Implementing Circumscription Using a Tableau Method

Ilkka Niemelä[1]

Abstract. A tableau calculus for first-order circumscriptive reasoning is developed. Parallel circumscription with fixed and varying predicates with respect to Herbrand models is treated. First a new clausal tableau calculus for first-order reasoning is developed where a hyper-type rule is combined with a restricted analytical cut rule. The use of a cut rule offers the advantages that when deciding logical consequence the space of counter-models is searched with a preference to (subset) minimal (Herbrand) models and each counter-model is not generated more than once. Then the calculus is extended to handle parallel circumscription. The circumscriptive calculus is sound in the general case and complete when no function symbols are allowed. Low space complexity is obtained by employing a groundedness property of minimal models that enables a one branch at a time approach to constructing tableaux for circumscriptive inference.

1 INTRODUCTION

We study the automation of first-order circumscriptive reasoning. Circumscription is a leading non-monotonic logic [6] and it has been applied to formalizing inheritance reasoning, actions, plans, and the semantics of logic programs. For example, in some of the most prominent semantics of disjunctive logic programs circumscriptive reasoning plays a key role [14].

A number of approaches have been proposed to implement circumscriptive inference [9, 7, 13, 12, 8, 10, 4, 1]. In some cases circumscription can be replaced by equivalent first-order formulas (see, e.g., [9, 4]). Some of the direct implementation approaches [8, 1] consider only a special subclass of circumscription which we call *minimal model reasoning* where there are no fixed and varying predicates and all predicates are minimized. The approaches [7, 10, 12] handling more general forms of circumscription are essentially restricted to the case where the Herbrand universe is finite. The most general existing approach appears to be that of Przymusinski [13]. He presents a resolution-based method for handling parallel circumscription in the ground case and then outlines how the method can be extended to cope with the non-ground case, too. However, the method has some computationally unattractive features and it seems to be unable to handle larger examples. This also holds for the other approaches [7, 10, 12, 8, 1] because of their exponential worst-case space requirements.

The goal of the paper is to develop a theorem proving method for a general form of circumscription with low space complexity. We restrict ourselves to Herbrand models and consider parallel circumscription [9] with both fixed and varying predicates.

As a basis of our approach we use analytic tableaux because tableau methods seem to be lend themselves to implementations of classical reasoning with restricted space complexity. In order to achieve low space complexity for circumscriptive reasoning we present a characterization of minimal models by which it can be decided whether a model is a minimal one independent of other models. This enables us to devise a tableau calculus for circumscription with restricted space complexity where the branches in the tableau can be constructed one at a time. The characterization is based on a groundedness property of minimal models which can be checked using a theorem prover.

Some problems related to the computational nature of classical cut-free analytic tableau [16] have been pointed out [3]. We present a first-order clausal tableau calculus which seems to be better-suited to circumscriptive reasoning than original cut-free analytic tableaux. The new calculus employs a restricted cut-type rule but is still analytic [16] in the sense that it obeys the subformula property. It explores the search space of counter-models with a preference to (subset) minimal (Herbrand) models and each counter-model is generated at most once. D'Agostino and Mondadori [3] have proposed a tableau calculus that also employs a cut rule but our usage of cut is more restricted.

This work is an extension of an earlier paper [11] where a similar tableau method is developed for propositional minimal model reasoning. The results obtained are quite encouraging and the performance of the method in solving propositional minimal model reasoning problems appears to be superior to other existing approaches [11]. This paper generalizes the earlier results in several respects. The earlier work [11] is restricted to the propositional case but here the first-order case is considered. In [11] only minimal model reasoning is treated where all predicates are minimized but here we allow general circumscription with both fixed and varying predicates. In [11] the closing condition for a branch is tested only for finished branches but here we devise a closing condition which can be tested for any branch.

Our approach differs considerably from related work on circumscriptive reasoning using tableau methods [12, 8, 1]. The methods presented in [8, 1] deal with minimal model reasoning where every predicate is minimized and [12] is essentially restricted to the case where the Herbrand universe is finite. Our approach is more general treating first-order circumscription with fixed and varying predicates. However, the main difference is in the technique employed to extend the underlying tableau calculus to handle circumscription. Our calculus enables a one branch at a time approach to constructing a tableau. This leads to low space complexity and enables us to handle cases involving non-finite minimal models. In the other approaches this is not possible and they have exponential worst-case space requirements even in the propositional case. In Olivetti's [12] approach a complete tableau is constructed and comparisons between different branches are performed. In the worst case a complete tableau can have exponential size. In the other two approaches [8, 1] a generated branch is compared against previously constructed branches. This leads to exponential worst-case space complexity and difficulties in handling first-order circumscription when function symbols are allowed. As general circumscription is highly undecidable [15],

[1] Dept. of Computer Science, University of Koblenz, Rheinau 1, D-56075 Koblenz, Germany, ini@informatik.uni-koblenz.de

ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

completeness cannot be expected in the general case but our approach is able to handle cases where the Herbrand universe is infinite. The computational properties of our approach differ considerably from those of Przymusinski's [13] resolution-based method for parallel circumscription. For example, for propositional Horn clauses Przymusinski's [13] method can lead to exponential computation but our method can handle minimal model reasoning for this class in polynomial time.

