九州大学学術情報リポジトリ
Kyushu University Institutional Repository

Finding Minimal Models by the Semantic Tableaux System

Hirata, Kouichi
Research Institute of Fundamental Information Science, Kyushu University

http://hdl.handle.net/2324/3068

出版情報：RIFIS Technical Report. 72, 1993-05. 九州大学理学部附属基礎情報学研究施設
バージョン：
権利関係：

KYUSHU UNIVERSITY

Finding Minimal Models by the Semantic Tableaux System

Kouichi Hirata
Research Institute of Fundamental Information Science,
Kyushu University 33, Fukuoka 812, Japan
e-mail: hirata@rifis.sci.kyushu-u.ac.jp

Abstract

The semantic tableaux system is a procedure to decide whether or not a given formula is valid. The system also works for finding minimal models of formulas. In this paper, we apply this function to circumscription. We show that we can obtain the minimal models of circumscription just by checking the models produced by the system, and hence we do not need to check all possible models of it. We also present an algorithm to produce the models of circumscription based on the system.

1 Introduction

Various methods of computing circumscription have been reported by many researchers; the reduction to first-order logic by Lifschitz [7], the compilation to logic programs by Gelfond and Lifschitz [4], a special type of resolution, called *MILO*-resolution by Przymusinski [11], and the method based on de Kleer's ATMS by Ginsberg [5]. All of these methods are of proving the circumscriptive theorem. In this paper, we introduce another method of finding minimal models of circumscription, which is based on the semantic tableaux system.

The *semantic tableaux* (*tableaux*, for short) system, introduced by Beth [1] and developed by Smullyan [13], Hintikka [6], Fitting [3], and Olivetti [10], is a theorem prover of first-order logic, that is, a refutation system like the resolution by Robinson [12]. The idea behind the present work is motivated by Hintikka [6]. He introduced a circumscriptive theorem prover based on the semantic tableaux system that works in the following way: Let us consider the McCarthy's sample theory [8, 6] consisting of the conjunction

$$Isblock(a) \wedge Isblock(b) \wedge Isblock(c).$$

Then by using the semantic tableaux system, we can produce a model

$$\{Isblock(a), Isblock(b), Isblock(c)\},$$

and obtain the following formula

$$\forall X(X = a \vee X = b \vee X = c).$$

The semantic tableaux system is based on disjunctive normal forms or dual clause forms, while the resolution is based on conjunctive normal forms or clause forms. In Section 2, we focus our attention on the models produced by the semantic tableaux system, which we call *tableau models*, and present an algorithm to produce such tableau models.

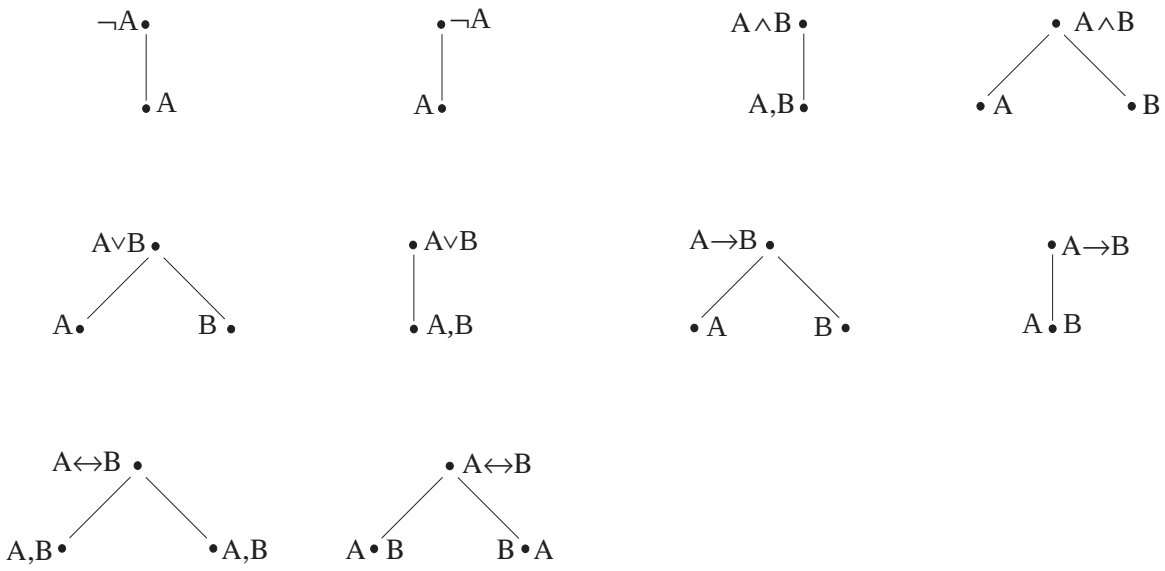

Figure 1: Propositional tableaux rules

In Section 4, we study tableau models of circumscription. In general, the model of circumscription of a formula is a minimal model of it, where a model is to be *minimal* if it is minimal for *all* the models. Hence we need to check *all* the models of a formula to obtain the minimal models, i.e., the models of circumscription. It is difficult to find all the models. We can, however, obtain the minimal models of a formula just by checking the *tableau* models, and hence we do not need to check all models of it. We also present an algorithm to produce the tableau models of circumscription. Furthermore we deal with predicate circumscription to minimize any set of predicate symbols, and deal with formula circumscription [9, 7], while Hintikka [6] only dealt with predicate circumscription to minimize all predicate symbols [8]. Thus we extend the tableau models so that we can apply our discussions to formula circumscription.

