ICOT Technical Report: TR-527

TR-527

On Theorem Provers for Circumscription

by

K. Inoue & N. Helft

February, 1990

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

On Theorem Provers for Circumscription

Katsumi Inoue and **Nicolas Helft**
ICOT
Research Center, Mita Kokusai Bldg. 21F
1-4-28 Mita, Minato-ku, Tokyo 108
Japan

March 16, 1990

Abstract

This paper concerns algorithms to answer queries in circumscriptive theories. Two recent papers present such algorithms that are relatively complex: Przymusinski's algorithm is based on MILO-resolution, a variant of ordered linear resolution; Ginsberg's theorem prover uses a backward-chaining ATMS. Because of their different concerns, formalisms, and implementation, it is not clear what their relative advantages are. This paper makes a detailed comparison of these relating them to a logical framework of abduction, explains their intuitive meaning, and shows how the efficiency of both can be improved. Additionally, some limitations of both circumscriptive theorem provers are also discussed.

Keywords: circumscription, abduction, theorem proving, linear resolution.

1 Introduction

Circumscription [McCarthy, 1980; Lifschitz, 1985] is one of the most powerful and well-developed formalizations of nonmonotonic reasoning as it is based on classical predicate logic. Although its formal properties are well investigated, there have been few attempts at effective query answering procedures or implementations for circumscriptive theories.

Recently, Przymusinski [1989] and Ginsberg [1989] have published algorithms to compute circumscription. Ginsberg acknowledges a strong connection between the results presented. However, not much is known about the algorithms' relative advantages and disadvantages.

The goal of this paper is twofold:

1. We further explore the connections between algorithms [Przymusinski, 1989; Ginsberg, 1989], showing that:

   (a) The theoretical results obtained in each of these papers are the same, and both can be re-expressed in a simple, general framework.

   (b) The algorithms presented have different computational properties; we provide a detailed comparison of these.

2. We show how the efficiency of both algorithms can be improved.

Sections 2, 3, and 4 consider the above questions. In Section 5, we further discuss two important problems that arise in Przymusinski's and Ginsberg's approaches and suggest some solutions to them.

2 Comparing the Theorems

We will consider ground theories, that is, first-order theories, without equality, consisting of finitely many ground formulas over the representation language $\mathcal{L}$; these are sufficient to illustrate the comparison between the algorithms of [Przymusinski, 1989; Ginsberg, 1989]. We will use the clausal form representation, and also assume that Unique Names Axioms (UNA) are satisfied for $\mathcal{L}$, as in both algorithms. According to Przymusinski's claims, however, the algorithms are applicable to the first-order case with UNA and equality axioms. Ginsberg adds the domain-closure axiom, which is unnecessary according to the results we shall present which indicate the equivalence between the theoretical results of [Przymusinski, 1989; Ginsberg, 1989]. In Section 5, we will return to the incompleteness problem, which is due to the infinite properties of first-order theories.

We briefly recall a basic property of circumscription, on which the algorithms are based. The predicate symbols of a theory T are divided into three disjoint sets: P, *minimized* predicates; Z, *variables*; and Q, *fixed*. Using this information, some models of T are defined as minimal with respect to the sets P and Z; we say they are (P, Z)-*minimal*. Let $CIRC(T; P; Z)$ be the circumscription of P in T with variable predicates Z. Then, for any formula F, $CIRC(T; P; Z) \models F$ iff $M \models F$ for every (P, Z)-minimal model M of T [McCarthy, 1980; Lifschitz, 1985].

Now, to compare the theoretical results of Przymusinski and Ginsberg, we use the notion of *characteristic clauses* which was introduced by Bossu & Siegel [1985] and was later generalized by Siegel [1987]. This concept can also give the computational aspect of ***abduction***. Informally speaking, characteristic clauses are intended to represent "interesting" clauses to solve a certain problem, and are constructed over a sub-vocabulary of $\mathcal{L}$ called a ***production field***.

Definition 2.1 A *production field* $\mathcal{P}$ is a set of ground literals. A clause C *belongs to a production field* $\mathcal{P}$ if every literal in C belongs to $\mathcal{P}$. The set of clauses that are logical consequences of a set of clauses T and that belong to $\mathcal{P}$ is denoted by $Th_{\mathcal{P}}(T)$.

If R is a set of predicate symbols, we denote by R^+ (respectively R^-) the positive (respectively negative) ground literals with predicates from R, which range over all constants in $\mathcal{L}$. Moreover, $R^+ \cup R^-$ is denoted $R^{\pm}$.

Example 2.2 Suppose that the language $\mathcal{L}$ contains predicates, $bird$, $flies$, ab, and $ostrich$, and that $tweety$ is a constant. Let $\mathcal{P}$ be $\{ab\}^+ \cup \{bird, ostrich\}^{\pm}$. Then, $\neg ostrich(tweety) \vee bird(tweety)$ belongs to $\mathcal{P}$, while $\neg ab(tweety)$ does not.

Definition 2.3 Let T be a set of formulas, F a formula, and $\mathcal{P}$ a production field.

1. The *characteristic clauses* of T are:

$$Carc(T) = \mu\,[\,Th_{\mathcal{P}}(T)\,]^{1},$$

where for a set of clauses Σ, by $\mu\,[\Sigma]$ we mean the set of clauses of Σ not subsumed by any other clause of Σ.

2. The *new characteristic clauses of F with respect to T* are:

$$Newcarc(T, F) = Carc(T \cup \{F\}) - Carc(T)\,,$$

that is, those characteristic clauses of $T \cup \{F\}$ that are not characteristic clauses of T.

Example 2.2 (continued) Let T be

$bird(tweety)$,

$\neg bird(tweety) \vee ab(tweety) \vee flies(tweety)$,

$\neg ostrich(tweety) \vee \neg flies(tweety)$.

In this well-known example, $P = \{ab\}$, $Z = \{flies\}$, and $Q = \{bird, ostrich\}$.

Let us fix as above $\mathcal{P}$ to be $P^+ \cup Q^{\pm}$, that is, positive occurrences of ab, or any occurrence of $bird$ and $ostrich$. Then,

[1] $Carc(T)$ depends on the production field $\mathcal{P}$, and thus a correct notation would be $Carc(T, \mathcal{P})$. As there will be no confusion about $\mathcal{P}$, we simply write $Carc(T)$.