The paper is organized as follows. First we recall the definition of circumscriptive inference and present a characterization of minimal models that plays a key role in our tableau calculus for circumscription. In Section 3 we introduce the tableau calculus for first-order clauses that serves as the basis of the calculus for circumscription. Section 4 extends the calculus to handle circumscription. Section 5 discusses issues related to implementing the calculus. We finish with some concluding remarks.

2 CHARACTERIZING MINIMAL MODELS

In this section we recall the definition of minimal Herbrand models and circumscriptive inference and devise a characterization of minimal models that provides the basis for the computational treatment of circumscriptive inference to be developed.

In this paper we consider formulae in a clausal form (conjunctive normal form). We write a clause (a disjunction) with positive literals $a_1, \ldots a_m$ and negative literals $\neg b_1, \ldots \neg b_n$ as

$$a_1 \vee \cdots \vee a_m \leftarrow b_1 \wedge \cdots \wedge b_n. \tag{1}$$

Often $a_1 \vee \cdots \vee a_m$ is referred to as the *head* of the clause and $b_1 \wedge \cdots \wedge b_n$ as the *body* of the clause. As usual the variables in a clause are considered to be universally quantified.

We consider only *Herbrand interpretations* which can be taken as subsets of the Herbrand base $\mathcal{B}_\mathcal{L}$ of the language $\mathcal{L}$ in question. The Herbrand base is the set of ground instances of all the atomic formulae with respect to the set of ground terms in $\mathcal{L}$ [2].

We consider parallel circumscription [9] which is a general form of circumscription with a set of predicates P that are minimized, a set of predicates F that are fixed and with the rest of the predicates varying during minimization. The corresponding notion of $\langle P, F\rangle$-minimality can be defined using the notion of a set $\mathrm{sat}(I, P)$ of *satisfied* instances of predicates from the set P in a Herbrand interpretation I:

$$\mathrm{sat}(I, P) = \{p(\vec{t}) \in I \mid p \in P\}.$$

Definition 1 *Let I_1 and I_2 be two Herbrand interpretations and P and F two disjoint sets of predicates.*

(i) We say that $I_1 \leq_{\langle P,F\rangle} I_2$ iff

(a) $\mathrm{sat}(I_1, P) \subseteq \mathrm{sat}(I_2, P)$ *and*
(b) $\mathrm{sat}(I_1, F) = \mathrm{sat}(I_2, F)$.

We say that $I_1 <_{\langle P,F\rangle} I_2$ iff $I_1 \leq_{\langle P,F\rangle} I_2$ holds but $I_2 \leq_{\langle P,F\rangle} I_1$ does not.

(ii) A Herbrand interpretation I is a $\langle P, F\rangle$-minimal model of a set of clauses Σ iff I is a model of Σ and there is no model I' of Σ such that $I' <_{\langle P,F\rangle} I$.

(iii) A set of clauses Σ $\langle P, F\rangle$-entails a formula ϕ ($\Sigma \mid\!\sim_{\langle P,F\rangle} \phi$) iff ϕ is true in every $\langle P, F\rangle$-minimal model of Σ.

Example 1 Let $\Sigma = \{p(x) \leftarrow q(x)\}$ and let the underlying language have only one ground term a. Then the Herbrand base $\mathcal{B}_\mathcal{L} = \{p(a), q(a)\}$. Let $P = \{p\}$. If $F = \{q\}$, then $\Sigma \mid\!\sim_{\langle P,F\rangle} \neg p(a)$ does not hold but if $F = \{\}$, then $\Sigma \mid\!\sim_{\langle P,F\rangle} \neg p(a)$ holds. ■

As the basis for the computational treatment of circumscriptive inference we present a characterization of minimal models (Proposition 3). For that we use the concept of groundedness. A similar characterization but for the propositional case has been used by Eiter and Gottlob [5] in their study of the computational complexity of propositional circumscription.

Definition 2 *Let Σ be a set of clauses and P and F sets of predicates. A set of ground atoms Λ is said to be* grounded *in $\langle \Sigma, P, F\rangle$ iff for all $p(\vec{t}) \in \Lambda$, $p \in P$ implies $\Sigma \cup \mathrm{N}^{\langle P,F\rangle}(\Lambda) \models p(\vec{t})$ where*

$$\begin{aligned}\mathrm{N}^{\langle P,F\rangle}(\Lambda) \quad = \quad & \{\neg q(\vec{t}) \mid q \in P \cup F, q(\vec{t}) \in \mathcal{B}_\mathcal{L} - \Lambda\} \cup \\ & \{q(\vec{t}) \mid q \in F, q(\vec{t}) \in \Lambda\}\end{aligned}$$

Proposition 3 *A Herbrand interpretation I is a $\langle P, F\rangle$-minimal model of Σ iff I is a model of Σ and grounded in $\langle \Sigma, P, F\rangle$.*

Proof. ($\Rightarrow$) Let I be a $\langle P, F\rangle$-minimal model of Σ. Let $p(\vec{t}) \in I$ where $p \in P$. Assume $\Sigma \cup \mathrm{N}^{\langle P,F\rangle}(I) \not\models p(\vec{t})$. Hence, there is a model I' such that $I' \models \Sigma \cup \mathrm{N}^{\langle P,F\rangle}(I)$ but $I' \not\models p(\vec{t})$. Our aim is to show that $I' \leq_{\langle P,F\rangle} I$ holds. This leads to a contradiction with the fact that I is a $\langle P, F\rangle$-minimal model of Σ because $I \not\leq_{\langle P,F\rangle} I'$.

