Computing Protected Circumscription

Jack Minker[1,2] and Donald Perlis[1]

Department of Computer Science[1]
University of Maryland

University of Maryland Institute[2]
for Advanced Computer Studies

College Park, MD 20742

ABSTRACT: This paper deals with computing circumscription in the case of Horn data with additional protection (indefinite data), an intermediate investigation between Reiter's result on predicate completion and Lifschitz's efforts to make general (formula) circumscription more efficient as a computational tool. Reiter has shown a close tie between McCarthy's circumscription and Clark's predicate completion. Here we investigate a similar tie between an extended version of circumscription involving protected data, and an extended version of predicate completion. When we have a fully ground atomic protected theory, we show that an extension to the relational algebra can be used to obtain all (and only) correct answers. When general Horn axioms are added to the protected theory, we show that Horn axioms also can be used to compute sound answers; however, some correct answers will not be found.

1. Introduction

During the past several years, a number of notions have been developed surrounding the topic of non-monotonic reasoning. A basic aspect of non-monotonic reasoning is that of computing answers to queries with negated atoms. Using first-order logic requires that the theory contain negated data, otherwise queries that contain negated atoms will return with the answer unknown. First-order logic is monotonic in that when a consistent set of axioms is given from which one can prove a statement S, and a new statement B is added to the set of axioms and B is consistent with

S, then one can still prove B. Adding negated statements to a database may overwhelm any practical system. It would, therefore, be of interest to consider systems which do not contain negated data and contain some rule that permits one to compute answers when negated information is requested.

A number of concepts have been investigated with respect to handling negation in non-monotonic systems. In logic programming applications, negation is handled by failure to prove the positive atom. Clark [1978] has shown that negation by failure corresponds to providing the "only if" statements to the "if" statements represented by the theory. It is interesting to note that the first-order theory with "only if" statements becomes cumbersome to compute answers. The non-monotonic theory represented by "negation as failure", deals only with Horn clauses, from which it is relatively easy to compute answers. McCarthy [1980] has introduced the notion of circumscription to handle negation. When the theory is Horn, and there are a finite number of constants in the domain, then circumscription and "negation by failure" are closely related [Reiter 1982]. Under other conditions it may not be easy to compute answers to arbitrary queries in circumscribed theories. If, for example, the theory is non-Horn, then the usual concept of "negation as failure" does not apply and must be replaced by the "generalized closed world assumption" (or "generalized failure by negation") as developed by Minker [1982]. In such a theory it is difficult to compute answers, as may be seen by the work of Grant and Minker [1984], Henschen and Yahya [1984], and Bossu and Siegel [1985].

The theory of circumscription as described by McCarthy [1980] does not cover the topic of protected circumscription [Minker and Perlis 1984b]. In circumscription, the object is to circumscribe a predicate so that only that which is known to be true will be true and that which is not know to be true will be circumscribed from being true and in fact will be considered false. In some situations we are interested in circumscribing a predicate except that it is desired not to conclude that the predicate is false for certain exceptional values of the argument to the predicate. McCarthy's [1984] later version of circumscription, called *formula* circumscription, does subsume protected circumscription, but it is not clear that Reiter's work extends to allow one to compute efficiently in this case. We note that Lifschitz [1984] has nonetheless taken major strides toward reducing formula circumscription to a more tractable form.

In this paper we investigate computational issues with respect to protected circumscription where the theory is Horn with some exceptions where the user specifies that he does not want to conclude that some values are either true or false. In section 2 we discuss the concept of protected data and propose an algorithm that can be implemented readily to answer queries when the theory consists entirely of ground atomic formulas and formulas that denote protection under the closed world assumption (Reiter [1978], Minker and Perlis [1984b]). We then formulate a logic representation of the same problem and prove that the results obtained in the two theories are equivalent. In section 3 we review the notion of protected circumscription and show that the results for the above ground case when using the logic representation or the concept of protected circumscription are equivalent. In section 4 we consider the case of deductive Horn databases with protection. That is, we are no longer concerned only with the ground case. Here we propose an extension to the algorithm presented in section 2. We show that the results obtained in computing answers to queries in the case of the algorithm and the results from protected circumscription are not equivalent. We then show that a suitable "preparation" of a database will prevent the algorithm from producing any answers that contradict circumscription, and also will produce all correct answers to queries in certain forms.

2. Protected Data

2.1 The concept of protected data

Consider a database that consists of the following data:

$$\{R(a),R(b),S(a),S(c)\}$$

There are two one-place predicates in the database: R and S. In such a database augmented by the closed world assumption, it is possible to prove not only the four (positive) atoms, but also the negative literals ¬R(c) and ¬S(b). We wish to investigate databases that contain a form of null value in which it is not known whether a particular atom is true or false. We refer to such unknown data as *exceptional*, and the process of representing this situation as *protection*.

Consider the above database where R(c) is to be exceptional, that is, its truth or falsity is unknown. We represent this as follows: {R(a), R(b), ER(c), S(a), S(c)}. By ER(c), we mean that it is unknown whether R(c) is true or false; that is ´c´ is protected in the relation R.