$Carc(T) = \{\, bird(tweety),\ ab(tweety) \vee \neg ostrich(tweety) \,\}$,

$Newcarc(T, bird(tweety)) = \phi$
because $bird(tweety) \in Carc(T)$,

$Newcarc(T, flies(tweety)) = \{\neg ostrich(tweety)\}$
as $\neg ostrich(tweety) \notin Carc(T)$ belongs to $Carc(T \cup \{flies(tweety)\})$.

There is a strong connection between the concept of the new characteristic clauses and a logical account of abductive or hypothetical reasoning defined by such as [Poole *et al.*, 1987; Poole, 1989].

Definition 2.4 Let T be a set of formulas, D a set of ground literals (called the *hypotheses*), and F a closed formula. A conjunction E of elements of D is an *explanation of* F *from* (T, D) if: (i) $T \cup \{E\}$ is satisfiable, and (ii) $T \cup \{E\} \models F$.

An explanation E of F from (T, D) is *minimal* if no proper sub-conjunct E' of E satisfies $T \cup \{E'\} \models F$.

An *extension of* (T, D) is the set of logical consequences of $T \cup \{E\}$ where E is a maximal conjunct of elements of D such that $T \cup \{E\}$ is satisfiable.

We will denote by $\neg \cdot \Sigma$ the set formed by taking the negation of each element in Σ.

Theorem 2.5 Let T, D and F be the same as Definition 2.4. The set of all minimal explanations of F from (T, D) is $\neg \cdot Newcarc(T, \neg F)$, where $\mathcal{P} = \neg \cdot D$.

Corollary 2.6 Let T, D and F be the same as Definition 2.4. There is no extension of (T, D) in which F holds iff there is no explanation of F from (T, D) iff $Newcarc(T, \neg F) = \phi$.

2.1 Przymusinski's Results

Przymusinski's [1989] algorithm is based on the following two theorems developed by Gelfond *et al.* [1989].

Theorem 2.7 [Przymusinski, 1989, Theorem 2.5]
If a formula F does not contain literals from Z, then $CIRC(T; P; Z) \models F$ iff there is no clause E such that (i) E does not contain literals in $Z^{\pm} \cup P^{-}$, and (ii) $T \models \neg F \vee E$ but $T \not\models E$.

Now let us rewrite this theorem using the notation introduced above. Condition (i) means that E belongs to the production field $\mathcal{P} = P^{+} \cup Q^{\pm}$. $T \models \neg F \vee E$ can be written as $T \cup \{F\} \models E$. So we are looking for a clause E belonging to the production field, implied by $T \cup \{F\}$ but not by T alone. This means that $E \in Th_{\mathcal{P}}(T \cup \{F\}) - Th_{\mathcal{P}}(T)$. The theorem requires that such E does not exist. Now, for a set of clauses Σ, $\Sigma = \phi$ iff $\mu[\Sigma] = \phi$. Therefore, by Lemma A.1, it is enough to check whether $Newcarc(T, F)$ is empty or not. That is,

Theorem 2.7 (new version) Let F be a formula not containing literals from Z. Let $\mathcal{P}$ be $P^{+} \cup Q^{\pm}$. Then

$$CIRC(T; P; Z) \models F \quad \text{iff} \quad Newcarc(T, F) = \phi.$$

This formulation helps to understand the intuition underlying the above theorem. We want to know if a query F not involving literals from Z is true or not in the (P, Z)-minimal models of a theory T. Now, every (P, Z)-minimal model of T is defined on interpretations of T by considering differences of extensions of P and equality of extensions of Q, but by ignoring differences of extensions of Z [Lifschitz, 1985]. Therefore, the characteristic clauses of T are representative of those minimal models, in the sense that if adding F to T produces a change (new one) in $Carc(T)$ then the addition of F has produced a change in the minimal models of T as well. The existence of a new characteristic clause of F means that F has altered the minimal models: thus if $Newcarc(T, F)$ is empty, the addition of F has no effect on the minimal models and the circumscriptive theory entails it.

For formulas containing predicates from Z, the following holds:

Theorem 2.8 [Przymusinski, 1989, Theorem 2.6]
Let F be any formula. $CIRC(T; P; Z) \models F$ iff either $T \models F$ or there is a formula G such that (i) G does not contain literals in $Z^{\pm} \cup P^{-}$, (ii) $T \models F \vee G$, and (iii) $CIRC(T; P; Z) \models \neg G$.

Now, $T \models F$ means $T \cup \{\neg F\}$ is unsatisfiable; note that in this case, $Newcarc(T, \neg F)$ would contain only $\Box$ (the empty clause). Condition (i) again means that G belongs to $\mathcal{P} = P^{+} \cup Q^{+}$; condition (ii) can be written as $T \cup \{\neg F\} \models G$; and condition (iii) is equivalent to $Newcarc(T, \neg G) = \phi$ by Theorem 2.7. In this case, the condition $T \not\models G$ is missing in Theorem 2.8; if $T \models G$, however, then $Newcarc(T, \neg G) = \{\Box\} \neq \phi$ holds for satisfiable T. Therefore, condition (ii) together with (iii) further implies that G is of the form of a conjunction of clauses of $Newcarc(T, \neg F)$ [2]. And in condition (iii), if G is $\Box$, then $\neg G$ is the formula *true* and adding it to T produces no new theorem: $Newcarc(T, true) = \phi$. We can now write:

Theorem 2.8 (new version) Let F be any formula, and $\mathcal{P} = P^{+} \cup Q^{\pm}$. Then $CIRC(T; P; Z) \models F$ iff there is a conjunct G of clauses in $Newcarc(T, \neg F)$ such that $Newcarc(T, \neg G) = \phi$.

While this formulation seems simpler than the original one, it still does not provide much insight. We will see it more clearly in Section 2.2, relating it with hypothetical reasoning. Let us review one of Przymusinski's examples with these new concepts.

Example 2.2 (continued) Przymusinski [1989, Example 3.10] shows that $CIRC(T; P; Z)$ does not imply $F_1 = flies(tweety)$ but implies $F_2 = ostrich(tweety) \vee flies(tweety)$. Let us verify these facts.