2 Semantic Tableaux System

The semantic tableaux system is a procedure to decide whether or not a given formula is valid. We use the notions on the semantic tableaux system by Bowen [2]. The *tableaux rules* for propositional logic are given in Figure 1. Note that, in the tableaux rules, we regard the lefthand side of $\bullet$ as components of the formula which are true, and the righthand side as those which are false. The rules have the following meanings: The first rule (we call $\neg\bullet$ rule) means that if $\neg A$ is true then A is false. The second rule ($\bullet\neg$ rule) means that if $\neg A$ is false then A is true. The third rule ($\wedge\bullet$ rule) means that if $A \wedge B$ is true then both A and B are true. The fourth rule ($\bullet\wedge$ rule) means that if $A \wedge B$ is false then either A or B is false, and so on.

By using tableaux rules, we can decide whether or not a given formula T is valid in the following way. At first, we put T on the righthand side of $\bullet$. For $\bullet T$, we produce a tree by applying the tableaux rules. By checking the leaves of the tree, we can decide whether or not T is valid by Theorem 1 bellow. We call this tree a *tableau proof* of T.

We denote a branch in a tree by its leaf $(L \bullet R)$, and we call L a *left column* and R a *right column*. If the same propositional variables occur on both left and right columns at the leaf, then we call a branch including its leaf a *closed branch*. Otherwise we call it an *open branch*.

Whether or not a given formula is valid is decidable by the following theorem.

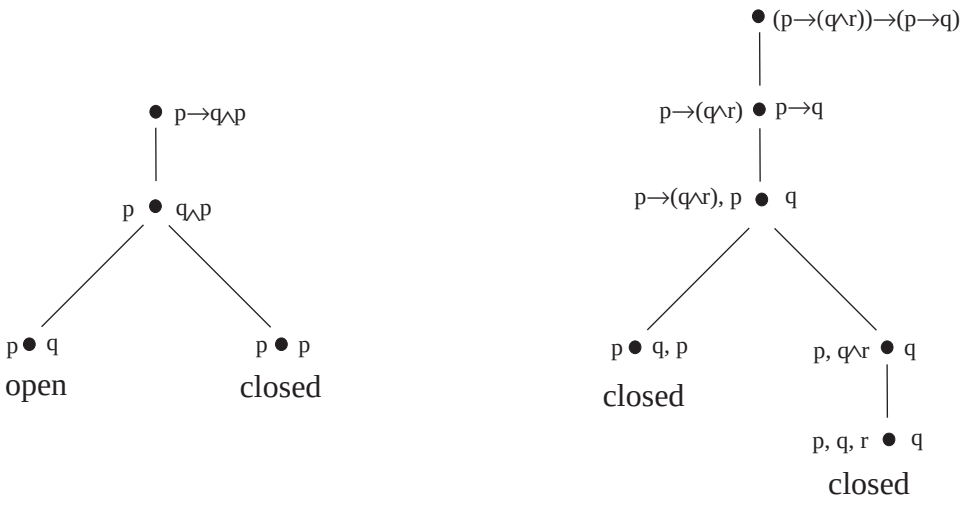

Figure 2: Tableau proofs of $A = p \rightarrow (q \wedge p)$ and $B = (p \rightarrow (q \wedge r)) \rightarrow (p \wedge q)$

Theorem 1 (Smullyan [13], Fitting [3]) All branches of the tableau proof of a given formula are closed if and only if the formula is valid.

Example 1 Let A be a formula $p \rightarrow (q \wedge p)$. Then the tableau proof of A is given as a tree in the lefthand side of Figure 2. Since the left branch in the tableau proof of A is open, A is not valid. In fact, if p is true and q is false then A is false. On the other hand, let B be a formula $(p \rightarrow (q \wedge r)) \rightarrow (p \wedge q)$. Then the tableau proof of B is given as a tree in the righthand side of Figure 2. Since all branches in the tableau proof of B are closed, B is valid.

The semantic tableaux system is based on disjunctive normal forms, but not on conjunctive normal forms. Hence we can use the semantic tableaux system not only for the refutation system, but also for finding models. Let T be a propositional formula. By putting T on the right column of $\bullet$, we can produce a tree for deciding whether or not T is valid, while by putting it on the right column of $\bullet$, we produce a tree for finding the models of T. Thus we can decide whether or not a leaf is a model. By $ltab(T)$ we denote such a tree.

Note that our semantics is an Herbrand model semantics. In this paper, we do not deal with other models than Herbrand models. Under this assumption, the following theorem holds.

Theorem 2 (Hintikka [6]) Let T be a formula. If a branch $B = (L \bullet R)$ in $ltab(T)$ is open, then $L \models T$.