In this first-order theory augmented by a closed-world assumption, it is possible to compute R(a), R(b), ER(c), S(a), and S(c). We desire that it be possible to compute neither R(c) nor ¬R(c), since ´c´ is protected for R. However, with the closed world assumption, both ¬R(c) and ¬S(b) are provable as well.

We refer the reader to [Zaniolo 1984] for an excellent discussion of the problem of indefinite data. We note that our concern here differs from that of Zaniolo, in that he is not concerned with the problem of negating conclusions when possible (the closed world assumption). He focusses instead on what he calls the problem of defining "a lower bound: the set of objects which, on the basis of available information, can be concluded to satisfy Q for sure..." The dual "upper bound" problem, in his words, "of preserving the closed world assumption when dealing with incomplete databases" is one we address here.

2.2 Algorithm for computing with protection

Again, consider the database: {R(a), R(b), ER(c), S(a), S(c)}. If we calculate as follows using this database: when one wants to compute the negation of an atom, one takes the entire domain, subtracts out those constants that satisfy the positive atom and also subtracts out the protected constants, then the constants remaining may be assumed to satisy the negation of the atom. Let $|W|$ denote the computation of a well formed formula by this method, then for the above example we have:

$$|\neg S(x)| = \{a, b, c\} - |S(x)| - |ES(x)|.$$

Since the atom S is not protected, $|ES(x)| = \emptyset$, and $|S(x)| = \{a, c\}$, we obtain

$$|\neg S(x)| = \{b\}.$$

In addition, if we calculate $|\neg R(x)|$ in the same way, since ´c´ is protected for R, we obtain $|\neg R(x)| = \emptyset$. That is, we cannot prove $\neg R(c)$, and hence, ´c´ is protected for R since we can also not prove R(c). Thus, the two methods of computation just described yield the same results for this example. We shall show later that the algorithm computes correct answers for the case of relational databases.

Our algorithm, which effectively extends relational algebra to handle protected data, is as follows:

Protected Relational Algebra (PRA) Algorithm for Answering Queries With Protected Data: Let A be a finite set of ground atomic formulas, allowing exceptions (protected data), but without function letters. Let PRA be the algorithm defined below, which begins with a query Q (in conjunctive normal form) and returns a set $|Q|$ of (tuples of) constants from the universal set W of all constant symbols in A (the Herbrand Universe of A). A protected (or exceptional) atom P(a) is indicated by the atom EP(a) in A.

Protected Relational Algebra Algorithm (PRA): input Q(x), output $|Q|$ where x is a vector $x_1,...,x_n$, **W** is W^n, and

if Q is atomic then $|Q| = \{x \in \mathbf{W} : Q(x) \in A\}$;

if Q is $\neg$B then $|Q| = \mathbf{W} - |B| - |EB|$, (B is atomic);

if Q is B & C then $|Q| = |B| \cap |C|$;

if Q is B v C then

if B v C is a tautology then $|Q| = \mathbf{W}$

else $|Q| = |B| \cup |C|$.

The manner in which negation is handled in the protected relational algebra is referred to as the *protected closed-world assumption.* Note that PRS can be applied to any wff Q, but if Q is not in conjunctive normal form then tautologies may not be correctly identified. For instance, the wff

$$[P(x) \ \& \ Q(x)] \ v \ [-P(x) \ v \ -Q(x)]$$

which is in disjunctive normal form, will not produce via PRA all tuples of **W** even though it is a tautology.

In the following sections, we show that the algorithm produces the same answers as two other approaches, which we call the logic representation and protected circumscription, in the ground function-free case.

2.3 Logic Representation

In the case of (Horn) databases we have a generalization of the idea of Clark [1978] who, when discussing negation as failure, showed that an ´if and only if´ condition was its analogue. For example, if P(a) and P(b) are known and we do not care about c or d, then we would write

$$(x = a) \ v \ (x = b) \ \leftrightarrow \ P(x).$$

If and only if conditions of this form are referred to as *completion axioms*. Now, if one wants to protect c while leaving d unprotected, our solution is simply to place (x = c) on both the right and left hand sides of the above formula, to obtain

$$(x = a) \ v \ (x = b) \ v \ (x = c) \ \leftrightarrow \ P(x) \ v \ (x = c).$$

If and only if conditions of this form we will call *protected completion axioms*. For later reference, note that we can re-write this as a conjunction of two formulas and then remove tautologies:

(1) $(x=a) \vee (x=b) \vee (x=c) \rightarrow P(x) \vee (x=c)$

(2) $(x=a) \vee (x=b) \vee (x=c) \leftarrow P(x) \vee (x=c)$

and then

(3) $(x=a) \vee (x=b) \rightarrow P(x)$

(4) $P(x) \rightarrow (x=a) \vee (x=b) \vee (x=c)$

(here we assume distinct constants stand for distinct entities). If the original theory, which here we can take to be $x=a \vee x=b \,.\!\rightarrow P(x)$, is denoted A, then A augmented by the protected completion axioms together with the unique names hypotheses [Reiter 1980] alluded to above--that distinct constants stand for distinct entities--we refer to as the logic representation, LR(A), of A.