(i) Let $q \in P \cup F$ and $q(\vec{t}) \in I'$. Then $\neg q(\vec{t}) \notin \mathrm{N}^{\langle P,F\rangle}(I)$ and, hence, $q(\vec{t}) \in I$. (ii) Let $q \in F$ and $q(\vec{t}) \in I$. Then $q(\vec{t}) \in \mathrm{N}^{\langle P,F\rangle}(I)$ and, hence, $q(\vec{t}) \in I'$. By (i) and (ii) $I' \leq_{\langle P,F\rangle} I$, a contradiction. Thus, $\Sigma \cup \mathrm{N}^{\langle P,F\rangle}(I) \models p(\vec{t})$ holds. Hence, I is grounded.

($\Leftarrow$) Let I be a model of Σ which is grounded in $\langle \Sigma, P, F\rangle$. Assume that there is a model I' of Σ such that $I' <_{\langle P,F\rangle} I$. We show that $I \leq_{\langle P,F\rangle} I'$ holds. Let $p(\vec{t}) \in I$ such that $p \in P$. Then $\Sigma \cup \mathrm{N}^{\langle P,F\rangle}(I) \models p(\vec{t})$ holds. We show that

$$I' \models \mathrm{N}^{\langle P,F\rangle}(I) \tag{2}$$

holds. Let $\neg q(\vec{t}) \in \mathrm{N}^{\langle P,F\rangle}(I)$. Then $q \in P \cup F$ and $q(\vec{t}) \notin I$. Hence, $q(\vec{t}) \notin I'$ as $I' <_{\langle P,F\rangle} I$. Let $q(\vec{t}) \in \mathrm{N}^{\langle P,F\rangle}(I)$. Then $q \in F$ and $q(\vec{t}) \in I$. Hence, $q(\vec{t}) \in I'$ as $I' <_{\langle P,F\rangle} I$. Thus, (2) holds.

Now (2) implies $p(\vec{t}) \in I'$ because $I' \models \Sigma$ and $\Sigma \cup \mathrm{N}^{\langle P,F\rangle}(I) \models p(\vec{t})$. This means that $I \leq_{\langle P,F\rangle} I'$ holds, a contradiction. So I is a $\langle P, F\rangle$-minimal model of Σ. □

Example 2 Let Σ be as in Example 1 and $P = \{p\}$. If $F = \{q\}$, then $I = \{p(a), q(a)\}$ is grounded in $\langle \Sigma, P, F\rangle$ because $\mathrm{N}^{\langle P,F\rangle}(I) = \{q(a)\}$ and $\Sigma \cup \mathrm{N}^{\langle P,F\rangle}(I) \models p(a)$ holds. If $F = \{\}$, then I is not grounded in $\langle \Sigma, P, F\rangle$ because $\mathrm{N}^{\langle P,F\rangle}(I) = \{\}$ and $\Sigma \cup \mathrm{N}^{\langle P,F\rangle}(I) \models p(a)$ does not hold. ■

3 A TABLEAU CALCULUS WITH A CUT

In this section we introduce a tableau calculus for clauses that serves as the basis of our calculus for circumscriptive inference. To construct a tableau for a set of *input clauses* Υ we start from an empty initial branch and use two rules for extending a branch B in the tableau.

H: If $a_1 \vee \cdots \vee a_m \leftarrow b_1 \wedge \cdots \wedge b_n \in \Upsilon$ and for one of its ground instances $a'_1 \vee \cdots \vee a'_m \leftarrow b'_1 \wedge \cdots \wedge b'_n$,

$$\{b'_1, \ldots, b'_n, \neg a'_1, \ldots, \neg a'_{j-1}, \neg a'_{j+1}, \ldots, \neg a'_m\} \subseteq B$$

but $a'_j \notin B$, then extend B by a'_j.

C: If $a_1 \vee \cdots \vee a_m \leftarrow b_1 \wedge \cdots \wedge b_n \in \Upsilon$ and for one of its ground instances $a'_1 \vee \cdots \vee a'_m \leftarrow b'_1 \wedge \cdots \wedge b'_n$, $\{b'_1, \ldots, b'_n\} \subseteq B$ but $\{a'_1, \ldots, a'_m\} \cap B = \emptyset$ and $\neg a'_j \notin B$ for some $j \in \{1, \ldots, m\}$, then extend B to two branches $B \cup \{\neg a'_j\}$ and $B \cup \{a'_j\}$.

