The Even More Liberalized δ–Rule in Free Variable Semantic Tableaux

Bernhard Beckert, Reiner Hähnle, Peter H. Schmitt

Institute for Logic, Complexity and Deduction Systems
University of Karlsruhe
Am Fasanengarten 5, 7500 Karlsruhe
Federal Republic of Germany
{beckert,haehnle,pschmitt}@ira.uka.de

Abstract. In this paper we have a closer look at one of the rules of the tableau calculus presented in [3], called the δ–rule, and the modification of this rule, that has been proved to be sound and complete in [6], called the δ^+–rule, which uses fewer free variables. We show that an even more liberalized version, the δ^{++}–rule, that in addition reduces the number of different Skolem–function symbols that have to be used, is also sound and complete. Examples show the relevance of this modification for building tableau–based theorem provers.

Introduction

The most popular version of the proof procedure which is usually called Analytic Tableaux or Semantic Tableaux is due to Raymond Smullyan [8] and goes back to Beth and Hintikka. Semantic tableaux have recently experienced a renewed interest by AI researchers, since their closeness to the semantic definitions of logical operators makes the basic system easily adjustable to a wide scope of non–standard logics. For example, in [2] tableaux are extended to cover first–order modal logic and in [4] to many–valued logics. Areas of application include Natural Language Processing, Non–Monotonic Reasoning and Logic Programming, just to name a few. The present paper is only concerned with quantifier rules in classical predicate logic, but the results are equally applicable to non–standard first–order tableaux systems.

We assume that the reader is familiar with the method of semantic tableaux (if not, excellent introductions can be found in [8] and [3]). Let us just recall that Smullyan introduced *unified notation*, a classification scheme for logical operators (and thus for tableau rules) that makes definitions and proofs clearer and much more compact. According to this scheme, there are four types of operators, namely α (conjunctive propositional), β (disjunctive propositional), γ (universal quantifiers) and δ (existential quantifiers) with corresponding rules. Semantic tableaux for classical logic come in two versions, signed and unsigned, from which we choose the latter. In Table 1 we have summarized γ– and δ–type formulas.

Table 1. γ– and δ–type formulas

γ	$\gamma_0(t)$
$(\forall x)\phi(x)$	$\phi(t)$
$\neg(\exists x)\phi(x)$	$\neg\phi(t)$

δ	$\delta_0(t)$
$\neg(\forall x)\phi(x)$	$\neg\phi(t)$
$(\exists x)\phi(x)$	$\phi(t)$

1 Free Variable Tableaux

In Smullyan's formulation the γ–rule requires the substitution of an arbitrary but fixed term[1] for the quantified variable, see Table 2. Since this "guess" may be wrong, the γ–rule may have to be applied again and again to the same universal type formula in a tableau proof. Obviously, this indeterminism can make proofs very long and it is a natural idea to postpone the instantiation in a γ–rule until more information on the instance actually needed has been collected. We know of two approaches in the literature where this has been expressed formally [7, 3]. We concentrate on the latter, which we assume the reader to be familiar with.

For convenience, we have given the free quantifier rules from [3] in Table 3.

Table 2. Ground tableau rules for quantified formulas.

$$\frac{\gamma}{\gamma_0(t)}$$

where t is any ground term.

$$\frac{\delta}{\delta_0(t)}$$

where t is a ground term not occurring on the current branch.

Table 3. Free tableau rules for quantified formulas.

$$\frac{\gamma}{\gamma_0(x)}$$

where x is a free variable.

$$\frac{\delta}{\delta_0(f(x_1,\ldots,x_n))}$$

where $x_1,\ldots,x_n$ are the free variables occurring on the current branch and f is a new function symbol.

1 Smullyan did not include function symbols in his first–order language, so in his case constants were the only ground terms. We assure the reader that in the extended language all results are still valid and the proofs may be adopted without any problems.

