum_1^P$ - completeness of PM then follows as in Section 2.2.

CONCLUSIONS.

An ability to compute minimal and perfect model membership is necessary when query processing requires the handling of negative goals and subgoals. We have seen that problems related to minimal models are computationally hard, especially for recursive databases. A number of special cases have been identified (Sections 2, 3.3, 3.5 and 7.6) in which the complexity is reduced.

We have also examined the structure of minimal and perfect models, and in particular shown that such models can be characterised by cyclic trees. Such trees can easily be constructed (Section 5.3) via a terminating construction, and the branch length within such trees is polynomially bounded. For first order databases such trees can be constructed directly at the first order level (Section 6), although certain constraints on the database appear to be needed. Such tree constructions can be used to compute minimal model membership, to test models for minimality, and to compile the GCWA.

Our results relating cyclic trees to perfect models (Section 7) are extended in [12] to provide a first order (function free) terminating query processing method for indefinite stratified databases under the perfect model semantics. These methods can also be modified to handle weaker forms of the GCWA [13, 14, 21].

References

[1] P. Chisholm, G. Chen, D. Ferbrache, P. Thanisch and M.H. Williams, Coping with indefinite and negative data in deductive databases: A survey, *Data Knowledge Engineering* 2 (1987) 259–284.

[2] J. Chomicki and V. S. Subrahmanian, The generalised closed world assumption is $\prod_2^0$ - complete, *Information Processing Letters*, 34 (1990) 289 - 291.

[3] T. Eiter and G. Gottlob, Propositional circumscription and extended closed world reasoning are $\prod_2^P$ - complete, *Theoretical Computer Science*, 114 (1993) 231 - 245.

[4] H. Gallaire, J. Minker and J.M. Nicolas, Logic and databases: A deductive approach, *ACM Comput. Surveys* 16 (1984) 153–185.

[5] J. Grant and J. Minker, Answering queries in indefinite databases and the null value problem, *Adv. Comput. Res.* 3(1986) 247–267.

[6] L. J. Henschen and A Yahya, Deduction in non-Horn databases, *J. Automated Reasoning*, 1 (1985), 141-160.

[7] L. J. Henschen and H. Park, Compiling the GCWA in indefinite databases, in J. Minker, ed., Foundations of Deductive Databases, pp 395-438, (Morgan Kaufmann, Washington, 1988).

[8] K. Hőrnig and W. Bibel, Improvements of a tautology testing algorithm, in: D. Loveland (Ed.), Proceedings of the 6th Conference on Automated Deduction, Lecture Notes in Computer Science, vol. 138 (Springer, Berlin, 1982), 326-341.

[9] C.A. Johnson, Handling indefinite and negative data in a deductive database, *Data and Knowledge Engineering* 6 (1991) 333–348.

[10] C.A. Johnson, Top down deduction in indefinite deductive database, Computer Science technical report TR93-08 (Keele University, UK). Also published in: F. Bry ed., Proceedings of the 1993 Journées Bases de Données Avancées, Toulouse, (INRIA, France), 119-138.

[11] C.A. Johnson, Deduction trees and the view update problem in indefinite deductive database, submitted, *J. Automated Reasoning.*

[12] C. A. Johnson, Query processing in indefinite stratified databases, Computer Science technical report TR95-14 (Keele University, UK).

[13] Q. Kong, M. H. Williams and G. Chen, The indefinite closed world assumption, Data and Knowledge Engineering, vol. 12 (1994), 297-311.

[14] J. Lobo, J. Minker and A. Rajasekar, Weak generalised closed world assumption, J. Automated Reasoning, vol. 5 (1989), 293-307.

[15] J. Lobo, J. Minker, and A. Rajasekar, Foundations of Disjunctive Logic Programming, (MIT Press, Cambridge, Massachusetts, 1992).

[16] J. Minker, On indefinite databases and the closed world assumption, *Proc. 6th Conf. on Automated Deduction, Lecture Notes in Computer Science* 138 (Springer, Berlin, 1982) 292–308.

[17] S. Naqvi, Some extensions to the closed world assumption in databases, *Proc. International Conference on Database Theory, Rome* (1986).

[18] C. Papadimitriou and M. Yannakakis, The complexity of facets (and some facets of complexity), *Journal of Comput. System Sci.*, 28 (1984) 244 - 259.

[19] T. Przymusinski, On the semantics of stratified deductive databases, in: J. Minker (Ed.), Foundations of Deductive Databases and Logic Programming (1986).

[20] R. Reiter, On closed world databases, in: H. Gallaire and J. Minker, eds., *Logic and Databases* (Plenum Press, New York, 1978) 55–76.

[21] K. A. Ross and R. W. Topor, Inferring negative information from disjunctive databases, J. Automated Reasoning, vol. 4 (1988), 397-424.

[22] J. C. Shepherdson, Negation in logic programming, in J. Minker, ed., Foundations of Deductive Databases, pp 19-88, (Morgan Kaufmann, Washington, 1988).

[23] R. Sommerhalder and S.C. van Westrhenen, The Theory of Computability (Addison Wesley, 1988).

[24] L. Stockmeyer, The polynomial time hierarchy, *Theoretical Computer Science*, 3 (1977), 1-22.

[25] K. Wagner and G. Wechsung, Computational Complexity (Reidel, 1985).

[26] K. Wagner, More complicated questions about maxima and minima, and some closures of NP, *Theoretical Computer Science*, 51 (1987), 53-80.

[27] C. Wrathall, Complete sets and the polynomial time hierarchy, *Theoretical Computer Science*, 3 (1977), 23-33.