2.4 Equivalence of Algorithm and Logic Representation

We will prove that the algorithm PRA and the logic representation LR(A) are equivalent, for queries of the form $\{x \mid B(x)\}$, where B is a quantifier free formula in conjunctive normal form. The following theorem can now be shown.

Theorem 1: $a \in |B|$ iff $LR(A) \vdash B(a)$ for all formulas B in conjunctive normal form, where A is an atomic function-free theory.

Proof: We proceed by induction on the length of B.

(i) If B is atomic then by definition of PRA, $a \in |B|$ implies $LR(A) \vdash B(a)$, since in fact then $A \vdash B(a)$. On the other hand, if $LR(A) \vdash B(a)$, then $A \vdash B(a)$ since A is a ground theory and the completion-plus-protection yields no new positive atoms. Then $a \in |B|$.

(ii) B is $\neg$P, where P is atomic. If a $\in$ $|B| = |\neg P|$ then a is neither in $|P|$ nor $|EP|$. But then A $\nvdash$ P(a) and as above then LR(A) $\nvdash$ P(a). Moreover, P(a) is not protected, so LR(A) $\vdash$ $\neg$P(a). On the other hand, if LR(A) $\vdash$ $\neg$P(a) then a is neither in $|EP|$ nor $|P|$ (the latter since for atomic wffs provability is the same as algorithmic validity). So we get a $\in$ $|\neg P|$.

(iii) B is P & Q. But then a $\in$ $|B|$ iff a $\in$ $|P|$ int $|Q|$ iff LR(A) $\vdash$ P(a) & Q(a) [by inductive hypothesis].

(iv) B is P v Q. If a $\in$ $|B|$ and B is not a tautology, then either a$\in$ $|P|$ (in which case LR(A) $\vdash$ P(a)) or a $\in$ $|Q|$ (in which case LR(A) $\vdash$ Q(a)). So in either case, LR(A) $\vdash$ P(a) v Q(a). If B however is a tautology, then we have LR(A) $\vdash$ P(a) v Q(a) also.

On the other hand, if LR(A) $\vdash$ P(a) v Q(a), and if P v Q is a tautology, then again by PRA a $\in$ $|B|$. But if P v Q is not a tautology, we proceed as follows. First consider the case in which P and Q consist of positive atoms. We want to show that at least one of P(a) and Q(a) is a theorem of LR(A). But a proof of P(a) v Q(a) by refutation-resolution can proceed only by producing negated atoms of the form a $\neq$ c_k, since $\neg$P(a) and $\neg$Q(a) will resolve only with clauses of the forms x=c_1 v x=c_2 v ... $\rightarrow$ P_i(x) and x=c_1 v x=c_2 v ... $\rightarrow$ Q_i(x). In order to fail, one of these negated atoms must fail, i.e. some a$\neq$$c_k$ must resolve with a=c_k; but this occurs on a single branch from either $\neg$P(a) or $\neg$Q(a), hence either P(a) is proven or Q(a) is proven. Then by the inductive hypothesis a is either in $|P|$ or $|Q|$.

Next, if Q, say, is a single negated atom (for instance if Q is $\neg$R), then we have LR(A) $\vdash$ P(a) v $\neg$R(a), where R is an atom and P is a disjunction of literals. Now a refutation-resolution proof derives a failure from $\neg$P(a) and R(a). But the latter can resolve only with protected completion clauses of the form Rx $\rightarrow$ x=c_1 v x=c_2 v ... v x=c_n, yielding a=c_1 v a=c_2 v ... v a=c_n. Now either these disjuncts are all false, or exactly one of them is true, since we take $c_1, c_2, ..., c_n$ to be distinct elements. In the former case, then all such clauses disappear and failure comes directly from R(a) so that LR(A) $\vdash$ ¬R(a). In the latter case, suppose a=c_i is true, and resolves with a unit clause a≠c_i in a branch derived from ¬P(a). Then a≠c_i is already false and so ¬P(a) fails by itself.

¬P(a)　　　　R(a)

a≠c_i　　　　a=c_i

nil

This then completes our proof by induction.

Thus, the protected completion axioms and the protected relational algebra yield the same answers to queries on a ground atomic theory database with protected data. The protected closed-world assumption in the Horn theory avoids the potentially complicated answering process in the first-order theory augmented by the completion axioms.

Note that when the formula B is not in conjunctive normal form, the situation is rather different. For instance, let the database be EPa and Qa, and let the query be [Pa & Qa] v [¬Pa & Qa]. Intuitively this should be answered "yes" (i.e., true), for it is logically equivalent to Q(a). But since Pa and ¬P(a) are separated into two clauses, PRA will not recognize this, and will not return this answer. Also note that the test for tautologies in PRA in fact is quite trivial in the intended case, namely for wffs B that are in conjunctive normal form; for this amounts to nothing more than determing whether any clause contains two opposite literals, which can be performed in time linear in the length of the clause.