Note also that we can regard a left column L as a set of propositional variables which are true in an Herbrand interpretation. Hence L in Theorem 2 is an Herbrand model.

Example 2 Figure 3 illustrates a tree $ltab((p \rightarrow (q \wedge r)) \rightarrow (p \wedge q))$. All branches in the tree are open, and $\{p\}$, ϕ, and $\{q\}$ are models of the formula by Theorem 2.

Although the semantic tableaux system does not find all Herbrand models of a given formula, the following lemma clearly holds.

Lemma 1 Let T be a formula. If M is a model of T, then there exists an open branch $B = (L \bullet R)$ in $ltab(T)$ such that $L \models T$ and $L \subseteq M$.

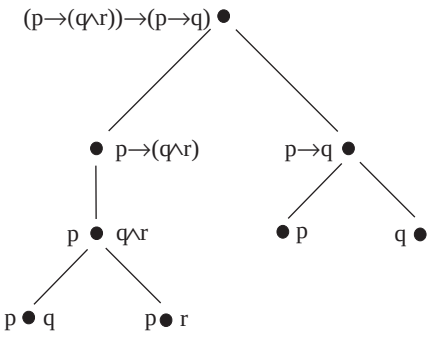

Figure 3: $ltab((p \to (q \wedge r)) \to (p \wedge q))$

Figure 4: First-order tableaux rules

By Lemma 1, any model of T includes some left column of open branch in $ltab(T)$. We call such a left column a *tableau model*.

In first-order logic, there have been reported many kinds of semantic tableaux systems that deal with quantifiers; the semantic tableaux system (Smullyan [13] and Fitting [3]), the free-variable tableaux system (Fitting [3]), and the systematic tableaux system (Smullyan [13]). We adopt here the systematic tableaux system. The systematic tableaux rules are divided into propositional tableaux rules in Figure 1 and first-order tableaux rules, $\forall\bullet$ rule, $\bullet\forall$ rule, $\exists\bullet$ rule, and $\bullet\exists$ rule in Figure 4, where a is any element of Herbrand universe.

A *substitution* is a finite set of the form $\{v_1 := t_1, \cdots, v_n := t_n\}$, where each v_i is a variable, each t_i is a term distinct from v_i and the variables $v_1, \cdots, v_n$ are mutually distinct. In Figure 4, although a is any element of Herbrand universe, we can regard $A(a)$ as the formula given by substituting a for x in $\forall x A(x)$. To apply substitutions to leaves, we need to introduce the *substitution rule* for first-order tableaux system:

Substitution rule: We apply a substitution to all leaves of the tableau proof.

Furthermore we need to introduce the *systematic tableaux rule*:

Systematic tableaux rule: For a given leaf of a tableau proof, first we apply the propositional tableaux rules, $\bullet\forall$ rule, and $\exists\bullet$ rule, and then apply $\forall\bullet$ rule and $\bullet\exists$ rule.

In this paper, we only deal with the tableau models of *function-free* formulas. Then we can show that Theorem 1,2 and Lemma 1 also hold for first-order logic. Let T be a formula and $U(T)$ be an Herbrand universe of T. For the open branch $B = (L \bullet R)$ in $ltab(T)$, we can regard the left column L as

$$\{p(\overline{t}) \mid p(\overline{x}) \in L,\ p \text{ is an } n\text{-ary predicate symbol, and } \overline{t} \in U^n(T)\},$$

which is the set of all ground atoms that are true in an Herbrand interpretation. We also call such a left column a *tableau model*.

Example 3 For a formula $p(a) \wedge \forall X(p(X) \rightarrow q(X)))$, $\{a\}$ is an Herbrand universe of the formula. Then the tableau model of the formula is $\{p(a), q(a)\}$.

Tableau models are found by the following algorithm.

Algorithm to find tableau models
input T : *formula*
produce $ltab(T)$
$B_i = (L_i \bullet R_i)$ $(1 \leq i \leq n)$: *branches in* $ltab(T)$
for $i = 1$ **to** n
if B : open **then**
output "L_i : tableau model of T"
else
output "L_i : not model of T"

3 Circumscription

By $p, q, \cdots$, we denote predicate constants and by $P, Q, \cdots$, predicate variables. We use $p(\overline{x})$ instead of $p(x_1, \cdots, x_n)$, and $\forall \overline{x}$ instead of $\forall x_1 \cdots \forall x_n$.

We introduce an order $<$ on predicate symbols as follows. Let p, q be predicate symbols with the same arity. Then $p \leq q$ stands for the formula $\forall \overline{x}(p(\overline{x}) \rightarrow q(\overline{x}))$. Let $\overline{p} = p_1, \cdots, p_n$ and $\overline{q} = q_1, \cdots, q_n$ be tuples of predicate symbols, where p_i and q_i have the same arity for any i. Then $\overline{p} \leq \overline{q}$ stands for the formula

$$(p_1 \leq q_1) \wedge \cdots \wedge (p_n \leq q_n).$$

Furthermore $\overline{p} < \overline{q}$ stands for the formula $(\overline{p} \leq \overline{q}) \wedge \neg(\overline{q} \leq \overline{p})$, and $\overline{p} = \overline{q}$ for the formula $(\overline{p} \leq \overline{q}) \wedge (\overline{q} \leq \overline{p})$.

