TaMeD : A Tableau Method for Deduction Modulo

Richard Bonichon
LIP6, Université Paris 6, 8 rue du Capitaine Scott, 75015 Paris, France
*(*richard.bonichon@lip6.fr*)*

Abstract.

Deduction modulo is a theoretical framework for reasoning modulo a congruence on propositions. Computational steps are thus removed from proofs, thus allowing a clean separatation of computational and deductive steps. A sequent calculus modulo has been defined in (Dowek et al., 2003) as well as a resolution-based proof search method, in which the congruences are handled through rewrite rules on terms and atomic propositions.

This article defines an automated proof search method for theorem proving modulo (TaMeD) based upon free-variable *tableaus* for classical logic .

Syntactic proofs for the soundness and completeness of the method are given with respect to provability in the sequent calculus modulo. The proofs follow a pattern similar to those of ENAR so that comparisons between some characteristics of the two methods can be drawn.

Finally, some applications of deduction modulo as well as hints at further or ongoing research in this field are briefly presented.

Keywords: tableau, automated theorem proving, rewriting, deduction modulo, sequent calculus modulo

Introduction

(Dowek et al., 2003) notice that automated theorem proving methods might lead to ineffective procedures if lacking some form of goalness. If one tries to prove the following example

$$(a + b) + ((c + d) + e) = a + ((b + c) + (d + e))$$

with the associativity and identity axioms using a naive strategy, it might end up running endlessly without finding the right solution .

It would obviously be better to be able to apply a deterministic and terminating strategy to check that the two terms are indeed the same modulo associativity. This problem would actually be more efficiently solved by *computation* (i.e blind execution) instead of *deduction* (non-deterministic search), thus replacing the

associativity axiom by a rewrite rule.
Advanced tableau-based methods allow to add equational theories to the classical tableau method described in (Smullyan, 1968). Methods for equality handling in tableaus can be found for example in (Fitting, 1996), (Beckert, 1991), (Beckert, 1994), (Beckert and Hähnle, 1992) or (Degtyarev and Voronkov, 2001) and it permits to get complete methods combining deduction and term-rewriting steps. However, rewriting on *propositions* is usually not considered in the tableau methods described in these papers though allowing rules rewriting propositions can be of practical use.
For example, the following axiom is part of the theory of integral domains

$$\forall x\ \forall y\ (x * y = 0 \Leftrightarrow (x = 0 \vee y = 0))$$

This yields the corresponding rewrite rule:

$$x * y = 0 \rightarrow x = 0 \vee y = 0$$

In this rule, an atomic proposition is turned into a disjunction and it is hard to see how it could be replaced by a rule rewriting terms. Having the rewriting rule above, we can prove the proposition:

$$\exists z\ (a * a = z \Rightarrow a = z)$$

but the closed tableau can not be derived from its negated disjunctive normal form

$$a * a = z \qquad a = z$$

since the traditional branch closure rule does not see that z can be instantiated by 0.

Therefore a rule called **Extended Narrowing** must be added. This rule suggests in this case the instantiation of z by 0 and the tableau can be closed. The use of rewriting on propositions is however restricted: the left parts of these rewriting rules must be atomic propositions, avoiding thereby potential conflicts between sequent inference steps and rewriting rules. Simply consider the rewriting rule $P \wedge Q \rightarrow R$ which left-hand side can be proved in the sequent calculus: this possibility is lost if it is rewritten to R.
The sequent calculus modulo is given in section 1 along with its principal definitions and properties. Then, the automated proof search method TaMeD is introduced in section 2 together with

some notations, before stating the main theorem we intend to prove in section 3 and the plan of the proof. Section 4 shows the soundness and completeness of an intermediate method called IC-TaMeD detailing the possible interactions between rewriting and sequent rules. After the soundness and completeness of the TaMeD method can be lifted from IC-TaMeD in section 5. A brief comparison with the ENAR method ((Dowek et al., 2003)) as well as hints at further research are finally given in the conclusion of section 6).