3. Protected Circumscription and Protected Data

3.1 Background on Circumscription

We review briefly the idea of circumscription. Given a predicate symbol P (other than the equality predicate symbol) and a formula A[P] containing P, the circumscription of P by A[P] can be thought of as saying that the P-things consist of certain ones as needed to satisfy A[P] and no more, in the sense that any P-things Z satisfying A[Z] already include *all* P-things:

$$C_A^P[Z]: \quad [A[Z] \,\&\, (x)(Z(x) \rightarrow P(x))] \;\rightarrow\; (y)(P(y) \rightarrow Z(y)).$$

A key example, emphasized by McCarthy, is the following: let A[P] be P(a) v P(b). Let Z_1(x) be x=a and Z_2(x) be x=b. Then from P(a) v P(b) we get that either Z_1 or Z_2 can serve for circumscription, i.e., either P(x) $\rightarrow Z_1$(x) or P(x) $\rightarrow Z_2$(x). Thus either a is the only P-thing, or b is: ¬P(a) v ¬P(b).

The concept of predicate circumscription initially developed by McCarthy [1980] was extended by McCarthy [1984] to include general formulas. Although McCarthy [1980] had a soundness result for predicate circumscription in terms of minimal models, he had no such result or model theory for formula circumscription. Minker and Perlis [1984b] provided a model theory and showed soundness for the intermediate case of protected circumscription. Etherington [1984] has provided a model theory and soundness for formula circumscription, generalizing the previous work. Perlis and Minker [1985] establish a partial converse to all these soundness results, for certain classes of theories; in particular, for theories all models of whose circumscriptions are minimal. One such type of theory is that in which the universe is provably finite.

3.2 Protected Circumscription

As a motivation, suppose a database DB is given, and that as is usual the queries Q that are answerable affirmatively are the ones that are true with respect to an intended real-world model, with the exception of certain queries regarding items that we know have not been specified completely yet in DB. E.g., we may know data is still being gathered on these items, such as, say, incomes of middle-level management in a large company, while all the other entries in DB may be complete. We may wish to reason about DB assuming that all data is known (closed world assumption [Reiter 1978]) except for these incomes. That is, we may wish to protect these data from circumscription.

Here we review a simple syntactic device which will yield the desired result in such cases. We suggest that once A has been selected as appropriate for circumscribing P, and if (perhaps later) it is desired to protect certain things (say, those satisfying the predicate EP) from this process so that circumscription will not be used to show EP-things are not P-things, we can keep the same criteria A, but alter the form of the schema itself. Writing EP_i(x) for protected from circumscription of the atom P_i, and writing T/U(x) for T(x) & ¬U(x), we alter the circumscription schema to read as follows:

Definition: The protected circumscription schema for predicate letters $P_1,...,P_k$ in the axioms A is

$$C[A;P_1,...,P_k;EP_1,...,EP_k]: \quad A[Z_1,...,Z_k] \ \& \ / \ (x)(Z_i/EP_i \rightarrow P_i) \quad \rightarrow \quad / \ (y)(P_i/EP_i \rightarrow Z_i)$$

An example of the algorithm and circumscription is as follows: Let A consist of the atoms

{R(a) R(b) S(a) S(c) ER(c)}

so that R(c) is protected. Then circumscription would allow us to prove ¬S(b), and no other literals than this and those already in A. The algorithm PRA will yield the same result: if we query |R|, we will get {a,b}; |ER| = {c}, and |¬R| yields {a,b,c} - |R| - |ER| = nil. Also, |S| = {a,c}. |ES| = nil, and |¬S| = {a,b,c} - |S| - |ES| = {b}.

If we circumscribe on all predicates at once, which is the form corresponding to predicate completion and to the algorithm PRA, then we use as many protection predicates EP_i as there are original predicates P_i.

3.3 Equivalence of Logic Representation and Protected Circumscription

In this section we show that the logic representation LR(A) of an atomic function-free theory A and its protected circumscription CIRC(A) are equivalent in a precise sense. We employ the notation A |c- B to mean B is provable from A by protected circumscription.

Theorem 2: LR(A) $\vdash$ B iff A |c- B, where A is an atomic function-free theory.

Proof: We simply circumscribe all P in A to derive LR(A), for the left-to-right direction. Let $Pc_1,...,Pc_n$ be the P-atoms in A, and let $Z_P(x)$ be $x=c_1 v ... v x=c_n$. Then circumscribing A with protected atoms $P\delta_1,...,P\delta_m$ yields the completion axioms

$$Px \text{ v } x=\delta_1 \text{ v } ... \text{ v } x=\delta_m \leftrightarrow x=c_1 \text{ v } ... \text{ v } x=c_n \text{ v } x=\delta_1 \text{ v } ... \text{ v } x=\delta_m .$$

Now for the converse. Note that for atomic theories A, a completeness theorem [Perlis & Minker 1985] holds for A: a wff B is true in all minimal models of A iff it is a circumscriptive theorem of A. Now if M is a model of LR(A) then M is a minimal model of A since if (the extension of) any wff P is reduced, M no longer is a model of LR(A), hence no longer a model of the circumscriptive schema (by the first part of the proof). Then all models of LR(A) are minimal, and it follows that if A |c- B then A |c= B whence LR(A) $\models$ B, and finally LR(A) $\vdash$ B.