Adding $\neg F_1 = \neg flies(tweety)$ to T gives

$$Newcarc(T, \neg F_1) = \{ab(tweety)\}.$$

Since adding $\neg ab(tweety)$ to T gives a new characteristic clause, $\neg ostrich(tweety)$, $CIRC(T; P; Z) \not\models F_1$ holds.

[2] In practice, the minimality condition involved by the μ operation is not crucial. See Section 3.1.

Now we add

$$\neg F_2 = \neg ostrich(tweety) \wedge \neg flies(tweety)$$

to T, which gives

$$Newcarc(T, \neg F_2) = \{\neg ostrich(tweety),\ ab(tweety)\}.$$

The negation of the conjunction of these two clauses is

$$ostrich(tweety) \vee \neg ab(tweety)\,.$$

Adding this formula to T produces no new characteristic clauses, as the only new theorems are

$$\{ostrich(tweety) \vee \neg ab(tweety),\ flies(tweety)\},$$

and neither belongs to $\mathcal{P}$. Thus, as expected, F_2 is in the circumscribed theory.

2.2 Ginsberg's Results

Ginsberg [1989] presents an another algorithm for computing circumscription. The algorithm however works only in the case where Q, the set of fixed predicates, is empty. We will transform Ginsberg's definitions and results to ours.

Definition 2.9 [Ginsberg, 1989, Definition 3.1]
Let D and T be two sets of formulas. G is *dnf wrt* D if it is written as a disjunction of conjunctions of elements of D. And F is *confirmed by* G (wrt T and D) if the following conditions hold:

1. $T \cup \{G\}$ is satisfiable,
2. $T \cup \{G\} \models F$, and
3. G is dnf wrt D.

Comparing Definition 2.9 with Definition 2.4, we see that F is confirmed by G wrt D if G is a disjunction of explanations of F from (T, D). Now, $\neg G$ is a conjunction of clauses belonging to the production field $\neg \cdot D$ by Theorem 2.5. Or, in other words,

Definition 2.9 (new version) Let $\mathcal{P} = \neg \cdot D$. F is confirmed by G if $\neg G$ is a conjunction of clauses in $Newcarc(T, \neg F)$. Moreover, F is *unconfirmed*, if no G confirms F: $Newcarc(T, \neg F) = \phi$.

Next is the main result:

Proposition 2.10 [Ginsberg, 1989, Proposition 3.2]
Let D be P^-. $CIRC(T; P; Z) \models F$ iff there is some G confirming F so that $\neg G$ is unconfirmed.

We can rewrite it as:

Proposition 2.10 (new version) Let $\mathcal{P} = \neg \cdot D$. $CIRC(T; P; Z) \models F$ iff there is a conjunct G of clauses in $Newcarc(T, \neg F)$ such that $Newcarc(T, \neg G) = \phi$.

Ginsberg briefly mentions connections with Przymusinski's work and the possibility of relaxing the assumption of all non-minimized predicates being variable. Our above results show that:

1. This last proposition is exactly Theorem 2.8.
2. All results can thus be extended to the case $Q \neq \phi$ (that is, not varying all predicates) just by setting $D = P^- \cup Q^{\pm}$, that is, $\mathcal{P} = \neg \cdot D = P^+ \cup Q^{\pm}$.

The intuition behind Theorem 2.8 and Proposition 2.10 is the following. From the viewpoint of abductive reasoning, those theorems say that $CIRC(T, P, Z) \models F$ iff there is a disjunct G of explanations from (T, D) such that there exists no explanation of $\neg G$ from (T, D) [3], and Poole [1989] introduces the similar condition for a formula to hold in all extensions of (T, D)[4]. For answering queries in circumscription, the hypotheses D must be carefully chosen in the direction of (P, Z)-minimization: for minimized predicates P, P^- should be hypothesized, and for fixed predicates Q, $Q^{\pm}$ should be taken into account. Now the existence of an explanation of F from (T, D) guarantees that F holds in at least one extension of (T, D) by Corollary 2.6. Clearly, if some disjunct G of explanations of F holds in all extensions, then F also holds in all extensions. Since this G is constructed over D and thus does not contain literals from Z, we see that G holds in all extensions of (T, D) iff $Newcarc(T, G) = \phi$ wrt $\mathcal{P} = \neg \cdot D$ (by Theorem 2.7) iff there is no explanation of $\neg G$ from (T, D) (by Corollary 2.6).

3 Comparing the Algorithms

In the last section we showed that both Przymusinski's and Ginsberg's algorithms were based on the same theoretical results. This section concerns the computational efficiency of the algorithms.

Przymusinski [1989] defines *MILO-resolution*, a variant of ordered linear (OL) resolution [Chang and Lee, 1973]. Given a clause C, MILO-resolution is used to deduce a set of minimal clauses belonging to $Th_{\mathcal{P}}(T \cup \{C\})$, called the *derivative of* $T + C$, with top clause C and the background theory T. The algorithm needs to check the non-deducibility of each clause in the derivative from T, in order to determine the new characteristic clauses. On the other hand, Ginsberg's [1989] circumscriptive theorem prover uses a "backward-chaining ATMS" [Reiter and de Kleer, 1987] to compute minimal explanations of formulas. This backward chaining procedure also uses a classical theorem prover.

While the structure of the proofs are similar, each algorithm has a different concern and extends a resolution procedure in a different way. Remember that we should produce clauses (i) in the production field, and (ii) the "new" and "minimal" of these, that is, neither implied by the

[3] Lin & Goebel [1989] independently derive the equivalent theorem from the result by [Gelfond *et al.*, 1989] within the Theorist framework [Poole *et al.*, 1987].

[4] Etherington [1987] has shown the equivalence of membership in all extensions and circumscriptive entailment for propositional theories without fixed predicates.

original theory nor by another produced clause. MILO-resolution provides the ability to restrict the resolution to some literals by which the algorithm directly focuses on producing the clauses relevant to answer the query, that is, those in the production field $P^+ \cup Q^\pm$. Przymusinski's concern is thus efficiency regarding the first of the above two points. Ginsberg uses a classical theorem prover; this means that no information concerning the production field is used during the proof. His algorithm has however another concern, that of the minimality of the produced formulas. For this, he uses a structure called a "bilattice" based on his previous work on multivalued logic [Ginsberg, 1988]. The role of this bilattice is to record inferences, in order to avoid making them more than once. He is thus concerned with the second of the above.