The proviso of the δ–rule ensures that the introduced Skolem term is new on the branch constructed so far, even when the free variables are instantiated later during the proof. Thus it can be safely given an appropriate meaning in order to preserve satisfiability of tableaux after δ–rule applications.

Let us henceforth call the tableau system with these rules the *free* version and the old one the *ground* version.

2 The Liberalized δ^+–Rule

Both versions of tableaux systems, free and ground, have essentially the same proviso in the δ–rule: under any substitutions, the introduced term has to be absolutely new on the current branch.

Hähnle and Schmitt showed in [6] that this proviso is somewhat stronger than is actually needed and formulated the *liberalized free δ–rule* stated in Table 4, which they called δ^+–rule.

To show the possible advantage of a system using δ^+ over one using δ, here is an example of a tableau proof using Fitting's δ–rule:

(1) $\neg(\exists x)(((\forall z)p(z)) \vee \neg p(x))$

(2) $\neg(((\forall z)p(z)) \vee \neg p(x_1))$

(3) $\neg(\forall z)p(z)$

(4) $\neg\neg p(x_1)$

(5) $p(x_1)$

(6) $\neg p(f(x_1))$

(7) $\neg(((\forall z)p(z)) \vee \neg p(x_2))$

(8) $\neg(\forall z)p(z)$

(9) $\neg\neg p(x_2)$

(10) $p(x_2)$

closed by (10) and (6)

Line (6) is obtained from line (3) by Fitting's δ–rule. It is not possible to close the tableau by using lines (6) and (5). Only after a second application of the γ-rule on the formula in line (1) resulting in line (7) closure can be obtained. A closer look at this proof reveals that it is in fact the shortest possible proof using the δ–rule. The same root formula yields the following tableau using the δ^+–rule:

(1) $\neg(\exists x)(((\forall z)p(z)) \vee \neg p(x))$

(2) $\neg(((\forall z)p(z)) \vee \neg p(x_1))$

(3) $\neg(\forall z)p(z)$

(4) $\neg\neg p(x_1)$

(5) $p(x_1)$

(6) $\neg p(c)$

closed by (5) and (6)

Using the δ^+–rule instead of the δ–rule can shorten tableau proofs exponentially. Since the system using the free δ–rule is complete, a system using the liberalized free rule δ^+ will also be complete. The problem thus lies in proving correctness of the δ^+–rule. A proof has been given in [6].

Table 4. Liberalized free tableau rules for quantified formulas, introduced in [6].

$\dfrac{\gamma}{\gamma_0(x)}$	$\dfrac{\delta}{\delta_0(f(x_1,\ldots,x_n))}$
where x is a free variable.	where $x_1,\ldots,x_n$ are the free variables occurring in δ and f is a new function symbol.

3 The Even More Liberalized δ^{++}–Rule

It is possible to liberalize the δ^+–rule even more. This additional liberalization does not effect the number of variables used as arguments for the Skolem–function that is introduced by a δ–rule application, but it effects the Skolem–function symbol itself. The restriction that the function symbol has to be *new* is weakened.[2] The same function symbol is used more than once when the δ^{++}–rule is applied to δ–formulas that are identical up to variable renaming.

Definition 1 Signature, Language. A signature $\Sigma = \langle P_\Sigma, F_\Sigma \rangle$ consists of a non–empty set P_Σ of predicate symbols and a set F_Σ of function symbols.[3]

$\mathcal{L}^\Sigma$ denotes the first–order language over Σ, i.e., the set of well–formed formulas over Σ.

[2] Smullyan introduced a δ–rule liberalized in this way for the *ground version* of tableaux: The same Skolem–function symbol may be used on different branches. In the ground version, however, this is easy to see; and it is not as liberal as our rule.

[3] Constants are functions of arity 0.