As a consequence, protected circumscription, the algorithm PRA, and the logic representation LR(A) are equivalent in terms of what conclusions they produce, for theories A that are atomic and function-free. In the following sections, we attempt to extend these results to more complex theories.

4. Deductive Databases and Protected Data

4.1 Protected Relational Algebra for Deductive Horn Databases

Let A be a finite set of definite Horn formulas, allowing exceptions (protected data), but without function letters. We wish to compute answers to queries in this case using an obvious extension to algorithm PRA presented earlier. Let PRAH be the algorithm defined below, which begins with a query Q (in conjunctive normal form) and returns a set $|Q|$ of constants from the universal set W of all constant symbols in A (the Herbrand universe of A). A protected (or exceptional) atom P(a) is indicated by the atom EP(a) in A.

Protected Relational Algebra Algorithm for Horn Theories (PRAH): input Q; output |Q| where

if Q is atomic then $|Q| = \{x \in W: A \vdash Q(x)\}$;

if Q is $\neg B$ then $|Q| = W - |B| - |EB|$; (B will be atomic)

if Q is B & C then $|Q| = |B| \cap |C|$;

if Q is B v C then

if B v C is a tautology then $|Q| = W$

else $|Q| = |B| \cup |C|$.

Note that the only difference between the algorithm PRAH and the algorithm PRA, aside from the fact that A now is a more general kind of theory, is that PRAH employs provability from A rather than simple axiomhood in the atomic case.

4.2 Non-equivalence

The case in which there are general Horn clauses in the theory that contain implication symbols and both the left and right sides are not empty, complicates matters. Consider a simple theory, $A = \{P(a), Q(x) \leftarrow P(x), EP(c)\}$. The corresponding first-order theory corresponding to A obtained using protected circumscription is:

$$CIRC(A) = \{P(x) \vee (x = c) \leftrightarrow (x = a) \vee (x = c), P(x) \leftrightarrow Q(x)\}.$$

If we modify our algorithm as above, to state provability, as the obvious generalization, we do not obtain the same results from the circumscribed theory and the algorithm. Applying the algorithm to A, we obtain $\neg Q(c)$, whereas in CIRC(A), we cannot prove $\neg Q(c)$.

Furthermore, although it is possible to circumvent this difficulty by addition of further protection atoms, the algorithm can fail for some disjunctive queries. For example, the theory, $A' =$

{P(a), Q(x) ← P(x), EP(c), EQ(x) ← EP(x)}, now protects ´c´ for the atom Q, and the corresponding protected theory finds negations correctly. The protected circumscription theory for A´ is given by:

CIRC(A´) =

$$\{P(x) \text{ v } (x = c) \leftrightarrow (x = a) \text{ v } (x = c),\ Q(x) \text{ v } (x = c) \leftrightarrow P(x) \text{ v } (x = c)\}.$$

But now, while CIRC(A´) "computes" the answer to the query ¬P(x) v Q(x) to be {a}, the same answer is not obtained by the use of the algorithm applied to theory A´. That is, PRAH will produce the answer $|\neg P(x) \text{ v } Q(x)| = \emptyset$.

4.3 Soundness of a suitably prepared database

We establish that the appropriate "preparation" of the database will force the extended algorithm to agree with protected circumscription, in the weak sense that all answers to queries given by the algorithm will be consistent with the results of circumscription. The preparation we have in mind is that of adding extra protection to the database, typified in the following situation: if P→Q and EP(a) are in the database, then EQ(a) is added, or alternatively, EP(x)→EQ(x). We will proceed by establishing several lemmas before giving our principal result on this.

Definition: Call a theory A *suitable* if it consists of a finite non-recursive Horn database **H** augmented with a set **E** of ground *protection* clauses of the form EP(a) where P is a predicate symbol, function free, complete in the sense of Perlis and Minker [1985] (i.e., all models of the circumscription of A are minimal).

Throughout we refer the reader to the following example for illustration: Let **H** consist of the clauses

{Fx←Hx, Ga←Ia, Ia←Ba, Ix←Cx&Fx, Ba←Ca&Da&Fa, Fb←Db, Ha},

and let A consist further of **H** together with the protection atom ECa.