We say that a rule is *applicable* on a branch B for a ground instance of a clause if the corresponding conditions given above for the rule in question are satisfied. The first rule is a restricted hyper-type rule which says that if a branch contains the body literals $(b_1, \ldots, b_n)$ of a ground instance of a clause and the negations of the head literals except possibly not that of a_j $(\neg a_1, \ldots, \neg a_{j-1}, \neg a_{j+1}, \ldots, \neg a_m)$, then the branch can be extended by a_j. The second rule is a limited cut-type rule which says that if a branch contains the body literals $(b_1, \ldots, b_n)$ of a ground instance of a clause, then for some head literal a_j, the branch can be extended into two branches containing $\neg a_j$ and a_j, respectively.

Definition 4 *(i) In a tableau for Υ a branch is said to be* closed *if it contains a complementary pair of literals or there is a ground instance $\leftarrow b_1 \wedge \cdots \wedge b_n$ of a clause in Υ such that $b_1, \ldots, b_n$ are on the branch.*

(ii) If a branch is not closed, it is said to be open. *A tableau is open if it has an open branch otherwise it is closed.*

(iii) A branch is said to be finished *when for every ground instance $a_1 \vee \cdots \vee a_m \leftarrow b_1 \wedge \cdots \wedge b_n$ of a clause in Υ, if $b_1, \ldots, b_n$ are on the branch, a_j is on the branch for some $j \in \{1, \ldots, m\}$.*

(iv) A tableau is finished if it is infinite or all of its branches are finished or closed.

Example 3 Below is a tableau for a set of clauses

$$\Sigma = \{p(x) \vee q(x) \vee r(x) \leftarrow, \leftarrow q(a), p(x) \leftarrow r(x)\}$$

where the underlying language contains only one ground term a.

$$\begin{array}{c|c} \begin{array}{c} \neg p(a) \\ \begin{array}{c|c} \neg q(a) & q(a) \\ r(a) & \times \\ p(a) & \\ \times & \end{array} \end{array} & \begin{array}{c} p(a) \\ \\ \\ \\ \\ \end{array} \end{array}$$

First rule (**C**) is applied to the first clause with respect to $p(a)$. Notice that now the right most branch is already finished and we do not extend the branch anymore. Then rule (**C**) is applied to the first clause with respect to $q(a)$. The resulting branch with $q(a)$ is closed as $\leftarrow q(a)$ is one of the input clauses and we do not extend a closed branch either. Now rule (**H**) is applied on the leftmost branch two times: first for the first clause and then for the last clause. This leads to a closed branch. Now all the branches in the tableau are either finished or closed. ■

For function-free clauses where the Herbrand base is finite this calculus provides a sound and complete decision procedure because for each clause there is only finitely many ground instances for which the two rules can be applied and, hence, the tableaux are finite. To ensure completeness in the general case where function symbols are allowed, we need a *systematic procedure* (cf. [16]) for building the tableau to ensure that all the possibilities for closing the branches are really covered. The essential requirement for the procedure (Systematic Procedure Assumption) can be stated as follows:

SPA: If one of the rules is applicable on a branch B for a ground instance $a_1 \vee \cdots \vee a_m \leftarrow b_1 \wedge \cdots \wedge b_n$ of a clause in Υ, then each extension of B that remains open must contain one of $\{a_1, \ldots, a_m\}$.

This assumption rules out tableaux where an applicable clause is ignored infinitely often. Consider, for example, a set of clauses $\Upsilon = \{\leftarrow q(a), p(a) \leftarrow, p(f(x)) \leftarrow p(x), q(x) \leftarrow p(x)\}$. If a tableau for Υ is built using a procedure satisfying (**SPA**), then the tableau cannot contain the branch $B = \{p(a), p(f(a)), p(f(f(a))), \ldots\}$ where the last clause is never used. This is because after applying rule (**H**) to the second clause, the rule is applicable for $q(a) \leftarrow p(a)$ but $q(a)$ is not contained in B which contradicts **SPA**. We discuss the problem of implementing a procedure satisfying (**SPA**) in Section 5.

It can be proved that this calculus is sound and complete in the general case when the tableaux are constructed using a procedure satisfying (**SPA**). The soundness proof is based on the following observation that for every model of the input clauses, there is an open branch with which the model agrees. An interpretation I is said to *agree* with a branch B if for any ground atom a, (i) $a \in B$ implies $I \models a$ and (ii) $\neg a \in B$ implies $I \not\models a$.

Proposition 5 *For every (Herbrand) model M of a set of clauses Υ, there is an open branch with which M agrees in any tableau for Υ.*

Proof. We can prove this by observing that the proposition holds initially and showing that it holds after each extension step of the tableau if it holds before the step. So let there be an open branch B with which the model agrees and assume that we are extending B.

(i) If the branch is extended by rule (**H**), then there is a ground instance $a_1 \vee \cdots \vee a_m \leftarrow b_1 \wedge \cdots \wedge b_n$ of a clause in Υ such that $\{b_1, \ldots, b_n, \neg a_1, \ldots, \neg a_{j-1}, \neg a_{j+1}, \ldots, \neg a_m\} \subseteq B$ and the branch is extended by a_j. Now $M \models a_j$ because otherwise $M \models b_1 \wedge \cdots \wedge b_n$ but for all $i \in \{1, \ldots, m\}$, $M \not\models a_i$ implying that M is not a model of Υ.