By $T(\overline{p})$ (*resp.* $T(\overline{p}; \overline{z})$), we mean the formula T with the tuple $\overline{p}$ (*resp.* $\overline{p}$ and $\overline{z}$) of predicate symbols. Then circumscription is defined as follows.

Definition 1 Let $T(\overline{p}; \overline{z})$ be a formula, and $\overline{p}$ and $\overline{z}$ be disjoint tuples of predicate symbols. Then *formula circumscription* $CIRC(T(\overline{p}; \overline{z}); \overline{p}; \overline{z})$ *of* $\overline{p}$ *in* $T(\overline{p}; \overline{z})$ *with variables* $\overline{z}$ is defined by a simpler second-order formula:

$$CIRC(T(\overline{p}; \overline{z}); \overline{p}; \overline{z}) = T(\overline{p}; \overline{z}) \wedge \forall \overline{P}\overline{Z}(\neg T(\overline{P}; \overline{Z}) \vee \overline{P} \not< \overline{p}).$$

If $\overline{z}$ is empty, then *predicate circumscription* $CIRC(T(\overline{p}); \overline{p})$ *of* $\overline{p}$ *in* $T(\overline{p})$ is defined by the following second-order formula:

$$CIRC(T(\overline{p}); \overline{p}) = T(\overline{p}) \wedge \forall \overline{P}(\neg T(\overline{P}) \vee \overline{P} \not< \overline{p}).$$

The model-theoretic meaning of circumscription can be expressed by the following notion.

Definition 2 Let $M_1 = (D, I_1)$ and $M_2 = (D, I_2)$ be the structures of first-order logic. Let $\overline{p} = p_1, \cdots, p_n$ and $\overline{q} = q_1, \cdots, q_n$ be tuples of predicate symbols.

1. $M_1 \leq^{\overline{p}} M_2$ if
   (a) for any $c \notin \overline{p}$, $I_1(c) = I_2(c)$, and

(b) for any $c \in \overline{p}$, $I_1(c) \subseteq I_2(c)$.

2. $M_1 \leq^{\overline{p};\overline{z}} M_2$ if

   (a) for any $c \notin \overline{p} \cup \overline{z}$, $I_1(c) = I_2(c)$, and

   (b) for any $c \in \overline{p}$, $I_1(c) \subseteq I_2(c)$.

3. $M_1 <^{\overline{p}} M_2$ if $M_1 \leq^{\overline{p}} M_2$ and $\neg(M_2 \leq^{\overline{p}} M_1)$.

4. $M_1 <^{\overline{p};\overline{z}} M_2$ if $M_1 \leq^{\overline{p};\overline{z}} M_2$ and $\neg(M_2 \leq^{\overline{p};\overline{z}} M_1)$.

Let M be the structure of T. Then M is *$\overline{p}$-minimal model* (*resp.* *$(\overline{p};\overline{z})$-minimal model*) *of* T if there exists no model N of T such that $N <^{\overline{p}} M$ (*resp.* $N <^{\overline{p};\overline{z}} M$).

The models of circumscription can be characterized as follows.

Theorem 3 (McCarthy [8], Lifschitz [7]) Let T be a formula, and $\overline{p}$ and $\overline{z}$ be tuples of predicate symbols.

1. M is a model of $CIRC(T(\overline{p});\overline{p})$ if and only if M is a $\overline{p}$-minimal model of T.

2. M is a model of $CIRC(T(\overline{p};\overline{z});\overline{p};\overline{z})$ if and only if M is a $(\overline{p};\overline{z})$-minimal model of T.

Example 4 Let P be a formula $bird \wedge ((bird \wedge \neg ab) \rightarrow fly)$. Then P has three models:

$$M_1 = \{bird, fly\},\ M_2 = \{bird, ab\},\ M_3 = \{bird, fly, ab\}.$$

Hence M_1 and M_2 are ab-minimal models, i.e., models of $CIRC(P;ab)$. Furthermore M_1 is an $(ab;fly)$-minimal model, i.e., a model of $CIRC(P;ab;fly)$.

4 Tableau Models of Circumscription

In general, circumscription is said to be *true* if it is true in *all* minimal models of a given formula. In other words, $CIRC(T(\overline{p};\overline{z});\overline{p};\overline{z}) \models T$ if and only if $M \models T$ for *any* $(\overline{p};\overline{z})$-minimal model M of T. However, to avoid the difficulty of computing *all* the minimal models, we present algorithms to find *some* minimal tableau models in this section. Then we show that we can obtain the minimal models just by checking the *tableau* models found by the algorithm. Hence we do not need to check *all* the models.

First we present an algorithm to find $\overline{p}$-minimal tableau models. We prove the following two lemmas.