As we saw in the examples of non-equivalence of CIRC(A) and PRAH, the algorithm may overdo the CWA in the sense that circumscription treats certain atoms as if protected while PRAH does not. In particular, the illustrated theory just mentioned has Ca explicitly protected, but circumscription will implicitly treat Fa, Ia, and Ga as if protected as well, even though PRAH will not. Thus CIRC(A) will not have ¬Ga as a theorem, while PRAH will yield $a \in \lfloor \neg G \rfloor$. To prevent PRAH from "missing" (and consequently negating) the implicitly protected atoms, we seek to identify which these atoms are, and subsequently to make explicit their protection with a suitable "preparation" of A. The following lemma and "marking algorithm" identify those atoms which are protected, implicitly or explicitly, in circumscriptive deductions.

Lemma 1: Let M be a (protected) minimal model of A, say, M = {Ga,...,Db,...}. Suppose $M \models \gamma$, where γ is an atom (for instance, it is useful to think of γ as the atom Ga). Then if $E\gamma \notin$ **E** there is a formula δ such that $M \models \delta$, $\gamma \leftarrow \delta$ is an axiom of A, $\delta = \delta_1$ ... δ_n, and for i = 1,...,n either $A \vdash \delta_i$ or $\delta_i \in$ **E**.

Proof: Note that by Perlis and Minker [1985], $\neg\gamma$ is not a theorem of CIRC(A). We wish to show, in effect, that γ's presence (truth) in M is allowed only because it is forced by some protected and/or provable atoms δ_i also true in M.

We now describe the formation of a tree that allows us to determine just how γ can be true in M. First, since M is (protected) minimal, then simply removing the atom γ from M will necessarily leave a structure M' that violates an axiom of A unless γ is explicitly protected (i.e., unless $\gamma \in$ **E**). Now if γ is so protected, then we are done. So suppose otherwise.

We let γ be the root of the tree to be described; its immediate children will be all conjunctions δ where $\gamma \leftarrow \delta$ is an axiom of H. (Note that no other axioms of A can be violated in M'.) Now, any such axiom, to be violated, requires δ to be true in M. We therefore repeat the above procedure, continuing the tree construction for each of these children nodes. Whenever we reach a node $\delta_1 ... \delta_n$ for which some $E\delta_i \in \mathbf{E}$, then in the next level of the tree replace δ_i with $E\delta_i$; and whenever a conjunct δ_i is an axiom of H, replace it with $\square$.

Now since A is suitable and hence has a finite number of non-recursive axioms and terms, this procedure must halt in a finite number of steps, yielding a tree T with a branch from γ to a leaf $\square$. By tracing back from $\square$ to γ, we then find the conjuncts that are either theorems of A or are protected in A, and whose conjunction suffices to guarantee γ true in M.

Now if γ is an atom such that neither it nor $\neg\gamma$ is a theorem of CIRC(A), then there is a minimal model of A, say M, such that γ holds in M, and so the lemma just proven applies, and indeed the proof of the lemma already contains an algorithm to produce a "justification" for γ appearing in M, namely, other atoms $\delta_1, ..., \delta_n$ of which some (at least one) are explicitly protected, and the rest are provable. We wish to know which these atoms γ are, so that we can make appropriate modifications in the use of PRAH. For this, we work backwards from all possible $\delta_1, ..., \delta_n$, using the following *marking algorithm* will determine the atoms γ that are effectively protected by circumscription.

Marking Algorithm: First, augment A by the new axiom $P \leftarrow EP$ for all $EP \in \mathbf{E}$. Then for each predicate γ, form a refutation tree with root $\leftarrow \gamma$, using the augmented theory A'. Whenever EP is produced, mark the corresponding axiom in that resolution step by placing a $\square$ around any P on the right side of the axiom; and whenever P is produced, mark the corresponding axiom in that resolution step by placing a $\bigcirc$ around any Q on the right side of the axiom; and whenever this results in an axiom all of whose right conjuncts are marked, including at least one $\square$, then mark also its left side with a $\square$. Call the theory resulting from the original A by adjoining all atoms EP where P is a marked left side, PREP(A), the "preparation" of A. In what follows, the answer set $|P|$ to a query P is understood to be with respect to our extended algorithm PRAH applied to the preparation of A.

Definition: A theory is *suitably prepared* if it is suitable and is the preparation of some theory.

> Lemma 2: Let A be suitably prepared. If CIRC(A) $\nvDash \neg P(a)$ then $a \notin |\neg P|$.
>
> Proof: Assume otherwise: CIRC(A) $\nvDash \neg P(a)$ and yet $a \in |\neg P|$. Then $a \in W - |P| - |EP|$. Now $\neg P(a)$ cannot hold in *all* minimal models since CIRC(A) $\nvDash \neg P(a)$. So P(a) must hold in at least one minimal model. We employ Lemma 1 where γ is P(a), and M is a minimal model in which P(a) holds. Then there exists a formula $\delta = \delta_1 ... \delta_n$ such that $M \vDash \delta$, $\gamma \leftarrow \delta \in A$, and for i=1,...,n, $A \vdash \delta_i$ or $E\delta_i \in \mathbf{E}$. The marking algorithm will place a $\square$ around γ, hence $E\gamma$ will be a theorem of PREP(A), i.e., PREP(A) $\vdash$ EP(a), so $a \in |EP|$. This contradicts $a \in W - |P| - |EP|$.