The next two subsections expand on these ideas. The discussion is based on each resolution procedure to compute $Newcarc(T, C)$, given a background theory T, a clause C and a production field $\mathcal{P}$.

3.1 What Needs to be Computed

From the results presented above, it appears that to answer a query, an algorithm should first compute the minimal explanations of a formula F from $(T, P^- \cup Q^\pm)$, or equivalently their negations, $Newcarc(T, \neg F)$, with the production field set to $P^+ \cup Q^\pm$. Ginsberg's theorem prover works exactly along this line of computation [5].

However, there is a set smaller than $Newcarc(T, F)$ that can be used to answer such a query. Let us divide the produced clauses $\mathcal{S}$ by using deductions with top clause C, the background theory T and the production field $\mathcal{P}$, possibly containing subsumed clauses (note that $Newcarc(T, C) \subset \mathcal{S}$; see Theorem A.3) into two sets, say $\mathcal{S}_1$ and $\mathcal{S}_2$, such that

$$\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \text{ and } T \cup \mathcal{S}_1 \models \mathcal{S}_2 .$$

Adding $\mathcal{S}_2$ to $\mathcal{S}_1$ does not change the models of the produced clauses, so only $\mathcal{S}_1$ needs to be computed model-theoretically. We call a set $\mathcal{S}_1$ verifying this condition a *precursor of* $\mathcal{S}$. Note that a clause in a precursor may not belong to $Newcarc(T, \mathcal{P})$, that is, the clause is not always minimal in the sense of set-inclusion, but it is the weakest in the sense that for any clause $A_2 \in \mathcal{S}_2$ there exists a clause $A_1 \in \mathcal{S}_1$ such that $T \cup \{\neg A_2\} \models \neg A_1$ holds (recall that for $A \in \mathcal{S}$, $\neg A$ is an explanation of $\neg C$ from $(T, \neg \cdot \mathcal{P})$ if it is consistent with T [6]).

MILO-resolution actually computes such a precursor, as the derivative of $T + C$, because it restricts the resolution to literals belonging to $Z^\pm \cup P^-$. In other words, when the first literal of the center clause belongs to $\mathcal{P} = P^+ \cup Q^\pm$, it is *skipped*. If it were resolved upon with a clause from the theory, the resulting leave obtained by chaining the inference would be implied by the one obtained with the skipping operation (see Theorem 4.2). This is best understood with an example.

Example 3.1 The theory is

$$T = \{\, p_1 \vee \neg p_2, \quad p_2 \vee \neg p_3, \quad p_3 \vee z_1 \,\} .$$

[5] This is in essence what [Lin and Goebel, 1989] does too.

[6] An explanation E_1 is called *less-presumptive than* E_2 [Poole, 1989] if $T \cup \{E_2\} \models E_1$. Therefore, an explanation in $\neg \cdot \mathcal{S}_1$ is a least-presumptive explanation of $\neg C$ from $(T, \neg \cdot \mathcal{P})$.

The production field is $\mathcal{P} = P^+ = \{p_1, p_2, p_3\}^+$. The query is z_1.

By adding $\neg z_1$ to T, MILO-resolution generates only p_3, the only new theorem that belongs to $\mathcal{P}$. Since this literal belongs to $\mathcal{P}$, the procedure skips it and stops. It then adds $\neg p_3$ to T which generates no characteristic clause, showing that $CIRC(T; P; Z) \models z_1$.

If the procedure would examine the remaining choice, resolving p_3 with the clause $p_2 \vee \neg p_3$, it would produce p_2, and a further step would produce p_1. This is exactly what Ginsberg's prover does, as it uses no information from $\mathcal{P}$ to stop the execution when p_3 is produced. The set of assumptions is $D = \neg \cdot \mathcal{P} = P^-$. The confirmation of z_1 produced by an ATMS is dnf:

$$\neg p_3 \vee \neg p_2 \vee \neg p_1 \,.$$

The negation of the confirmation is

$$p_3 \wedge p_2 \wedge p_1 \,,$$

which is unconfirmed. Using Ginsberg's terminology, two additional contexts have been produced, $\{\neg p_1\}$ and $\{\neg p_2\}$, in which z holds (the three are produced because none of them is a subset of another). MILO-resolution did not need to generate them. As explained above, the reason is that $\{p_3\}$ is the precursor of the others, as:

$$T \cup \{p_3\} \models p_1 \wedge p_2 \,.$$

There is another big difference between Przymusinski's and Ginsberg's provers concerning checking the consistency of hypotheses. Recall that to apply Theorem 2.8 or Proposition 2.10, we need two steps; firstly computing a set of clauses belonging to $Newcarc(T, \neg F)$ for the query F, then checking whether a conjunct G of those clauses satisfies $Newcarc(T, \neg G) = \phi$. Ginsberg's prover first computes the minimal explanations $\mathcal{E}$ of F from $(T, \neg \cdot \mathcal{P})$, then computes the minimal explanations of $\neg \bigvee_{E \in \mathcal{E}} E$ from $(T, \neg \cdot \mathcal{P})$; on the contrary, Przymusinski's prover first computes the derivative $\mathcal{D}_1$ of $T + F$, without checking the non-deducibility of each clause in $\mathcal{D}_1$ from T, then computes the derivative $\mathcal{D}_2$ of $T + \neg \bigvee_{A \in \mathcal{D}_1} A$, checking whether $T \not\models B$ for each clause B in $\mathcal{D}_2$ one by one. Clearly, in the second step, we need not compute all the minimal explanations for the negation of the disjunct; if it has at least one explanation then we can stop the computation immediately. For the first step, Przymusinski's prover may include some clauses belonging to $Th_{\mathcal{P}}(T)$ in $\mathcal{D}_1$, which are excluded from the produced clauses by Ginsberg's prover [7]. However, since it takes much computation for this consistency checking (non-decidable for the first-order case), it seems rather efficient even if these extra clauses are taken into account in the second step (indeed, the efficiency may depend on the knowledge base).