1. The sequent calculus modulo

1.1. Definitions

The notions of *terms, atomic propositions, propositions, sentences* are defined as usual in first-order logic, as they can for example be found in (Fitting, 1996) or (Goubault-Larrecq and Mackie, 1997). The standard substitution avoiding capture of the term t for the variable x in a proposition P is written $P[x := t]$. Moreover, some definitions of (Dowek et al., 2003) are recalled.

DEFINITION 1. (Rewriting-related vocabulary). *A* term rewrite rule *is a pair of terms $l \rightarrow r$, where the free variables of r must occur in l.*
An equational axiom *is a pair of terms $l = r$.*
A proposition rewrite rule *is a pair of propositions $l \rightarrow r$, where l is an atomic proposition, and the free variables of r must occur in l.*

DEFINITION 2. (Class rewrite system). *A class rewrite system is a pair denoted $\mathcal{RE}$ consisting of $\mathcal{R}$ — a set of proposition rewrite rules — and $\mathcal{E}$ — a set of term rewrite rules and equational axioms.*

Let us define the rewriting relations used in the paper.

DEFINITION 3. ($\xrightarrow{\mathcal{R}}$ and $\xrightarrow{\mathcal{RE}}$). *Let $\mathcal{R}$ be a proposition rewrite system, the proposition P* $\mathcal{R}$-rewrites *to P', denoted $P \xrightarrow{\mathcal{R}} P'$, if $P_{|\omega} = \sigma(l)$ and $P' = P[\sigma(r)]_\omega$, for some rule $l \rightarrow r \in \mathcal{R}$, some*

occurrence ω *in* P *and some substitution* σ*. As usual when applying* σ*, quantified variables of* r *are renamed to avoid captures.*
Let $\mathcal{RE}$ *be a class rewrite system, the proposition* P $\mathcal{RE}$-rewrites *to* P'*, denoted* $P \stackrel{\mathcal{RE}}{\rightarrow} P'$*, if* $P =_{\mathcal{E}} Q$*,* $Q_{|\omega} = \sigma(l)$ *and* $P' =_{\mathcal{E}} Q[\sigma(r)]_{\omega}$*, for some rule* $l \rightarrow r \in \mathcal{R}$*, some proposition* Q*, some occurrence* ω *in* Q *and some substitution* σ*.*

1.2. The sequent calculus modulo

The sequent calculus modulo (see figure 1) is an extension of the sequent calculus defined for first-order classical logic and if the congruence $=_{\mathcal{RE}}$ is taken to be the identity, this sequent calculus becomes the usual one. One might notice that the axiom rule requires not just unifiability of the left proposition and the right one but they have to be identical modulo the congruence. *Thus free variables cannot be instantiated and are treated as constants.*

The next two propositions are direct results of the definitions:

PROPOSITION 1. *If* $=_{\mathcal{RE}}$ *is a decidable congruence, then proof checking for the sequent calculus modulo is decidable. This is in particular the case when the rewrite relation* $\longrightarrow_{\mathcal{RE}}$ *is confluent and (weakly) terminating.*

PROPOSITION 2. *If* $P =_{\mathcal{RE}} Q$ *then* $\Gamma \vdash_{\mathcal{RE}} P, \Delta$ *if and only if* $\Gamma \vdash_{\mathcal{RE}} Q, \Delta$ *and* $\Gamma, P \vdash_{\mathcal{RE}} \Delta$ *if and only if* $\Gamma, Q \vdash_{\mathcal{RE}} \Delta$ *and the proofs have the same size.*

PROPOSITION 3. *If a closed sequent* $\Gamma \vdash_{\mathcal{RE}} \Delta$ *has a proof, then it also has a proof where all the sequents are closed.*

1.3. Equivalence between $\vdash$ and $\vdash_{\mathcal{RE}}$

This subsection states a really important property regarding the equivalence between the classical sequent calculus and the sequent calculus modulo, as proved in (Dowek et al., 2003). Indeed, it states the soundness and completeness of the sequent calculus modulo with respect to first-order logic.