(ii) If the branch is extended by rule (**C**), then for some ground atom a_j the branch is extended to two branches B_1 and B_2 containing $\neg a_j$ and a_j, respectively. If $M \models a_j$, then M agrees with B_2 and otherwise with B_1.

Hence, in a tableau for Υ there is an branch B with which M agrees. This branch cannot contain a complementary pair of literals a and $\neg a$ because then $M \models a$ and $M \not\models a$. There cannot be a ground instance $\leftarrow b_1 \wedge \cdots \wedge b_n$ of a clause in Υ with $\{b_1, \ldots, b_n\} \subseteq B$ as then M would not be a model of Υ. Hence, the branch is open. □

Theorem 6 (Soundness and completeness) *Let Υ be a set of clauses.*

(i) If there is a closed tableau for Υ, then Υ is unsatisfiable.

(ii) If Υ is unsatisfiable, then any finished tableau for Υ is closed.

Proof. (i) (Soundness) Assume that Υ has a model. Then it has a Herbrand model and by Proposition 5 there is an open branch in any tableau for Υ.

(ii) (Completeness) Assume that there is a finished tableau for Υ that has an open (possibly infinite) branch B. We show that the set of atoms $\hat{B}$ on the branch B is a model of Υ.

Let $a_1 \vee \cdots \vee a_m \leftarrow b_1 \wedge \cdots \wedge b_n$ be a ground instance of a clause in Υ and $\hat{B} \models b_1 \wedge \cdots \wedge b_n$. Then there is a finite set of literals B' such that the branch B is an extension of the branch B' and $\{b_1, \ldots, b_n\} \subseteq B'$. If for some $j \in \{1, \ldots, m\}$, $a_j \in B'$, then $a_j \in B$. Otherwise rule (**C**) is applicable and by (**SPA**) B contains one of $\{a_1, \ldots, a_m\}$. So in either case, for some $j \in \{1, \ldots, m\}$, $\hat{B} \models a_j$ and hence, $\hat{B} \models a_1 \vee \cdots \vee a_m \leftarrow b_1 \wedge \cdots \wedge b_n$ holds.

Let $\leftarrow b_1 \wedge \cdots \wedge b_n$ be a ground instance of a clause in Υ, then $\hat{B} \not\models b_1 \wedge \cdots \wedge b_n$ because otherwise the branch would be closed. Hence, $\hat{B}$ is a model of all the ground instances of the clauses in Υ. Thus, $\hat{B}$ is a model of Υ and Υ is satisfiable. □

4 A CALCULUS FOR CIRCUMSCRIPTION

In this section we present an extension of the tableau calculus introduced in Section 3 that is capable of performing circumscriptive inference, i.e., capable of establishing that $\Sigma_O \mid\!\sim_{\langle P,F\rangle} \phi$ holds for a set of first-order clauses Σ_O and for a first-order formula ϕ. The tableau method for circumscriptive inference starts by taking Σ_O and $\neg\phi$ in clausal as premises. Let us call this set of clauses Υ. The two rules (**H**) and (**C**) are used to construct a classical tableau for Υ. If the tableau closes, then ϕ follows classically from Σ_O and, of course, also $\Sigma_O \mid\!\sim_{\langle P,F\rangle} \phi$ holds. However, if the tableau has some open branches, an extension of the calculus is clearly needed because circumscriptive inference is stronger than classical: there are formulae ϕ for which $\Sigma_O \mid\!\sim_{\langle P,F\rangle} \phi$ holds although ϕ is not a classical consequence of Σ_O. For example, $\{p(a)\} \mid\!\sim_{\langle P,F\rangle} \neg q(a)$ holds where $P = \{q\}$.

Proposition 3 provides a way to strengthen the classical tableau calculus in an appropriate manner. The idea is the following. An open branch in the tableau for Υ provides a model for Υ, i.e., a counter-model for classical consequence where Σ_O is true but ϕ is false. However, this does not show that $\Sigma_O \mid\!\sim_{\langle P,F\rangle} \phi$ does not hold because the model might not be $\langle P,F\rangle$-minimal. So when closing branches, we can consider branches corresponding to non-$\langle P,F\rangle$-minimal models also closed. Next we devise a closing condition for branches that can be tested for any branch at any stage during the construction of a tableau (and not only for, e.g., finished branches). For stating the condition we need the concept of a *fully determined predicate* q on a branch which means that for every $q(\vec{t}) \in \mathcal{B}_{\mathcal{L}}$ either $q(\vec{t})$ or $\neg q(\vec{t})$ is on the branch.