3.2 How it is Computed

Now suppose both algorithms have to compute the same set, that is, MILO-resolution computes all the new characteristic clauses by avoiding the skipping of literals, or Ginsberg's theorem prover is restricted to computing a precursor. In that case, another advantage of using the information on the production field $\mathcal{P}$ during the deduction is that fewer clauses are generated.

[7] If there is a clause $A \in \mathcal{D}_1$ such that $T \models A$, then in the second step, the negation of the disjuncts contains $\neg A$. Since its valuation is false, this does not affect the result of the query answering.

For a very simple example, suppose that the center clause is $z \vee q$, where $z \notin \mathcal{P}$, while $q \in \mathcal{P}$. If z cannot be resolved upon against clauses of the theory in such a way that the result of the deduction produces a clause belonging to $\mathcal{P}$, MILO-resolution will never try to resolve on the next literal q. Conventional theorem provers will give no priority to z over q and thus will try all the resolutions on q as well, making unnecessary computation. This example, although trivial, is representative of what will happen in many realistic situations.

We said above that a central concern of [Ginsberg, 1989] was to avoid computing the same clauses more than once. The role of the bilattice is to record information and use it to avoid redundant derivations by making subsumption tests. Avoiding the exploration of unnecessary portions of the search space, and in particular the non-production of subsumed clauses has been a central concern of automated theorem proving and is one of the motivations behind all the refinements of resolution [Loveland, 1978]. Many of these use the information of literals that have been resolved upon to avoid producing many of redundant clauses. For example, OL-resolution on which MILO-resolution uses *framed literals* to record the history of the deduction. Regarding other problems related to irredundancy and control, a thorough analysis can be found in the chapter on subsumption in [Loveland, 1978]. There is not enough information in [Ginsberg, 1989; Ginsberg, 1988] to determine whether the bilattice represents a better alternative.

4 Improving Efficiency

We show here how MILO-resolution's search space can be reduced. MILO-resolution is based on Chang and Lee's [1973] version of OL-resolution; this procedure is augmented with the ability to skip literals when they belong to the production field.

Now, actually there exist superior versions of linear resolution that can be augmented with skipping operations. Most notably, Model Elimination [Loveland, 1978] and SL-resolution [Kowalski and Kuhner, 1971]. Basically, the Model Elimination procedure introduced the restriction that without loss of completeness, it can avoid resolving the center clause with clauses from the theory that have literals equal to framed literals at the right of the literal resolved upon, as this would produce only clauses subsumed by some previous center clause.

Example 4.1 Suppose a clause,

$$F = a \vee d\,,$$

resolves with a clause,

$$\neg a \vee b\,,$$

in T, giving

$$b \vee [a] \vee d\,.$$

Now suppose there is a clause,

$$\neg b \vee a\,,$$

in T. The above restriction tells us that this clause may safely be skipped in the deduction, as it contains a, a literal appearing framed in the center clause. In effect, it would give the clause,

$$a \vee [b] \vee [a] \vee d\,,$$

which is subsumed by a previous center clause, F.

Clearly, this has two advantages: it restricts the search space, and avoids many of the subsumption tests in step (iii) of MILO-resolution [Przymusinski, 1989, Definition 3.1].

Additional improvements in efficiency were introduced by [Shostak, 1976] and [Bibel, 1982]. Shostak [1976] shows that we can record still more information on center clauses. When the first literal of the center clause is framed, previous versions of linear resolution delete it from the clause. Shostak's procedure complements it and keeps it in a different position called the *C-point* in the clause, where it can still be used later in the reduction to reduce the search space and do ancestor resolution as with ordinary framed literals.

Example 4.1 (continued) If now the above clause,

$$b \vee [a] \vee d\,,$$

is resolved with,

$$\neg b\,,$$

the result will be

$$d \vee (\neg b) \vee (\neg a)\,,$$

where, the notation $(\neg l)$ is used for the truncated framed literal $[l]$ moved to the C-point; thus the information that $\neg a$ and $\neg b$ are proved is kept.

4.1 Summary

We thus propose the following procedure schema. Given a set of clauses T, a clause C, and a production field $\mathcal{P}$, a deduction of a clause K from $T + C$ (the background theory T with top clause C) and $\mathcal{P}$ consists of a sequence of structured clauses, $C_0, C_1, \ldots, C_n$, such that:

1. $C_0 = \langle\, \Box,\ C\,\rangle$,
2. $C_n = \langle\, K,\ \Box\,\rangle$, and
3. $C_{i+1} = \langle\, K_{i+1},\ R_{i+1}\,\rangle$ is obtained from $C_i = \langle\, K_i,\ R_i\,\rangle$ by applying either of the following operations. We assume that R_i is ordered and l is the first literal of R_i.
   (a) **(Skip)** If $l \in \mathcal{P}$, then $K_{i+1} = K_i \vee l$ and R_{i+1} is obtained by removing l from R_i.
   (b) Otherwise, $K_{i+1} = K_i$ and R_{i+1} is obtained by a linear resolution procedure where the center clause is R_i and the background theory is T.

By using this procedure we can find a precursor without computing all the $Newcarc(T, C)$:

Theorem 4.2 If a clause L belongs to $Newcarc(T, C)$, then there is a deduction of a clause $M \in Th_{\mathcal{P}}(T \cup \{C\})$ from $T + C$ and $\mathcal{P}$ such that $T \cup \{M\} \models L$.

The query answering procedure for circumscriptive theories [Przymusinski, 1989, Algorithm 4.1] that calls MILO-resolution remains identical; the definition of the derivative of $T + C$ is just changed to be the output of the above procedure instead of the output of MILO-resolution.