Now we can prove that if CIRC(A) $\nvDash$ B(a) where B(x) is any formula in conjunctive normal form, then $a \notin |B|$.

> Theorem 3: If A is suitably prepared, then the PRAH algorithm produces only answers B that also are provable by circumscription.
>
> Proof: We use induction on the number of (instances of) predicate symbols in B(a). When B(a) has only one predicate symbol, then either it is an atom, or the negation of an atom. If B(a) is an atom, then circumscription and PRAH both reduce to mere provability in A, so we are done; and if B(a) is a negated atom, Lemma 2 is the result we seek.
>
> So we pass to the case of more than one predicate symbol; then B(a) is either a conjunction or a disjunction of subformulas F and G. If B(a) is F & G, then we have

$$|B| = |F| \cap |G|$$

$$\subseteq \{x: \mathrm{CIRC}(A) \vdash F(x)\} \cap \{x: \mathrm{CIRC}(A) \vdash G(x)\}$$

$$= \{x: \mathrm{CIRC}(A) \vdash F(x)\ \&\ G(x)\}$$

since by the inductive hypothesis PRAH produces as answers a subset of those derivable by circumscription, for queries of less predicate symbols than B(a). Finally, for the case of F v G, a parallel argument suffices using union instead of intersection.

We know of course that the full converse to Theorem 3 is false; however, we can show that a partial converse does hold, giving rise to the following result. Let PRAH' be simply PRAH except in that a negative disjunction ¬P v ¬Q is replaced by ¬R and the database is augmented by the axioms R ← P&Q, P ← R, and Q ← R. [This has the effect of introducing a new predicate letter R that is equivalent to the conjunction P&Q, and will be used in the sequel.] A query is *pure* if it is in conjunctive normal form and each conjunct consists of disjuncts of the same sign, i.e., all positive or all negative; in effect, no implications are present.

Theorem 4: Protected circumscription and PRAH' agree for suitably prepared theories A and all pure queries B.

Proof: We need only show that for pure queries B, proof by circumscription implies a positive answer by PRAH, since the converse is contained in Theorem 3. Our result essentially follows from Lemma 1 and its proof. We consider five cases: pure positive disjunctions P v Q, negated atoms ¬P, pure negative disjunctions ¬P v ¬Q, and pure negative conjunctions ¬P & ¬Q, and mixed conjunctions P & ¬Q. >From these the cases of all pure queries follow.

(i) Pa v Qa: We assume CIRC(A) ⊢ Pa v Qa. If also CIRC(A) ⊢ Pa or CIRC(A) ⊢ Qa, then we are done since for atoms PRAH and CIRC reduce to simple provability. So suppose neither CIRC(A) $\vdash$ Pa nor CIRC(A) $\vdash$ Qa. We will derive a contradiction. Let M be a minimal model of A such that M $\not\models$ Qa. Then M $\models$ Pa. As in the proof of Lemma 1, determine the atoms δ_i, and remove from M both Pa and any δ_i where A $\nvdash$ δ_i and Eδ_i $\in$ **E**, thereby producing a new structure M'; M' will be a model of A by the construction of the tree in the proof of Lemma 1. But M' violates the theorem Pa v Qa, and M' either is minimal or has a minimal submodel [Etherington et al 1984], so we have a contradiction. This shows that when Pa v Qa is a circumscriptive theorem over a suitably prepared theory A, so is one of Pa or Qa, and from this it follows that a $\in$ $|$P v Q$|$.

(ii) $\neg$Pa: If CIRC(A) $\vdash$ $\neg$Pa then a $\notin$ $|$P$|$ (assuming A consistent). Assume a $\in$ $|$EP$|$. Then let M be a minimal model of A such that M $\not\models$ Pa. Add Pa to M, and also all atoms δ such that A $\vdash$ Pa$\rightarrow\delta$, to get a structure M'. Then M' is also a minimal model of A, and M' $\models$ Pa, contradicting CIRC(A) $\vdash$ $\neg$Pa. So a $\notin$ $|$EP$|$. Then we have a $\in$ $|\neg$P$|$ = W - $|$P$|$ - $|$EP$|$.

(iii)$\neg$Pa v $\neg$ Qa: This case we handle simply by adjoining to A the additional axioms R $\leftarrow$ P&Q, P $\leftarrow$ R, and Q $\leftarrow$ R, and working with the query $\neg$R (which PRAH' does), and placing us in case (ii) above.

(iv) $\neg$P & $\neg$Q: This we do directly with intersections as in the algorithm.

(v) Pa & $\neg$Qa: This is also done directly with intersections.