Definition 7 *(i) A branch B is said to be* ungrounded *w.r.t.* $\langle \Sigma_O, P, F\rangle$ *when there is a ground atom $p(\vec{t})$ in B such that $p \in P$ and*

$$\Sigma_O^* \cup \mathrm{N}^*(B) \not\models p(\vec{t})$$

where Σ_O^* is obtained from Σ_O by removing from every clause in Σ_O every literal $(\neg)q(\vec{t})$ where $q \in F$ and q is not fully determined on the branch B and

$$\begin{aligned} \mathrm{N}^*(B) \quad = \quad & \{\neg q(\vec{t}) \mid q \in P, q(\vec{t}) \in \mathcal{B}_{\mathcal{L}} - B\} \cup \\ & \{\neg q(\vec{t}) \in B \mid q \in F\} \cup \{q(\vec{t}) \in B \mid q \in F\}. \end{aligned}$$

*(ii) We call a branch $\langle \Sigma_O, P, F\rangle$-*closed *if it is (classically) closed or ungrounded w.r.t. $\langle \Sigma_O, P, F\rangle$.*

Hence, a tableau proof for $\Sigma_O \mid\!\sim_{\langle P,F\rangle} \phi$ is an $\langle \Sigma_O, P, F\rangle$-closed tableau for $\Sigma_O \cup \{\neg\phi\}$.

Example 4 Consider the following set of clauses

$$\Sigma_O = \{p(x) \vee q(x) \leftarrow, p(f(x)) \leftarrow p(x)\}$$

where the Herbrand base of the underlying language is infinite. Let $P = \{p\}$ and $F = \{\}$. Let us use the tableau calculus to establish that $\Sigma_O \mid\!\sim_{\langle P,F\rangle} \neg p(a)$ holds. Consider a tableau for $\Sigma_O \cup \{p(a) \leftarrow\}$ with a single open branch $B = \{p(a)\}$. Then $\Sigma_O^* = \Sigma_O$ and $\mathrm{N}^*(B) = \{\neg p(t) \mid p(t) \in \mathcal{B}_{\mathcal{L}}, t \neq a\}$. Hence, $\Sigma_O^* \cup \mathrm{N}^*(B) \not\models p(a)$ holds and B is ungrounded w.r.t. $\langle \Sigma_O, P, F\rangle$. Thus, we have found a tableau proof for $\Sigma_O \mid\!\sim_{\langle P,F\rangle} \neg p(a)$ even when the Herbrand base is infinite.

On the other hand, if $F = \{q\}$, then $\Sigma_O^* = \{p(x) \leftarrow, p(f(x)) \leftarrow p(x)\}$. Hence, $\Sigma_O^* \cup \mathrm{N}^*(B) \models p(a)$ holds and B is not ungrounded w.r.t. $\langle \Sigma_O, P, F\rangle$ when $F = \{q\}$. ■

The fundamental property guaranteeing soundness is stated in the following proposition and can be proved using Proposition 3.

Proposition 8 *Let Υ be a set of clauses containing Σ_O. If a branch B is $\langle \Sigma_O, P, F\rangle$-closed in a tableau for Υ, there is no $\langle P,F\rangle$-minimal model of Σ_O that is a model of Υ and agrees with B.*

Proof. Assume that there is a $\langle P,F\rangle$-minimal model M of Σ_O that is a model of Υ and agrees with a branch B. Then B cannot be (classically) closed. We show that B is not ungrounded w.r.t. $\langle \Sigma_O, P, F\rangle$ implying that B is not $\langle \Sigma_O, P, F\rangle$-closed. This can be proved by showing that for all $p \in P$,

$$\Sigma_O \cup \mathrm{N}^{\langle P,F\rangle}(M) \models p(\vec{t}) \text{ implies } \Sigma_O^* \cup \mathrm{N}^*(B) \models p(\vec{t}) \qquad (3)$$

which by Proposition 3 implies that B is not ungrounded w.r.t. $\langle \Sigma_O, P, F\rangle$. Assume that there is a Herbrand interpretation I where $\Sigma_O^* \cup \mathrm{N}^*(B)$ is true and $p(\vec{t})$ false. Let

$$\begin{aligned} I' \quad = \quad & I - \{q(\vec{t}) \in \mathcal{B}_{\mathcal{L}} \mid q \in F, q(\vec{t}) \notin M\} \cup \\ & \{q(\vec{t}) \in \mathcal{B}_{\mathcal{L}} \mid q \in F, q(\vec{t}) \in M\}. \end{aligned}$$

Now $I' \models \mathrm{N}^{\langle P,F\rangle}(M)$ and $I' \not\models p(\vec{t})$. It can be proved quite straightforwardly that also $I' \models \Sigma_O$ holds. Hence, (3) holds. □

Notice that the resulting calculus is somewhat unusual because the closing condition for a branch involves a test for classical satisfiability. It can be shown that this calculus is sound.