The following definition describes which δ–formulas are assigned the same Skolem–function symbol. In addition, for a given signature Σ a new signature $\Sigma^\star$ is defined. $\Sigma^\star$ contains all Skolem–function symbols that can occur during the construction of a tableau for a formula over Σ. After n induction steps the new signature contains all Skolem–function symbols that are needed for the Skolemization of a formula $\phi \in \mathcal{L}^\Sigma$ provided the maximal nesting of existential quantifiers in ϕ is not greater than n.

The new δ^{++}–rule is stated in Table 5.

Definition 2 Assigned Skolem–function, Signature $\Sigma^\star$, Rank. Let $\Sigma = \langle P_\Sigma, F_\Sigma \rangle$ be a signature. For each δ–formula $\delta \in \mathcal{L}^\Sigma$, the equivalence class $[\delta]$ is the set of all formulas that are identical to δ up to variable renaming (including renaming of the bound variables in δ).

All formulas in an equivalence class $[\delta]$ are assigned the same unique Skolem–function symbol $f_{[\delta]} \notin \Sigma$.

The signature Σ_{sk} is defined by

$$\Sigma_{sk} = \langle P_\Sigma, F_\Sigma \cup \{ f_{[\delta]} \ : \ \delta \in \mathcal{L}^\Sigma, \delta \text{ is a } \delta\text{–formula}\} \rangle \ .$$

Based on that the signature $\Sigma^\star$ is inductively defined by

$$\begin{aligned} \Sigma^0 &= \Sigma \\ \Sigma^{n+1} &= (\Sigma^n)_{sk} \qquad\qquad (n \geq 0) \\ \Sigma^\star &= \bigcup_{n \geq 0} \Sigma^n \ . \end{aligned}$$

The rank of a new function symbol $f_{[\delta]}$ is the least n such that $f_{[\delta]} \in \Sigma^n$.

Example 1. The formulas $(\exists x)p(x,y,z)$ and $(\exists y)p(y,x,u)$ are assigned the same Skolem–function symbol

$$f_{[(\exists x)p(x,y,z)]} = f_{[(\exists y)p(y,x,u)]} \ ,$$

but $(\exists x)p(x,x,x)$ is assigned a different one.

Example 2. Supposed $p \in P_\Sigma$ is a binary predicate symbol. Then

$$\delta = (\exists x)(\exists y)p(x,y) \in \mathcal{L}^\Sigma$$

and $f_{[\delta]} \in \Sigma_{sk} = \Sigma^1$; $f_{[\delta]}$ is of rank 1 (apart from $f_{[\delta]}$ there are more new Skolem–function symbols in Σ_{sk} — in fact, there is an infinite number of them).

This new Skolem–function symbol can now be used to build formulas over Σ_{sk}. Therefore,

$$\delta' = (\exists y)p(f_{[\delta]}, y) \in \mathcal{L}^{\Sigma^1}$$

and $f_{[\delta']} \in (\Sigma_{sk})_{sk} = \Sigma^2$ ($f_{[\delta']}$ is of rank 2).

Finally, we have

$$\delta'' = p(f_{[\delta]}, f_{[\delta']}) \in \mathcal{L}^{\Sigma^2} \ .$$

This shows that the Skolemization of δ can be done using the signature $\Sigma^\star$, since $F_{\Sigma^\star} \supset F_{\Sigma^2} \supset F_{\Sigma^1}$.

Table 5. The even more liberalized free tableau rules for quantified formulas.

$\dfrac{\gamma}{\gamma_0(x)}$	$\dfrac{\delta}{\delta_0(f_{[\delta]}(x_1,\ldots,x_n))}$
where x is a free variable.	where $x_1,\ldots,x_n$ are the free variables occurring in δ and $f_{[\delta]}$ is the function symbol assigned to δ (see Def. 2).

The same construction as above could be applied to the more comprehensive equivalence classes $[(\exists x)\phi(x,\bar{y})]$ consisting of all renamings of formulas $(\exists x)\psi(x,\bar{y})$ such that $(\forall \bar{y})(\forall x)(\phi(x,\bar{y}) \leftrightarrow \psi(x,\bar{y}))$ is a tautology. But we could find no sensible application of this.