Lemma 2 Let T be a closed formula and $(L_i \bullet R_i)$ be the open branch in $ltab(T)$. Suppose that $L_j \not\leq^{\overline{p}} L_i$ for any open branch $(L_j \bullet R_j)$. If there exists a model M of T such that $M <^{\overline{p}} L_i$, then there exists a k such that

1. $(L_k \bullet R_k)$ is an open branch,

2. $L_k \subset M \subset L_i$,

3. for any $\alpha \in L_i - M$, $pred(\alpha) \in \overline{p}$, and

4. there exists a $\beta \in M - L_k$ such that $pred(\beta) \notin \overline{p}$.

Proof. By the assumption, $L_j \not\leq^{\overline{p}} L_i$ for any open branch $(L_j \bullet R_j)$, and $M <^{\overline{p}} L_i$. Since M is a model, by Lemma 1, there exists an open branch $(L_k \bullet R_k)$ such that $L_k \subseteq M$ and $L_k \models T$. If $L_k = M$, then $L_k \not\leq^{\overline{p}} L_i$ and $L_k <^{\overline{p}} L_i$. Hence $L_k \subset M$. Furthermore $M \subset L_i$ by $M <^{\overline{p}} L_i$. Therefore there exists a k such that $(L_k \bullet R_k)$ is open and $L_k \subset M \subset L_i$.

Let $M = L_k \cup L_1$ and $L_i = L_k \cup L_1 \cup L_2$. Then the following two facts hold.

1. $L_k \cup L_1 <^{\overline{p}} L_k \cup L_1 \cup L_2$,
2. $L_k \not\leq^{\overline{p}} L_k \cup L_1 \cup L_2$.

Suppose that there exists $\gamma \in L_2$ such that $pred(\gamma) \notin \overline{p}$. Then the interpretation of $pred(\gamma)$ in $L_k \cup L_1$ is different from the interpretation of $pred(\gamma)$ in $L_k \cup L_1 \cup L_2$. This contradicts the condition 1. Therefore $pred(\alpha) \in \overline{p}$ for any $\alpha \in L_2$.

Suppose that $pred(\delta) \in \overline{p}$ for any $\delta \in L_1$. Then $pred(\alpha) \in \overline{p}$ for any $\alpha \in L_2$. L_k and $L_k \cup L_1 \cup L_2$ are equal to the interpretation without $\overline{p}$, and the interpretation of $\overline{p}$ in L_k is included in the interpretation of $\overline{p}$ in $L_k \cup L_1 \cup L_2$. Hence $L_k <^{\overline{p}} L_k \cup L_1 \cup L_2$. This contradicts the condition 2. Therefore there exists a $\beta \in L_1$ such that $pred(\beta) \notin \overline{p}$. □

We can easily prove the following lemma.

Lemma 3 Let T be a closed formula and $(L_i \bullet R_i)$ be the open branch in $ltab(T)$. If there exists an i such that $L_j \not\leq^{\overline{p}} L_i$ for any open branch $(L_j \bullet R_j)$, then either of the following conditions holds.

1. L_i is a $\overline{p}$-minimal model.
2. There exist a k and a model M of T such that
   (a) $M \not\leq^{\overline{p}} L_i$,
   (b) $(L_k \bullet R_k)$ is an open branch,
   (c) $L_k \subset M \subset L_i$,
   (d) for any $\alpha \in L_i - M$, $pred(\alpha) \in \overline{p}$, and
   (e) there exists a $\beta \in M - L_k$ such that $pred(\beta) \notin \overline{p}$.

By the above lemmas, the following theorem holds.

Theorem 4 Let T be a closed formula and $(L_i \bullet R_i)$ be an open branch in $ltab(T)$. L_i is not a $\overline{p}$-minimal model if and only if either of the following conditions holds.

1. There exists a j such that $(L_j \bullet R_j)$ is an open branch and $L_j \leq^{\overline{p}} L_i$.
2. $L_j \not\leq^{\overline{p}} L_i$ for any open branch $(L_j \bullet R_j)$, and there exist a k and a model M of T such that
   (a) $(L_k \bullet R_k)$ is an open branch,
   (b) $L_k \subset M \subset L_i$,
   (c) $pred(\alpha) \in \overline{p}$ for any $\alpha \in L_i - M$, and
   (d) there exists a $\beta \in M - L_k$ such that $pred(\beta) \notin \overline{p}$.

Based on Theorem 4, we can construct an algorithm to find the $\overline{p}$-minimal models just by checking the tableau models. By UL_i, we denote atoms (predicates) to be minimized, i.e., $UL_i = \{q(\overline{t}) \mid q \in \overline{p}, q(\overline{t}) \in L_i\}$. We can replace the first condition of Theorem 4 by the following three conditions: B_i is an open branch, $L - UL = L_i - UL_i$, and $UL_i \subset UL$. The second condition of Theorem 4 means that $L_i \subset L$, and $L - L_i$ includes α and β such that $pred(\alpha) \in \overline{p}$ and $pred(\beta) \notin \overline{p}$. In other words, this means that $L_i \subset L$, and neither $pred(\alpha) \in \overline{p}$ nor $pred(\alpha) \notin \overline{p}$ for every $\alpha \in L - L_i$. Hence we can replace the second condition by the following four conditions: B_i is an open branch, $L_i \subset L$, $UL - UL_i \neq \phi$, and $(L - UL) - (L_i - UL_i) \neq \phi$.