Theorem 9 (Soundness) *Let Σ_O be a set of clauses, ϕ a formula and Υ a set of clauses equivalent to $\Sigma_O \cup \{\neg\phi\}$. If there is a $\langle \Sigma_O, P, F\rangle$-closed tableau for Υ, then $\Sigma_O \mid\!\sim_{\langle P,F\rangle} \phi$ holds.*

Proof. Assume that $\Sigma \mid\!\sim_{\langle P,F\rangle} \phi$ does not hold. Then there is a $\langle P,F\rangle$-minimal model M of Σ_O where ϕ is not true. Then M is a model of Υ. By Proposition 5 there is an open branch B with which M agrees. By Proposition 8 the branch cannot be $\langle \Sigma_O, P, F\rangle$-closed. □

Circumscription is highly undecidable in the general case [15] but completeness can be achieved when the underlying first-order fragment is decidable. This is the case, e.g., when function symbols are not allowed. In this case the calculus is complete if there are no fixed predicates. To achieve completeness in the presence of fixed predicates we make the cut rule applicable for fixed predicates.

Cf: If $q(\vec{t}) \in \mathcal{B}_{\mathcal{L}}, q \in F$ but $q(\vec{t}) \notin B$ and $\neg q(\vec{t}) \notin B$ holds, then extend B to two branches $B \cup \{\neg q(\vec{t})\}$ and $B \cup \{q(\vec{t})\}$.

Furthermore, we tighten the finishedness condition. We call a branch finished when it is finished in the original sense (Definition 4 (iii)) and every fixed predicate is fully determined on the branch. With these extensions the calculus can be shown to be complete when there are no function symbols but when fixed predicates are allowed.

Theorem 10 (Completeness) *Let Σ_O be a set of clauses ϕ a formula and Υ a set of clauses equivalent to $\Sigma_O \cup \{\neg\phi\}$ with no function symbols. If $\Sigma_O \mid\!\sim_{\langle P,F\rangle} \phi$ holds, then any finished tableau for Υ is $\langle \Sigma_O, P, F\rangle$-closed.*

Proof. As there are no function symbols, the Herbrand base is finite and every finished tableau is finite. Assume that a finished tableau for Υ has a finished branch B which is not $\langle \Sigma_O, P, F\rangle$-closed. Now the set of atoms $\hat{B}$ on the branch B is a model of Υ because B is open and finished. Hence, $\hat{B} \models \Sigma_O \cup \{\neg\phi\}$ holds.

As B is not ungrounded w.r.t. $\langle \Sigma_O, P, F\rangle$, for every $p(\vec{t}) \in B$, $p \in P$ implies $\Sigma_O^* \cup \mathrm{N}^*(B) \models p(\vec{t})$. As B is finished, every $q \in F$ is fully determined in B. Hence, $\Sigma_O^* = \Sigma_O$ holds. The fact that every $q \in F$ is fully determined in B implies also

$$\mathrm{N}^*(B) = \mathrm{N}^{\langle P,F\rangle}(\hat{B}). \qquad (4)$$

Clearly, $N^*(B) \subseteq N^{\langle P,F\rangle}(\hat{B})$ holds. If $q(\vec{t}) \in N^{\langle P,F\rangle}(\hat{B})$, then $q \in F$ and $q(\vec{t}) \in B$. Hence, $q(\vec{t}) \in N^*(B)$. Let $\neg q(\vec{t}) \in N^{\langle P,F\rangle}(\hat{B})$. Then $q(\vec{t}) \notin B$. If $q \in P$, $\neg q(\vec{t}) \in N^*(B)$. If $q \in F$, then as q is fully determined in B, $\neg q(\vec{t}) \in B$ and $\neg q(\vec{t}) \in N^*(B)$. Thus, $N^{\langle P,F\rangle}(\hat{B}) \subseteq N^*(B)$ and (4) holds.

Hence, for every $p(\vec{t}) \in B, p \in P$ implies $\Sigma_O \cup N^{\langle P,F\rangle}(B) \models p(\vec{t})$ and B is grounded in $\langle \Sigma_O, P, F\rangle$. Then by Proposition 3 $\hat{B}$ is a $\langle P, F\rangle$-minimal model of Σ_O and, thus, $\Sigma_O \mid\!\sim_{\langle P,F\rangle} \phi$ does not hold. □

5 TOWARDS IMPLEMENTATION

In this section we discuss implementation of the circumscriptive tableau calculus. First we note that the calculus can be implemented using a "one branch at a time" approach (constructing one branch until it is finished or closed) because the closing condition of a branch can be tested independent of other branches. This leads to low space complexity as previously constructed branches need not to be stored and the space requirements are dominated by the space needed to test the ungroundedness condition, i.e., to test classical satisfiability.

Concerning ungroundedness tests we note the following. (i) Given a branch B, actually only one satisfiability test is needed to verify whether the branch is ungrounded, i.e., a branch B is ungrounded w.r.t. $\langle \Sigma_O, P, F\rangle$ iff $\Sigma_O^* \cup N^*(B) \cup \{\leftarrow p_1(\vec{t_1}) \wedge \cdots \wedge p_k(\vec{t_k})\}$ is satisfiable where $p_1(\vec{t_1}), \ldots, p_k(\vec{t_k})$ are the positive atoms on the branch for which the predicate $p_i \in P$. (ii) The set of ground literals $N^*(B)$ employed in testing ungroundedness often restricts the required satisfiability test considerably as ground literals can be used to prune the search space. Note that the set $N^*(B)$ is infinite if the Herbrand base is infinite. Hence, it cannot be represented explicitly but it can be handled indirectly using the branch in question which is always finite. For example, in a tableau prover the set $N^*(B)$ can be handled as follows: all ground literals $(\neg)q(\vec{t}) \in B$ with $q \in F$ are taken as normal premises and for the rest of the literals $\neg q(\vec{t}) \in N^*(B)$, an additional closing condition is adopted saying that a branch containing $q(\vec{t})$ such that $q \in P$ and $q(\vec{t}) \notin B$ can be closed.