We note that there is an obvious extension to PRAH that remains sound and comes closer to producing all correct answers (with respect to protected circumscription) for suitably prepared databases. This extension, which we might call PRAHX (extended PRAH), simply amounts to one additional source of answers in the case of a disjunctive query B v C; namely, if B v C is neither a tautology nor pure (all positive or all negative literals) then it is equivalent to the Horn clause A$\rightarrow$B (if C is $\neg$A) and the answer is

$$|B| \cup |\neg A| \cup \{x \in \mathbf{W}\colon Ax \rightarrow Bx \in A\}.$$

The additional answers are then those that are already present in the database axioms, if the database is compiled in advance to include all possible implications derivable from A. In general then many additional answers can be obtained this way. Unfortunately, even then not all correct answers can be guaranteed. The following example illustrates this phenomenon.

Let the database A consist of EPa, EQa, Qb, Pa→Qa. Then the query Px v ¬Qx will have no answers as far as PRAHX is concerned, even though protected circumscription yields x=a as an answer. we leave it then as an open question, whether a natural and computationally feasible modification of PRAHX can be sound and complete for protected circumscription and suitably prepared theories.

4.4 PROLOG Implementation

PROLOG provides an implementation of the concept of the closed world assumption, or negation by failure, and thereby provides a valuable tool to implement logic databases. In a logic database one has relational data augmented by general Horn axioms as part of the database.

It is easy to see that a modification to PROLOG will permit an implementation of the protected closed world assumption. It basically follows the way in which negation is handled in PROLOG. If we desire to define $\neg_E$ to be a new negation relative to the protected closed world assumption, we may use the following representation:

(1) $\neg_E$ P ← P &/& fail.

(2) $\neg_E$ P ← EP &/& fail.

(3) $\neg_E$ P ← .

5. Conclusions

We have shown that it is possible to compute answers effectively in the case of protected circumscription when using Horn axioms. This leads to the open question, to what extent Horn theories will be useful replacements of circumscribed theories. There are now at least two cases where one can effectively compute such theories--these are the Horn axioms augmented by Clark's completion axioms [Clark 1978] as shown by Reiter [1982], and the work presented in this paper. In the case of protected theories, we do not obtain all answers for some queries.

It would be of considerable interest to investigate formula circumscription for Horn theories to determine the class of formulae circumscribed that lead to varying degrees of effective computation.

Acknowledgements

We gratefully acknowledge the very helpful comments made to us by Vladimir Lifschitz. This research was supported in part by grants from the following organizations:

AFOSR (AFOSR-82-0303)

NSF (MCS-83-05992)

ARO

Martin Marietta Corporation

References

(1) Bossu, G. and Siegel, P. [1985] Saturation, nonmonotonic reasoning, and the closed-world assumption. Artificial Intelligence, 25 (1), pp. 13-63.

(2) Clark, K. [1978] Negation as failure. In: Logic and Databases, Gallaire, H. and Minker, J. (eds.). Plenum Press, New York, pp. 293-322.

(3) Etherington, D. [1984] Personal communication, Week on Logic and AI, U of Md.

(4) Etherington, D., Mercer, R., and Reiter, R. [1984] On the adequacy of predicate circumscription for closed-world reasoning. University of British Columbia, Dept. of Computer Science, Tech. Report 84-5.

(5) Grant, J. and Minker, J. [1984] Answering queries in indefinite databases and the null value problem. Tech. Report 1374, University of Maryland.

(6) Henschen, L. and Yahya [1985] Deduction in Non-Horn Databases, to appear, J. of Automated Reasoning.

(7) Lifschitz, V. [1984] Some results on circumscription. Workshop on Nonmonotonic Reasoning, Mohonk.

(8) McCarthy, J. [1980] Circumscription--a form of non-monotonic reasoning. Artificial Intelligence, 13 (1,2), pp. 27-39.

(9) McCarthy, J. [1984] Applications of circumscription to formalizing common sense knowledge. Workshop on Nonmonotonic Reasoning, Mohonk.

(10) Minker, J. [1982] On indefinite databases and the closed-world assumption. Lecture Notes in Computer Science, v. 138, pp. 292-308, Springer. (6th Conference on Automated Deduction).

(11) Minker, J. [1983] On theories of definite and indefinite databases. Tech. Report, University of Maryland.

(12) Minker, J. and Perlis, D. [1984a] Applications of protected circumscription. Lecture Notes in Computer Science, v. 170, pp. 414-425, Springer. (7th Conference on Automated Deduction).

(13) Minker, J. and Perlis, D. [1984b] Protected circumscription. Workshop on Nonmonotonic Reasoning, Mohonk.

(14) Perlis, D. and Minker, J. [1985] Completeness results for circumscription. Draft.

(15) Reiter, R. [1978] On closed world databases. In: Logic and Databases, Gallaire, H. and Minker, J. (eds.), Plenum, pp. 55-76.

(16) Reiter, R. [1980] Equality and domain closure in first-order databases. J. ACM, v. 27, pp. 235-249.

(17) Reiter, R. [1982] Circumscription implies predicate completion (sometimes). Proc. AAAI-82, pp. 418-420.

(18) Zaniolo, C. [1984] Database relations with null values. J. Comp. and System Sci., v. 28, pp. 142-166.