By the above notions, $\overline{p}$-minimal tableau models are found by the following algorithm.

Algorithm to find $\overline{p}$-minimal tableau models

input $B, B_1, B_2, \cdots, B_n$: *branches in* $ltab(T)$
/* $(L \bullet R) = B, (L_i \bullet R_i) = B_i$ */
if B : open **then**
  for $i = 1$ **to** n
    if $\{(B_i$: open $) \wedge (L \neq L_i) \wedge (L - UL = L_i - UL_i) \wedge (UL_i \subset UL)\}$ **then**
    /* L satisfies the first condition of Theorem 4 */
      output "L : not $\overline{p}$-minimal model of T"
      halt
  for $i = 1$ **to** n
    if $\{(B_i$: open $) \wedge (L \neq L_i) \wedge (L_i \subset L) \wedge (UL - UL_i \neq \phi)$
      $\wedge((L - UL) - (L_i - UL_i) \neq \phi)\}$ **then**
    /* L satisfies the second condition of Theorem 4 */
      output "L : not $\overline{p}$-minimal model of T"
      halt
  output "L : $\overline{p}$-minimal model of T"
else
  output "L : not model of T"

Example 5 Let P be a formula $bird \wedge ((bird \wedge \neg ab) \rightarrow fly)$. Suppose that ab is a minimized propositional variable. Then a tree of $ltab(P)$ is illustrated by Figure 4. Let B_1, B_2, and B_3 be $(\{bird\} \bullet \{bird\})$, $(\{bird, ab\} \bullet \phi)$, and $(\{bird, fly\} \bullet \phi)$, respectively. Since B_1 is closed, L_1 is not a model of P. Furthermore

$$L_2 = \{bird, ab\}, \quad UL_2 = \{ab\}, \qquad L_2 - UL_2 = \{bird\},$$
$$L_3 = \{bird, fly\}, \quad UL_3 = \phi, \qquad L_3 - UL_3 = \{bird, fly\}.$$

Since $L_2 - UL_2 \neq L_3 - UL_3$ and $L_3 \not\subset L_2$, L_2 does not satisfy the conditions in Theorem 4. Hence L_2 is an ab-minimal model of P. Since $L_3 - UL_3 \neq L_2 - UL_2$ and $L_2 \not\subset L_3$, L_3 does not satisfy the conditions in Theorem 4. Hence L_3 is an ab-minimal model of P. By Theorem 4, L_2 and L_3 are proved to be ab-minimal models just by checking L_3 and L_2 respectively.

Now, we present an algorithm to find $(\overline{p}; \overline{z})$-minimal tableau models. By $M|_{\overline{z}}$, we denote $\{\alpha \in M \mid pred(\alpha) \in \overline{z}\}$. Then the following lemmas also hold in the case of formula circumscription.

Lemma 4 Let T be a closed formula and $(L_i \bullet R_i)$ be the open branch in $ltab(T)$. Suppose that $L_j \not<^{\overline{p};\overline{z}} L_i$ for any open branch $(L_j \bullet R_j)$. If there exists a model M of T such that $M <^{\overline{p};\overline{z}} L_i$, then there exists a k such that

1. $(L_k \bullet R_k)$ is an open branch,

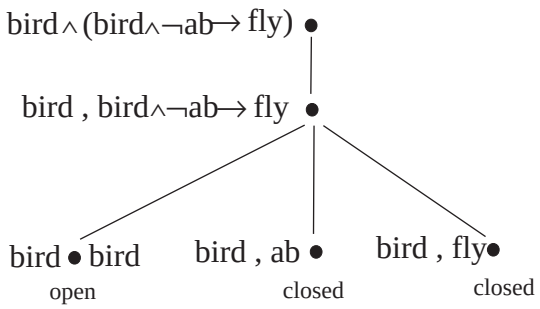

Figure 5: $ltab(bird \wedge ((bird \wedge \neg ab) \rightarrow fly))$

2. $(L_k - L_k|_{\overline{z}}) \subset (M - M|_{\overline{z}}) \subset (L_i - L_i|_{\overline{z}})$,
3. for any $\alpha \in (L_i - L_i|_{\overline{z}}) - (M - M|_{\overline{z}})$, $pred(\alpha) \in \overline{p}$, and
4. there exists a $\beta \in (M - M|_{\overline{z}}) - (L_k - L_k|_{\overline{z}})$ such that $pred(\beta) \notin \overline{p}$.

Proof. By the assumption, $L_j \not\leq^{\overline{p};\overline{z}} L_i$ for any j, and $M <^{\overline{p};\overline{z}} L_i$. Since M is a model, by lemma 1, there exists a k such that $L_k \subseteq M$. If $L_k = M$, then $L_k \not\leq^{\overline{p};\overline{z}} L_i$ and $L_k <^{\overline{p};\overline{z}} L_i$. Hence $L_k \subset M$. Furthermore $(M - M|_{\overline{z}}) \subset (L_i - L_i|_{\overline{z}})$ by $M <^{\overline{p};\overline{z}} L_i$. Therefore there exists a k such that $(L_k - L_k|_{\overline{z}}) \subset (M - M|_{\overline{z}}) \subset (L_i - L_i|_{\overline{z}})$.