4 Advantages of Using the δ^{++}–Rule

The example in Section 2, that has been taken from [6], shows the advantage of using the δ^{+}–rule.

The following examples illustrate that using the δ^{++}–rule instead of the δ^{+}–rule can lead to even simpler and shorter tableau proofs:

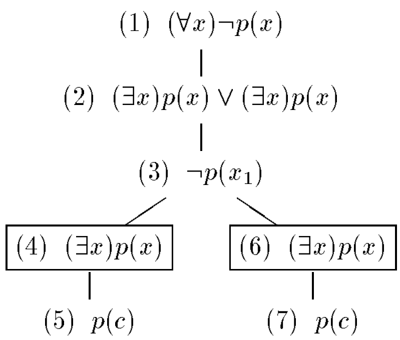
Example 3. If several instances of the formula $(\forall x)(\exists y)p(x,y)$ are generated using the γ– and the δ^{++}–rule, they are of the form[4]

$$p(x_1, f(x_1)),\ p(x_2, f(x_2)),\ \ldots$$

whereas using the δ^{+}–rule leads to instances

$$p(x_1, f_1(x_1)),\ p(x_2, f_2(x_2)),\ \ldots\ .$$

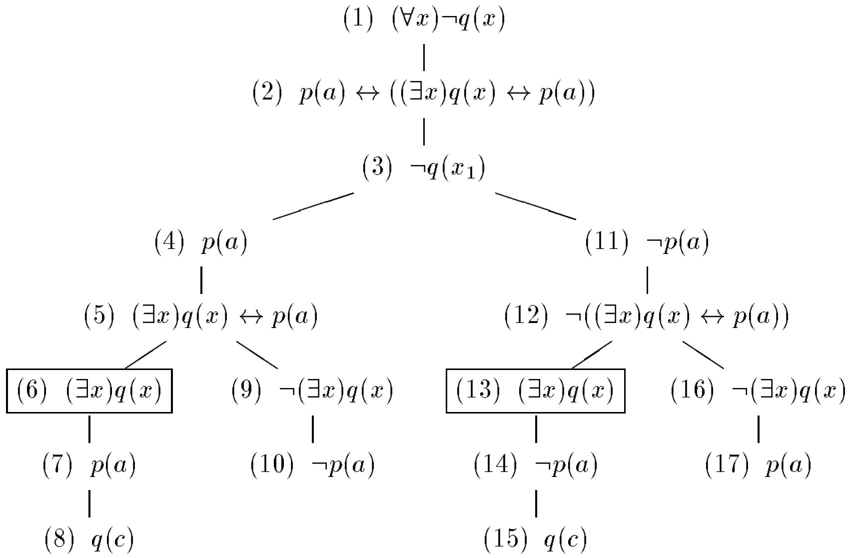
Example 4. The closed tableau (with the substitution $\{x_1 \leftarrow c\}$)



[4] Here and in the following examples we use f and c as abbreviations for $f_{[\delta]}$ (where δ is a more or less complex formula).

has been built using the δ^{++}–rule. If it had been built using the δ^{+}–rule, the formulas (5) and (7) would be of the form $p(c_1)$ and $p(c_2)$ respectively, and the tableau could only be closed by applying the γ–rule a second time to (1).

Example 5. The previous example appears to be somewhat artificial. However, identical δ–formulas quite often occur multiply if equivalences are present:



This tableau is closed with the substitution $\{x_1 \leftarrow c\}$, but, as in Example 4, if the tableau had been built using the δ^{+}–rule, the formulas (8) and (15) would be of the form $p(c_1)$ and $p(c_2)$ respectively, and the tableau could only be closed by applying the γ–rule a second time to (1).