To achieve completeness for classical reasoning in the first-order case with function symbols we must ensure that the implementation satisfies the systematic procedure assumption (**SPA**). There are various ways of achieving this: one can maintain a queue of applicable instances of clauses for each branch and always apply the first applicable clause in the queue and after extending the branch add new clauses that became applicable to the end of the queue. One can also impose a bound on the complexity of the terms appearing in the instances of the clauses used for extending the branch and increase the bound when the terms of lower complexity have been exhausted.

An important issue is the heuristic to choose the next applicable rule. It is advantageous to apply the hyper-type rule (**H**) whenever possible as it does not branch the tableau. To support the idea of jumping to conclusions, the ungroundedness test can be preferred to rule (**C**). Then a tableau can be closed by the ungroundedness condition without constructing a complete classical tableau. In this setting circumscription can lead to speed-ups in reasoning as the satisfiability test for establishing ungroundedness can be computationally easier than the original theorem proving tasks: (i) the set of premises is reduced, i.e. Σ_O^* is used instead of Σ_O, and (ii) a large number of ground literals ($N^*(B)$) are provided as additional premises.

6 CONCLUSIONS

We have presented a clausal tableau calculus for first-order reasoning and extended it to handle general circumscription with respect to Herbrand models. Our tableau calculus employs an analytical cut rule which implies that when deciding logical consequence, the search space of counter-models is explored with a preference to (subset) minimal (Herbrand) models and each counter-model is generated at most once. The extension to handle circumscription is devised in such a way that tableaux can be constructed one branch at a time. This enables implementations with low space complexity. Furthermore, the calculus is sound in the general case and can handle examples where the Herbrand universe is infinite.

ACKNOWLEDGMENTS

The author is grateful to Chandrabose Aravindan, Peter Baumgartner, Ulrich Furbach, Katrin Erk, and Michael Kühn for their helpful comments and discussions.

REFERENCES

[1] F. Bry and A. Yahya, 'Minimal model generation with positive unit hyper-resolution tableaux', in *Proceedings of the Fifth Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, pp. 143–159, Terrasini, Italy, (May 1996). Springer-Verlag.

[2] C. Chang and R. Lee, *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, New York, 1973.

[3] M. D'Agostino and M. Mondadori, 'The taming of the cut', *Journal of Logic and Computation*, **4**, 285–319, (1994).

[4] Patrick Doherty, Witold Lukaszewicz, and Andrzej Szalas, 'Computing circumscription revisited: Preliminary report', in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp. 1502–1508, Montreal, Canada, (August 1995). Morgan Kaufmann Publishers.

[5] T. Eiter and G. Gottlob, 'Propositional circumscription and extended closed world reasoning are Π_2^p-complete', *Theoretical Computer Science*, **114**, 231–245, (1993).

[6] *Handbook of Logic and Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning*, eds., D.M. Gabbay, C.J. Hogger, and J.A. Robinson, Oxford University Press, Oxford, 1994.

[7] M.L. Ginsberg, 'A circumscriptive theorem prover', *Artificial Intelligence*, **39**, 209–230, (1989).

[8] K. Inoue, M. Koshimura, and R. Hasegawa, 'Embedding negation as failure into a model generation theorem prover', in *The 11th International Conference on Automated Deduction*, pp. 400–415, Saratoga Springs, NY, USA, (June 1992). Springer-Verlag.

[9] V. Lifschitz, 'Computing circumscription', in *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pp. 121–127, Los Angeles, California, USA, (August 1985). Morgan Kaufmann Publishers.

[10] A. Nerode, R.T. Ng, and V.S. Subrahmanian, 'Computing circumscriptive databases: I. theory and algorithms', *Information and Computation*, **116**, 58–80, (1995).

[11] I. Niemelä, 'A tableau calculus for minimal model reasoning', in *Proceedings of the Fifth Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, pp. 278–294, Terrasini, Italy, (May 1996). Springer-Verlag.

[12] N. Olivetti, 'A tableaux and sequent calculus for minimal entailment', *Journal of Automated Reasoning*, **9**, 99–139, (1992).

[13] T.C. Przymusinski, 'An algorithm to compute circumscription', *Artificial Intelligence*, **38**, 49–73, (1989).

[14] T.C. Przymusinski, 'Static semantics for normal and disjunctive logic programs', *Annals of Mathematics and Artificial Intelligence, Special Issue on Disjunctive Programs*, (1995). To appear.

[15] J.S. Schlipf, 'How uncomputable is general circumscription?', in *Proceedings of the Symposium on Logic in Computer Science*, pp. 92–95, Cambridge, USA, (June 1986). IEEE Computer Society Press.

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