Let $L'_k = L_k - L_k|_{\overline{z}}$, $M' = M - M|_{\overline{z}}$, and $L'_i = L_i - L_i|_{\overline{z}}$. Then the following two facts hold.

1. $L'_k <^{\overline{p};\overline{z}} L'_i$, and
2. $M' \not\leq^{\overline{p};\overline{z}} L'_i$.

Hence

1. $(L_k - L_k|_{\overline{z}}) <^{\overline{p}} (L_i - L_i|_{\overline{z}})$, and
2. $(M - M|_{\overline{z}}) \not\leq^{\overline{p}} (L_i - L_i|_{\overline{z}})$.

By Lemma 2, this completes the proof. □

Lemma 5 Let T be a closed formula and $(L_i \bullet R_i)$ be the open branch in $ltab(T)$. If there exists an i such that $L_j \not\leq^{\overline{p};\overline{z}} L_i$ for any open branch $(L_j \bullet R_j)$, Then either of the following facts holds.

1. L_i is a $(\overline{p};\overline{z})$-minimal model.
2. There exist a k and a model M of T such that
   (a) $M \not\leq^{\overline{p};\overline{z}} L_i$,
   (b) $(L_k \bullet R_k)$ is an open branch,
   (c) $(L_k - L_k|_{\overline{z}}) \subset (M - M|_{\overline{z}}) \subset (L_i - L_i|_{\overline{z}})$,
   (d) for any $\alpha \in (L_i - L_i|_{\overline{z}}) - (M - M|_{\overline{z}})$, $pred(\alpha) \in \overline{p}$, and
   (e) there exists a $\beta \in (M - M|_{\overline{z}}) - (L_k - L_k|_{\overline{z}})$ such that $pred(\beta) \notin \overline{p}$.

By the above lemmas, the following corollary of Theorem 4 holds for formula circumscription.

Corollary 1 Let T be a closed formula and $(L_i \bullet R_i)$ be the open branch in $ltab(T)$. L_i is not a $(\overline{p};\overline{z})$-minimal model if and only if either of the following conditions holds.

1. There exists a j such that $(L_j \bullet R_j)$ is an open branch and $L_j \leq^{\overline{p};\overline{z}} L_i$.
2. $L_j \not\leq^{\overline{p};\overline{z}} L_i$ for any open branch $(L_j \bullet R_j)$, and there exist a k and a model M of T such that
   (a) $(L_k \bullet R_k)$ is an open branch,
   (b) $(L_k - L_k|_{\overline{z}}) \subset (M - M|_{\overline{z}}) \subset (L_i - L_i|_{\overline{z}})$,
   (c) for any $\alpha \in (L_i - L_i|_{\overline{z}}) - (M - M|_{\overline{z}})$, $pred(\alpha) \in \overline{p}$, and
   (d) there exists a $\beta \in (M - M|_{\overline{z}}) - (L_k - L_k|_{\overline{z}})$ such that $pred(\beta) \notin \overline{p}$.

Based on Corollary 1, we present an algorithm to find the $(\overline{p};\overline{z})$-minimal models. By OL_i, we denote atoms (predicates) to be allowed to vary, i.e., $OL_i = \{q(\overline{t}) \mid q \in \overline{z}, q(\overline{t}) \in L_i\}$. We can replace the first condition of Corollary 1 by the following three conditions: B_i is an open branch, $L-(UL \cup OL) = L_i-(UL_i \cup OL_i)$, and $UL_i \subset UL$. The second condition of Corollary 1 means that $L_i - OL_i \subset L - OL$, and $(L - OL) - (L_i - OL_i)$ includes α and β such that $pred(\alpha) \in \overline{p}$ and $pred(\beta) \notin \overline{p}$. In other words, this means that $L_i - OL_i \subset L - OL$, and neither $pred(\alpha) \in \overline{p}$ nor $pred(\alpha) \notin \overline{p}$ for every $\alpha \in (L - OL) - (L_i - OL_i)$. Hence we can replace the second condition by following four conditions: B_i is an open branch, $L_i - OL_i \subset L - OL$, $UL - UL_i \neq \phi$, and $(L - (UL \cup OL)) - (L_i - (UL_i \cup OL_i)) \neq \phi$.

By the above notions, $(\overline{p};\overline{z})$-minimal tableau models are found by the following algorithm.