As a matter of fact, using the δ^{++}–rule instead of the δ^{+}–rule can reduce the length of tableau proofs *exponentially*. Therefore, free variable tableaux with the δ^{+}–rule cannot polynomially simulate free variable tableaux with the δ^{++}–rule. On the other hand, using the δ^{++}–rule instead of the δ^{+}–rule, never lengthens the proof of a formula ϕ, because every closed tableau that has been built using the δ^{+}–rule, would be closed as well if it had been built using the δ^{++}–rule.

Theorem 3. *There is a class of formulas ϕ_n $(n \geq 1)$ such that, if $b^{++}(n)$ is the number of branches of the shortest closed tableau for ϕ_n using the δ^{++}–rule, then the shortest closed tableau for ϕ_n using the δ^{+}–rule has*

$$b^{+}(n) = O(2^{b^{++}(n)})$$

branches.

Proof. This can, for example, be proved using the class of formulas ϕ_n $(n \geq 1)$ defined recursively by

$$\phi_1 = \bot$$
$$\phi_n = (\forall x)(\phi_{n-1} \vee (p_n(x) \wedge ((\exists y)(\neg p_n(y)) \vee (\exists y)(\neg p_n(y))))$$

The shortest closed tableau using the δ^{++}–rule has

$$b^{++}(n) = b^{++}(n-1) + 2 = 2n - 1$$

branches, whereas the shortest closed tableau for ϕ_n using the δ^{+}–rule has

$$b^{+}(n) = 2b^{+}(n-1) + 1 = 2^n - 1$$

branches. □

In contrast to the other δ–rules, the δ^{++}–rule does not take into account the whole tableau or a whole branch, but only the local δ–formula. One does not have to keep track of the Skolem–function symbols already used.

In addition, when using the δ^{++}–rule, it suffices to have only a finite number of function symbols at hand. The number of different function symbols that have to be used while building a tableau for a formula ϕ is not larger than the number of subformulas of ϕ. That and the locality of the δ^{++}–rule are both important advantages if one wants to implement the tableau calculus (we have built the δ^{++}–rule into our tableau based theorem prover $_3T^AP$ [5]).

5 A Soundness Proof

The following proof for the soundness[5] of the δ^{++}–rule is similar to that of the δ^{+}–rule given in [6]. First, satisfiability of tableaux is defined (Def. 5), and then it is proved that satisfiability is preserved when a tableau is expanded (Lemma 6) or a substitution is applied to a tableau (Lemma 7).[6]

We consider a tableaux T, that may contain free variables. Usually we think of the free variables as being introduced by the γ–rule. First we give some preliminary definitions:

Definition 4 Structure, Model, Variable Assignment. A structure $\mathcal{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for a signature Σ consists of a domain $\mathbf{D}$ and an interpretation $\mathbf{I}$ which gives meaning to the function and predicate symbols of Σ.

A variable assignment is a mapping $\mu : \mathbf{Var} \rightarrow \mathbf{D}$ from the set of variables to the domain $\mathbf{D}$.

[5] Since the δ^{++}–rule is a liberalization of the δ–rule, completeness is obviously preserved and does not have to be proved.

[6] The satisfiability of semantic tableaux as defined in Definition 5 differs from that used in [3], which is *not* preserved if the δ^{+}– or the δ^{++}–rule is applied. This is the main reason why the soundness proof for the free δ–rule in [3], as it stands, does not carry over to the tableau system using the δ^{+}– or the δ^{++}–rule.

The combination of an interpretation $\mathbf{I}$ and an assignment μ associates (by structural recursion) with each term t over Σ an element $t^{\mathbf{I},\mu}$ in $\mathbf{D}$ and with each formula $\phi \in \mathcal{L}^{\Sigma}$ a truth value

$$val_{\mathbf{I},\mu}(\phi) \in \{\mathbf{true}, \mathbf{false}\} \ .$$

If $val_{\mathbf{I},\mu}(\phi) = \mathbf{true}$, we call $\mathcal{M}$ a model of the formula ϕ for the assignment μ (denoted by $(\mathcal{M}, \mu) \models \phi$). If $(\mathcal{M}, \mu) \models \phi$ holds for all assignments μ, we use the abbreviation $\mathcal{M} \models \phi$.