4.2 Example

We might use Shostak's GC procedure as a linear resolution procedure for step 3(b) above, and get the following:

Example 4.3 (modified version of [Przymusinski, 1989, Example 3.5]) We apply the procedure to formulas with variables in order to show that it is not limited to the ground case. Let T contain the following formulas, with $P = \{learns, senior\}$, and $Z = \phi$.

$$\forall X\ senior(X) \supset learns(X, latin) \vee learns(X, greek),$$

$$\forall X\ \ senior(X) \supset learns(X, french),$$

$$senior(ann),$$

$$\forall X\ \ learns(X, greek) \supset senior(X).$$

In clausal form and with the obvious abbreviations, the theory is

$$l(X, lt) \vee l(X, gr) \vee \neg s(X), \qquad (1)$$

$$\neg s(X) \vee l(X, fr), \qquad (2)$$

$$s(a), \qquad (3)$$

$$\neg l(X, gr) \vee s(X). \qquad (4)$$

The production field $\mathcal{P}$ is then $P^+ = \{l, s\}^+$. Consider the query $\neg l(a, gr) \vee \neg l(a, fr)$. The following is a deduction obtained by our procedure.

$C_0 = \langle\, \Box,\ \neg l(a, gr) \vee \neg l(a, fr)\, \rangle$ —Given.

$C_1 = \langle\, \Box,\ l(a, lt) \vee \neg s(a) \vee [\neg l(a, gr)] \vee \neg l(a, fr)\, \rangle$

—Resolution with 1 (*).

$C_2 = \langle\, l(a, lt),\ \neg s(a) \vee [\neg l(a, gr)] \vee \neg l(a, fr)\, \rangle$

—Skip the literal from $\mathcal{P}$ (**).

$C_3 = \langle\, l(a, lt),\ [\neg s(a)] \vee [\neg l(a, gr)] \vee \neg l(a, fr)\, \rangle$

—Resolution with 3.

$C_4 = \langle\, l(a, lt),\ \neg l(a, fr) \vee (s(a)) \vee (l(a, gr))\, \rangle$

—Recording of solved literals.

$C_5 = \langle\, l(a, lt),\ \neg s(a) \vee [\neg l(a, fr)] \vee (s(a)) \vee (l(a, gr))\, \rangle$

—Resolution with 2.

$C_6 = \langle\, l(a, lt),\ [\neg l(a, fr)] \vee (s(a)) \vee (l(a, gr))\, \rangle$

—Truncation using the solved literal (***).

$C_7 = \langle\, l(a, lt),\ \Box\, \rangle$ —Reduction.

Now according to Theorem 2.7, we can answer "no" to the above query, that is,

$$CIRC(T; P; Z) \not\models \neg l(a, gr) \vee \neg l(a, fr),$$

because $l(a, lt)$ is not implied by T.

Let us look at some advantages of this deduction over MILO-resolution and Ginsberg's theorem prover.

- At point (*), Ginsberg's prover, which lacks information on the production field, will resolve on $l(a, lt)$, instead of skipping it, thus exploring branches that are pruned by this skipping operation.
- At point (**), MILO-resolution will behave as in the above deduction, but keeps an additional choice that results from resolving $\neg s(a)$ with clause 4. Our procedure avoids this because clause 4 contains the literal $\neg l(a, gr)$ that appears framed in the center clause, thus indicating the remaining choice is unnecessary.
- At point (***), MILO-resolution would have lost information about $s(a)$, and thus makes a resolution against all clauses containing $\neg s(a)$. Reduction with solved literal avoids this exploration.

5 Concluding Remarks

We have compared two algorithms to compute circumscription, relating them to abductive reasoning, and showing that they are based on the same theoretical results. We have also analyzed their computational properties, showing their different concerns: [Przymusinski, 1989] defines the set of formulas that needs to be computed and uses skipping operations to compute them directly; [Ginsberg, 1989] concerns avoiding redundancy by recording information during a deduction.

The skipping operation can be applied to other, more efficient versions of linear resolution, and further improvements on these methods can be incorporated into the procedure. Other techniques of theorem proving can be used to improve efficiency still more. For example, we can "compile" the theory, producing either its prime implicates [Reiter and de Kleer, 1987] or the sub-clauses implied by it. In both cases, the resultant theory has the same models, and thus the same (P, Z)-minimal models as the former. Deduction from this compiled theory will give the same results as from the former.

The improvements in the present paper are based on direct refinements of linear resolution procedures, and actually applicable to efficient computation of $Newcarc(T, \neg F)$ for a query F and $Newcarc(T, \neg G)$ for some G in Theorem 2.8. However, both Przymusinski's [1989] and Ginsberg's [1989] algorithms are naive implementations of Theorem 2.8, which states the need for the existence of a certain conjunct G of $Newcarc(T, \neg F)$, ignoring that many of clauses in $Newcarc(T, \neg F)$ may exist. Therefore they turn out to suffer from the following two problems in their computations:

1. Both "algorithms" do not work for the case in which there are potentially an *infinite number of* clauses in $Newcarc(T, \neg F)$. Even if the number of $Newcarc(T, \neg F)$ is finite, not all of them are the relevant parts needed to determine that F is in the circumscribed theory [8].

2. Neither algorithm can handle the *answer extraction for open queries.* That is, when a query contains variables, the algorithms cannot return the substitution values of the variables for which the query holds. This is a much broader problem than "Yes/No" type questions.

A procedure attempting to solve the first problem is proposed by Poole [1989], which is the dialectical implementation of membership in all extensions. In [Helft *et al.*, 1989], which complements this paper, a solution for both the first and the second problems was proposed, which finds a minimal, rather than maximal conjunct G of $Newcarc(T, \neg F)$. While we will not discuss it further in this paper, we should note that a certain subset of $Newcarc(T, \neg F)$ must be computed anyway. Therefore, the improvements proposed in this paper can still be applied to any proof procedure attempting to solve these problems.

Acknowledgment

We would like to thank David Poole, Koichi Furukawa and Randy Goebel for their helpful discussions on this topic and their useful comments on an earlier version of this paper. We are also grateful to Mark Stickel for his comments about improving efficiency of proof procedures.

References

[Bibel, 1982] Wolfgang Bibel. A Comparative Study of Several Proof Procedures. *Artificial Intelligence* **18** (1982) 269–293.

[Bossu and Siegel, 1985] Genevieve Bossu and Pierre Siegel. Saturation, Nonmonotonic Reasoning, and the Closed-World Assumption. *Artificial Intelligence* **25** (1985) 23–67.