Algorithm to find $(\overline{p};\overline{z})$-minimal tableau models
input $B, B_1, B_2, \cdots, B_n$: *branches in* $ltab(T)$
/* $(L \bullet R) = B, (L_i \bullet R_i) = B_i$ */
if B : open **then**
  for $i = 1$ **to** n
    if $\{(B_i$: open $) \wedge (L \neq L_i) \wedge (L - (UL \cup OL) = L_i - (UL_i \cup OL_i))$
      $\wedge (UL_i \subset UL)\}$ **then**
    /* L satisfies the first condition of Corollary 1 */
      output "L : not $(\overline{p};\overline{z})$-minimal model of T"
      halt
  for $i = 1$ **to** n
    if $\{(B_i$: open $) \wedge (L \neq L_i) \wedge (L_i - OL_i \subset L - OL) \wedge (UL - UL_i \neq \phi)$
      $\wedge \{(L - (UL \cup OL)) - (L_i - (UL_i \cup OL_i)) \neq \phi)\}$ **then**
    /* L satisfies the second condition of Corollary 1 */
      output "L : not $(\overline{p};\overline{z})$-minimal model of T"
      halt
  output "L : $(\overline{p};\overline{z})$-minimal model of T"
else
  output "L : not model of T"

Example 6 Let P be a formula in Example 5. Suppose that ab is a minimized propositional variable and fly is a propositional variable allowed to vary. Then a tree of $ltab(P)$ is illustrated by Figure 4 in Example 5. Let B_1, B_2, and B_3 be $(\{bird\} \bullet \{bird\})$, $(\{bird, ab\} \bullet \phi)$, and $(\{bird, fly\} \bullet \phi)$, respectively. Since B_1 is closed, L_1 is not a model of P. Furthermore

$$L_2 = \{bird, ab\}, \quad UL_2 = \{ab\}, \quad OL_2 = \phi, \quad UL_2 \cup OL_2 = \{ab\},$$
$$L_3 = \{bird, fly\}, \quad UL_3 = \phi, \quad OL_3 = \{fly\}, \quad UL_3 \cup OL_3 = \{fly\}.$$

Since $L_2 - (UL_2 \cup OL_2) = L_3 - (UL_3 \cup OL_3)$ and $L_3 \subset L_2$, L_2 satisfies the first condition of Corollary 1. Hence L_2 is not an $(ab; fly)$-minimal model of P. Since $L_2 \not\subset L_3$ and $L_2 - OL_2 \not\subset L_3 - OL_2$, L_3 does not satisfy the conditions of Corollary 1. Hence L_3 is an $(ab; fly)$-minimal model of P. By Corollary 1, L_3 is proved to be a $(ab; fly)$-minimal model just by checking L_2.

5 Conclusion

In this paper, we have defined tableau models and presented algorithms to find the models of circumscription just by checking the tableau models.

We have implemented the algorithms in Prolog by applying Fitting's Dual Clause Form Program [3]. The implemented programs are to find propositional tableau models and $\overline{p}$- and $(\overline{p}; \overline{z})$-minimal tableau models for given formula $\overline{p}$ and $\overline{z}$.

Our algorithms find some minimal models, but not all the minimal models. For example, for the formula

$$T = bird \wedge ((bird \wedge \neg ab) \rightarrow fly) \wedge (ostrich \rightarrow \neg fly)$$

due to Przymusinski [11], $M_1 = \{bird, fly\}$ and $M_2 = \{ostrich, bird, ab\}$ are $(ab; fly)$-minimal models of T. Among them our algorithms only find M_1. Our algorithms cannot find all the models. However we can improve them by using negative inmformation.

References

[1] Beth, E.W., *The Foundation of Mathematics*, North-Holland Publishing, 1959.

[2] Bowen, K.A., *Modal Theory for Modal Logic*, D. Reidel Publishing Company, 1979.

[3] Fitting, M., *First-Order Logic and Automated Theorem Proving*, Springer-Verlag, 1990.

[4] Gelfond, M., Lifschitz, V., Compiling Circumscriptive Theories into Logic Programs : Preliminary Report, *Proceedings of 7th National Conference on Artificial Intelligence*, 455–459, 1988.

[5] Ginsberg, M.L., A Circumscriptive Theorem Prover, *Artificial Intelligence* **39**, 209–230, 1989.

[6] Hintikka, J., Model Minimization - An Alternative to Circumscription, *Journal of Automated Reasoning* **4**, 1–13, 1988.

[7] Lifschitz, V., Computing Circumscription, *Proceedings of 9th International Joint Conference on Artificial Intelligence*, 121–127, 1985.

[8] McCarthy, J., Circumscription - A Form of Non-monotonic Reasoning, *Artificial Intelligence* **13**, 81–132, 1980.

[9] McCarthy, J., Applications of Circumscription to Formalizing Common-Sense Knowledge, *Artificial Intelligence* **28**, 89–116, 1986.

[10] Olivetti, N., Tableaux and Sequent Calculus for Minimal Entailment, *Journal of Automated Reasoning* **9**, 99–139, 1992.

[11] Przymusinski, T., An Algorithm to Compute Circumscription, *Artificial Intelligence* **38**, 49–73, 1989.

[12] Robinson, J.A., A Machine-Oriented Logic Based on the Resolution Principle, *Journal of the Association for Computing Machinery* **12**, 23–41, 1965.

[13] Smullyan, R.M., *First-Order Logic*, Springer-Verlag, 1968.