Definition 5 Satisfiability of Tableaux. A tableau T is satisfiable if there is a structure $\mathcal{M}$, such that for every variable assignment μ we have $(\mathcal{M}, \mu) \models T$, i.e., there is a branch B in T, such that $(\mathcal{M}, \mu) \models B$.

If we want to be more specific we say that T is satisfied by $\mathcal{M}$ and write in symbols $\mathcal{M} \models T$.

A branch B is considered here as a set of formulas and $(\mathcal{M}, \mu) \models B$ means $(\mathcal{M}, \mu) \models \phi$ for all formulas ϕ in B.

Lemma 6. *If T is a tableau whose root is labeled by a satisfiable closed formula ϕ, then T is satisfiable.*

Proof. Let $\mathcal{M}^0 = \langle \mathbf{D}, \mathbf{I}^0 \rangle$ be a structure for the signature $\Sigma = \Sigma^0$ that satisfies $\phi \in \mathcal{L}^{\Sigma}$.

We inductively define a sequence $(\mathcal{M}^n)_{n \geq 0}$ of structures that all have the domain $\mathbf{D}$. $\mathcal{M}^{n+1} = \langle \mathbf{D}, \mathbf{I}^{n+1} \rangle$ is a structure for the signature Σ^{n+1}; $\mathbf{I}^{n+1}$ coincides with $\mathbf{I}^n$ on all symbols in Σ^n. The function symbols $f_{[\delta]}$ of rank $r \leq n$ have already been interpreted in $\mathcal{M}^n$. Consider $f_{[\delta]}$ of rank $n+1$ with $\delta = (\exists x)\delta_0(x, \bar{y})$; its interpretation $f_{[\delta]}^{n+1}$ is for all argument tupels $\bar{b} \subset \mathbf{D}$ of the appropriate length defined by:

1. If there is a variable assignment μ with $\mu(\bar{y}) = \bar{b}$ and $(\mathcal{M}^n, \mu) \models \delta$ we choose an element $c \in \mathbf{D}$ with[7]

   $$(\mathcal{M}^n, \mu[x \leftarrow c]) \models \delta_0(x, \bar{y})$$

   and set

   $$f_{[\delta]}^{n+1}(\bar{b}) = c \ .$$

2. Otherwise we set

   $$f_{[\delta]}^{n+1}(\bar{b}) = c$$

   for an arbitrary element $c \in \mathbf{D}$.

[7] Since $f_{[\delta]}$ is of rank $n+1$, the symbols in δ are from the signature Σ^n.

We can think of the sequence $(\mathcal{M}^n)_{n\geq 0}$ as an approximization to a structure $\mathcal{M}^\star = \langle \mathbf{D}, \mathbf{I}^\star \rangle$ for the signature $\Sigma^\star$. $\mathbf{I}^\star$ coincides with $\mathbf{I}^n, \mathbf{I}^{n+1}, \mathbf{I}^{n+2}, \ldots$ on the symbols in Σ^n.

Since T is a tableau whose root is labeled by ϕ, there has to be a sequence

$$\phi = T^0, \ldots, T^m = T$$

of tableaux, where T^{i+1} is constructed from T^i by applying a single tableau rule. By induction on m we will prove that $\mathcal{M}^\star$ satisfies all the tableaux $T^0, \ldots, T^m$ (and in particular T).

$m = 0$: $\mathcal{M}^0 \models \phi$, ϕ is a formula in signature Σ, and $\mathcal{M}^0$ and $\mathcal{M}^\star$ coincide on all symbols in Σ. Therefore $\mathcal{M}^\star$ satisfies $T^0 = \phi$.