[Chang and Lee, 1973] Chin-Liang Chang and Richard Char-Tung Lee. *Symbolic Logic and Mechanical Theorem Proving.* Academic Press, New York, 1973.

[Etherington, 1987] David W. Etherington. Relating Default Logic and Circumscription. *Proc. IJCAI-87*, Milan (1987) 489–494.

[Gelfond *et al.*, 1989] Michael Gelfond, Halina Przymusinska, and Teodor Przymusinski. On the Relationship between Circumscription and Negation as Failure, *Artificial Intelligence* **38** (1989) 75–94.

[Ginsberg, 1988] Matthew L. Ginsberg. Multivalued Logics: A Uniform Approach to Reasoning in Artificial Intelligence. *Computational Intelligence* **4** (1988) 265–316.

[8] While Przymusinski's prover does not compute all of $Newcarc(T, \neg F)$ but computes a precursor of them, in some cases the precursor may have potentially infinite clauses especially for the first-order case.

[Ginsberg, 1989] Matthew L. Ginsberg. A Circumscriptive Theorem Prover. *Artificial Intelligence* **39** (1989) 209–230.

[Helft *et al.*, 1989] Nicolas Helft, Katsumi Inoue, and David Poole. Extracting Answers in Circumscription. ICOT Technical Memorandum TM-855, ICOT, Tokyo, 1989.

[Kowalski and Kuhner, 1971] Robert A. Kowalski and D. G. Kuhner. Linear Resolution with Selection Function. *Artificial Intelligence* **2** (1971) 227–260.

[Lifschitz, 1985] Vladimir Lifschitz. Computing Circumscription. *Proc. IJCAI-85*, Los Angeles (1985) 121–127.

[Lin and Goebel, 1989] Dekang Lin and Randy Goebel. Computing Circumscription of Ground Theories with Theorist. Technical Report TR 89-26, Department of Computer Science, The University of Alberta, Edmonton, 1989.

[Loveland, 1978] Donald W. Loveland. *Automated Theorem Proving: A Logical Basis.* North-Holland, Amsterdam, 1978.

[McCarthy, 1980] John McCarthy. Circumscription—A Form of Non-Monotonic Reasoning. *Artificial Intelligence* **13** (1980) 27–39.

[Minicozzi and Reiter, 1972] Eliana Minicozzi and Raymond Reiter. A Note on Linear Resolution Strategies in Consequence-Finding. *Artificial Intelligence* **3** (1972) 175–180.

[Poole, 1989] David Poole. Explanation and Prediction: An Architecture for Default and Abductive Reasoning. *Computational Intelligence* **5** (1989) 97–110.

[Poole *et al.*, 1987] David Poole, Randy Goebel and Romas Aleliunas. Theorist: A Logical Reasoning System for Defaults and Diagnosis. In: Nick Cercone and Gordon McCalla, editors, *The Knowledge Frontier: Essays in the Representation of Knowledge.* Springer-Verlag, New York (1987) 331–352.

[Przymusinski, 1989] Teodor C. Przymusinski. An Algorithm to Compute Circumscription. *Artificial Intelligence* **38** (1989) 49–73.

[Reiter and de Kleer, 1987] Raymond Reiter and Johan de Kleer. Foundations of Assumption-Based Truth Maintenance Systems: Preliminary Report. *Proc. AAAI-87*, Seattle (1987) 183–187.

[Shostak, 1976] Robert E. Shostak. Refutation Graphs. *Artificial Intelligence* **7** (1976) 51–64.

[Siegel, 1987] Pierre Siegel. Représentation et Utilisation de la Connaissance en Calcul Propositionnel. PhD thesis, University of Aix-Marseille II, Marseille, 1987.

A Appendix: Proofs of Theorems

The next lemma is used to prove Theorem 2.5.

Lemma A.1 Let T be a set of clauses, F a formula.

$$Newcarc(T, F) = \mu\,[\,Th_{\mathcal{P}}(T \cup \{F\}) - Th_{\mathcal{P}}(T)\,]\,.$$

> **Proof:** Let $A = Th_{\mathcal{P}}(T \cup \{F\})$ and $B = Th_{\mathcal{P}}(T)$. Notice that $B \subseteq A$. We will prove that $\mu[A - B] = \mu[A] - \mu[B]$.
>
> Let $c \in \mu[A - B]$. Then obviously $c \in A - B$ and thus $c \in A$. Now assume that $c \notin \mu[A]$. Then $\exists d \in A$ such that $d \subset c$. By the minimality of $c \in A - B$, $d \in B$. Since $d \subset c$, $c \in B$, contradiction. Therefore $c \in \mu[A]$. Clearly, by $c \notin B$, $c \notin \mu[B]$. Hence, $c \in \mu[A] - \mu[B]$.
>
> Conversely, assume that $c \in \mu[A] - \mu[B]$. Firstly we must prove that $c \in A - B$. Suppose to the contrary that $c \in B$. Since $c \notin \mu[B]$, $\exists d \in B$ such that $d \subset c$. However, as $B \subseteq A$, $d \in A$, contradicting the minimality of $c \in A$. Therefore, $c \in A - B$. Now assume that c is not minimal in $A - B$. Then, $\exists e \in A - B$ such that $e \subset c$, again contradicting the minimality of $c \in A$. Hence, $c \in \mu[A - B]$. □

Theorem 2.5 Let T be a set of clauses, D a set of ground literal, F a formula. The set of all minimal explanations of F from (T, D) is $\neg \cdot Newcarc(T, \neg F)$, where $\mathcal{P} = \neg \cdot D$.