$m \to m+1$: Let B^m be a branch in T^m. T^{m+1} is obtained from T^m by applying a tableau rule to a formula on B^m.

By assumption $\mathcal{M}^\star$ satisfies T^m. Let μ be a fixed assignment. Thus we have $(\mathcal{M}^\star, \mu) \models B_0^m$ for some branch B_0^m of T^m. If B_0^m is different from B^m, then B_0^m is also a branch of T^{m+1} and we are through.

If on the other hand $B_0^m = B^m$ and therefore $(\mathcal{M}^\star, \mu) \models B^m$, we show that $(\mathcal{M}^\star, \mu)$ satisfies one of the branches of T^{m+1} by cases according to which tableau rule is applied to obtain T^{m+1} from T^m.

β–rule: Let β be a β–formula in B^m. T^{m+1} is obtained from T^m by adding β_1 to B^m obtaining B_1^{m+1} and adding β_2 to B^m obtaining B_2^{m+1}.

We have $(\mathcal{M}^\star, \mu) \models \beta$. By the property of β–formulas this entails that $(\mathcal{M}^\star, \mu) \models \beta_1$ or $(\mathcal{M}^\star, \mu) \models \beta_2$. Therefore $(\mathcal{M}^\star, \mu) \models B_1^{m+1}$ or $(\mathcal{M}^\star, \mu) \models B_2^{m+1}$.

α–rule: Similar to the β–rule and left to the reader.

γ–rule: Let γ be a γ–formula in B^m and T^{m+1} be obtained from T^m by adding $\gamma_0(x)$ to B^m obtaining the branch B^{m+1}.

We have $(\mathcal{M}^\star, \mu) \models \gamma$. By definition of $\models$ this gives $(\mathcal{M}^\star, \mu[x \leftarrow d]) \models \gamma_0(x)$ for all elements $d \in \mathbf{D}$. Since this is in particular true for $d = \mu(x)$ we get $(\mathcal{M}^\star, \mu) \models \gamma_0(x)$ and therefore also $(\mathcal{M}^\star, \mu) \models B^{m+1}$.

δ^{++}–rule: Let $\delta = (\exists x)\delta_0(x, \bar{y})$ be a δ–formula in B^m and T^{m+1} be obtained from T^m by adding $\delta_0(f_{[\delta]}(\bar{y}), \bar{y})$ to B^m obtaining the branch B^{m+1}. Let r be the rank of $f_{[\delta]}$.

We have $(\mathcal{M}^\star, \mu) \models (\exists x)\delta_0(x, \bar{y})$. By definition of the structures $\mathcal{M}^0, \mathcal{M}^1, \ldots$ and $\mathcal{M}^\star$ that implies $(\mathcal{M}^{r-1}, \mu) \models (\exists x)\delta_0(x, \bar{y})$; and the function value

$$c = f^\star_{[\delta]}(\mu(\bar{y})) = f^r_{[\delta]}(\mu(\bar{y}))$$

was chosen such that $(\mathcal{M}^r, \mu[x \leftarrow c]) \models \delta_0(x, \bar{y})$ and therefore $(\mathcal{M}^\star, \mu[x \leftarrow c]) \models \delta_0(x, \bar{y})$. Thus $(\mathcal{M}^\star, \mu) \models \delta_0(f_{[\delta]}(\bar{y}), \bar{y})$ and so $(\mathcal{M}^\star, \mu) \models B^{m+1}$ follow. □

Because the construction of $\mathcal{M}^\star$ in the proof of Lemma 6 does not depend on the tableau T, we not only have shown that a tableau for a satisfiable formula ϕ is satisfiable, but that there is a single structure $\mathcal{M}^\star$ satisfying all tableaux for ϕ.

Lemma 7. *Let T be a satisfiable tableau and τ a substitution, that associates with every free variable in T a term in the language of T, then $T\tau$ is also satisfiable.*

Proof. By hypothesis there is a structure $\mathcal{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ such that for all variable assignments μ we have $(\mathcal{M}, \mu) \models T$. We claim that for the same structure $\mathcal{M}$, we have also for all assignments ξ that $(\mathcal{M}, \xi) \models T\tau$.

To prove the above claim we consider a given variable assignment ξ. Let the variable assignment μ be defined by

$$\mu(x) = (x\tau)^{\mathbf{I},\xi} \quad \text{for all } x \in \mathbf{Var} \ .$$

That implies for all terms $t \in \mathbf{Term}$, and in particular for all terms t in the tableau T,

$$(t\tau)^{\mathbf{I},\xi} = t^{\mathbf{I},\mu}$$

and therefore

$$(T\tau)^{\mathbf{I},\xi} = T^{\mathbf{I},\mu} \ ,$$

and, since $(\mathcal{M}, \mu) \models T$, as well $(\mathcal{M}, \xi) \models T\tau$. □

Definition 8 Closed Tableau. A tableau T is closed, if there is a substitution τ such that every branch of $T\tau$ contains a complementary pair of formulas.

Theorem 9 Soundness of Semantic Tableau with the δ^{++}–rule. *If T is a closed tableau whose root is labeled by the closed formula $\neg\phi$, then ϕ is universally valid.*

Proof. If T is closed, then there is a substitution τ, such that $T\tau$ is not satisfiable. By Lemmata 6 and 7 the root cannot be satisfiable; its negation is therefore universally valid. □

6 Conclusion

We have presented a Skolemization method that is very suitable for implementation, because it suffices to have only a finite number of Skolem–function symbols at hand; in addition only the formula being Skolemized (not the whole branch) has to be taken into concern. Using the δ^{++}–rule never lengthens tableau proofs. They can, however, be shortened exponentially. Proofs of typical benchmark problems are only slightly shortened, because they usually do not contain formulas multiply — in contrary to many natural problems.

There are other Skolemization methods [1] that can be used to Skolemize arbitrary formulas (that do not have to be in prenex normal form). These methods can be used to Skolemize formulas in a pre–processing step before they are handed over to the tableau prover.

Andrew's Skolemization method, for example, provides the same (or better) results than using the δ^+–rule, but, still, there are examples where using the δ^{++}–rule shortens the proofs exponentially.

It is, however, possible to combine the δ^{++}–rule and methods that do not use the prenex normal form, and thus combine the advantages of both methods.

Acknowledgements

Many thanks to Stefan Gerberding for pointing out some errors in an earlier version of this paper and to Bertram Ludäscher for fruitful discussions.

References

1. Peter B. Andrews. *An Introduction to Mathematical Logic and Type Theory.* Academic Press, 1986.
2. Melvin C. Fitting. First-order modal tableaux. *Journal of Automated Reasoning*, 4:191 – 213, 1988.
3. Melvin C. Fitting. *First–Order Logic and Automated Theorem Proving.* Springer, New York, 1990.
4. Reiner Hähnle. *Automated Proof Search in Multiple–Valued Logics.* Oxford University Press, forthcoming, 1993.
5. Reiner Hähnle, Bernhard Beckert, Stefan Gerberding, and Werner Kernig. The Many–Valued Tableau–Based Theorem Prover $_3T^AP$. IWBS Report 227, Wissenschaftliches Zentrum Heidelberg, IWBS, IBM Deutschland, July 1992.
6. Reiner Hähnle and Peter H. Schmitt. The liberalized δ–rule in free variable semantic tableaux. *Journal of Automated Reasoning, to appear*, 1993.
7. Steve V. Reeves. Semantic tableaux as a framework for automated theorem-proving. Department of Computer Science and Statistics, Queen Mary College, Univ. of London, 1987.
8. Raymond Smullyan. *First–Order Logic.* Springer, New York, 1968.