> **Proof:** Now, suppose that E is an explanation of F from (T, D). By Definition 2.4, it is observed that (i) the fact that $T \cup \{E\}$ is satisfiable means $T \not\models \neg E$, (ii) $T \cup \{E\} \models F$ can be written as $T \cup \{\neg F\} \models \neg E$, and $\neg E$ is a clause all of whose literals belong to $\neg \cdot D$. Thus $\neg E \in Th_{\mathcal{P}}(T \cup \{\neg F\}) - Th_{\mathcal{P}}(T)$. By Lemma A.1, E is a minimal explanation of F from (T, D) iff $\neg E \in Newcarc(T, \neg F)$. □

We need the following preliminaries for the proof of Theorem 4.2. In the subsequent discussion, we will denote a clause as a set of literals. Firstly, a complete abductive procedure is defined by modifying the procedure described in Section 4.1 as follows:

Definition A.2 Given a set of clauses T, a clause C, and a production field $\mathcal{P}$, an *LS (Skipping Linear) deduction of a clause K from $T + C$ and $\mathcal{P}$* consists of a sequence of structured clauses, $C_0 = \langle\, \Box,\ C\,\rangle, \ldots, C_i, C_{i+1}, \ldots, C_n = \langle\, K,\ \Box\,\rangle$, such that $C_{i+1} = \langle\, K_{i+1},\ R_{i+1}\,\rangle$ is obtained from $C_i = \langle\, K_i,\ R_i\,\rangle$ by applying either of the following operations (we assume that R_i is ordered and l is the first literal of R_i):

> 3(a′) **(Skip)** If $l \in \mathcal{P}$, then $K_{i+1} = K_i \cup \{l\}$ and $R_{i+1} = R_i - \{l\}$.
>
> 3(b′) $K_{i+1} = K_i$ and R_{i+1} is obtained by a linear resolution procedure where the center clause is R_i and the background theory is T.

An example of LS resolution is proposed by Siegel [1987], which incorporates the restriction rule used in Example 4.1. The only difference between an LS deduction and one in Section 4.1 is that while the rules 3(a) and 3(b) in the latter are exclusive, in the former 3(a′) and 3(b′) are not; for $l \in \mathcal{P}$ either rule can be applied. The next theorem shows that LS resolution is complete for finding new characteristic clauses.

Theorem A.3 If a clause L belongs to $Th_{\mathcal{P}}(T \cup \{C\}) - Th_{\mathcal{P}}(T)$, then there is an LS deduction of a clause $M \in Th_{\mathcal{P}}(T \cup \{C\})$ from $T + C$ and $\mathcal{P}$ such that M subsumes L.

> **Proof:** The proof can be seen as an extension of the completeness result for consequence-finding in linear resolution by Minicozzi & Reiter [1972] augmented with the skipping operation. And the result also follows easily using the same method as in the completeness proof for the procedure described in [Siegel, 1987]. □

By using Theorem A.3, we can show that $Newcarc(T, C)$ is a subset of the set of clauses derived using LS deductions from $T + C$ and $\mathcal{P}$. Now, we will prove that, if a clause L is derived by using an LS deduction from $T + C$ and $\mathcal{P}$, then there is an LS deduction of a clause M from $T + C$ and $\mathcal{P}$ by using only the **Skip** rule for each first literal $l \in \mathcal{P}$ in every center clause, such that $T \cup \{M\} \models L$. This result completes the proof of Theorem 4.2.

Theorem 4.2 If a clause L belongs to $Newcarc(T, C)$, then there is a deduction of a clause $M \in Th_{\mathcal{P}}(T \cup \{C\})$ from $T + C$ and $\mathcal{P}$ such that $T \cup \{M\} \models L$.

> **Proof:** Let $C_0, C_1, \ldots, C_n$ be an LS deduction of L from $T + C$ and $\mathcal{P}$. Let l_i be the first literal of R_i, where $C_i = \langle K_i,\ R_i \rangle$ and $0 \leq i \leq n - 1$. Firstly, if **Skip** is applied for every l_j $(0 \leq j \leq n-1)$ such that $l_j \in \mathcal{P}$, then L is actually derived from $T + C$ and $\mathcal{P}$, and of course $T \cup \{L\} \models L$ holds.
>
> Next, suppose that $\exists C_j$ in the LS deduction such that $l_j \in \mathcal{P}$ is resolved upon with a clause $B_j \in T$. In the following proof, to simplify the discussion, we assume that there are no identical, truncated, or reduced literals in R_{y+1} and denote R_{y+1} by removing the framed literals in it; if they exist, then we can modify the proof appropriately. Let x $(1 \leq x \leq n)$ be the number of such clauses, and C_y be such a clause where y $(0 \leq y \leq n - 1)$ is the largest number. In this case, $C_{y+1} = \langle K_{y+1},\ R_{y+1} \rangle$, where $K_{y+1} = K_y$ and $R_{y+1} = (B_y - \{\neg l_y\}) \cup (R_y - \{l_y\})$. Now, let U be a clause LS derived from $T + (B_y - \{\neg l_y\})$ and $\mathcal{P}$, V a clause LS derived from $T + (R_y - \{l_y\})$ and $\mathcal{P}$. Here, we can choose such U and V to satisfy $L = K_y \cup U \cup V$, because L is LS derived from $T + (K_{y+1} \cup R_{y+1})$ and $\mathcal{P}$.
>
> Now assume that instead of resolving R_y with B_y, **Skip** is applied to K_y, deducing $K'_{y+1} = \langle K'_{y+1},\ R'_{y+1} \rangle$, where $K'_{y+1} = K_y \cup \{l_y\}$ and $R'_{y+1} = R_y - \{l_y\}$. Then, $K_y \cup \{l_y\} \cup V$ is LS derived from $T + (K'_{y+1} \cup R'_{y+1})$ and $\mathcal{P}$, and thus from $T + C$ and $\mathcal{P}$. Since $T \cup \{l_y\} \models B_y - \{\neg l_y\}$, $T \cup \{l_y\} \models U$ holds, and thus $T \cup \{(K_y \cup \{l_y\} \cup V)\} \models L$ holds.
>
> Now let $M_0 = L$ and $M_1 = (K_y \cup \{l_y\} \cup V)$. In the similar way, we can find an LS deduction of M_2 from $T + C$ and $\mathcal{P}$ such that $T \cup \{M_2\} \models M_1$, by resetting y to the

second largest number. By using the bottom-up manner, we can successively find clauses M_j $(1 \leq j \leq x)$ LS derived from $T + C$ and $\mathcal{P}$ such that $T \cup \{M_j\} \models M_{j-1}$. Therefore, $T \cup \{M_x\} \models M_{x-1}$, $T \cup \{M_{x-1}\} \models M_{x-2}$, ..., $T \cup \{M_1\} \models M_0$. Hence, $T \cup \{M_x\} \models M_0$, and we get the theorem. □