DEFINITION 4. (Compatibility). *A set of axioms* $\mathcal{K}$ *and a class rewrite system* $\mathcal{RE}$ *are said to be* compatible *if:*

$$\frac{}{P \vdash_{\mathcal{RE}} Q}\text{axiom} \quad \text{if } P =_{\mathcal{RE}} Q$$

$$\frac{\Gamma, P \vdash_{\mathcal{RE}} \Delta \quad \Gamma \vdash_{\mathcal{RE}} Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} \Delta}\text{cut} \quad \text{if } P =_{\mathcal{RE}} Q$$

$$\frac{\Gamma, Q_1, Q_2 \vdash_{\mathcal{RE}} \Delta}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}\text{contr-l} \quad \text{if } P =_{\mathcal{RE}} Q_1 =_{\mathcal{RE}} Q_2$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} Q_1, Q_2, \Delta}{\Gamma \vdash_{\mathcal{RE}} P, \Delta}\text{contr-r} \quad \text{if } P =_{\mathcal{RE}} Q_1 =_{\mathcal{RE}} Q_2$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} \Delta}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}\text{weak-l}$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} \Delta}{\Gamma \vdash_{\mathcal{RE}} P, \Delta}\text{weak-r}$$

$$\frac{\Gamma, P, Q \vdash_{\mathcal{RE}} \Delta}{\Gamma, R \vdash_{\mathcal{RE}} \Delta}\wedge\text{-l} \quad \text{if } R =_{\mathcal{RE}} (P \wedge Q)$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} P, \Delta \quad \Gamma \vdash_{\mathcal{RE}} Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} R, \Delta}\wedge\text{-r} \quad \text{if } R =_{\mathcal{RE}} (P \wedge Q)$$

$$\frac{\Gamma, P \vdash_{\mathcal{RE}} \Delta \quad \Gamma, Q \vdash_{\mathcal{RE}} \Delta}{\Gamma, R \vdash_{\mathcal{RE}} \Delta}\vee\text{-l} \quad \text{if } R =_{\mathcal{RE}} (P \vee Q)$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} P, Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} R, \Delta}\vee\text{-r} \quad \text{if } R =_{\mathcal{RE}} (P \vee Q)$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} P, \Delta \quad \Gamma, Q \vdash_{\mathcal{RE}} \Delta}{\Gamma, R \vdash_{\mathcal{RE}} \Delta}\Rightarrow\text{-l} \quad \text{if } R =_{\mathcal{RE}} (P \Rightarrow Q)$$

$$\frac{\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} R, \Delta}\Rightarrow\text{-r} \quad \text{if } R =_{\mathcal{RE}} (P \Rightarrow Q)$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} P, \Delta}{\Gamma, R \vdash_{\mathcal{RE}} \Delta}\neg\text{-l} \quad \text{if } R =_{\mathcal{RE}} \neg P$$

$$\frac{\Gamma, P \vdash_{\mathcal{RE}} \Delta}{\Gamma \vdash_{\mathcal{RE}} R, \Delta}\neg\text{-r} \quad \text{if } R =_{\mathcal{RE}} \neg P$$

$$\frac{}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}\bot\text{-l} \quad \text{if } P =_{\mathcal{RE}} \bot$$

$$\frac{\Gamma, \{t/x\}Q \vdash_{\mathcal{RE}} \Delta}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}(Q, x, t)\ \forall\text{-l} \quad \text{if } P =_{\mathcal{RE}} \forall x Q$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} \{c/x\}Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} P, \Delta}(Q, x, c)\ \forall\text{-r} \quad \text{if } P =_{\mathcal{RE}} \forall x Q \text{ and } c \text{ fresh constant}$$

$$\frac{\Gamma, \{c/x\}Q \vdash_{\mathcal{RE}} \Delta}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}(Q, x, c)\ \exists\text{-l} \quad \text{if } P =_{\mathcal{RE}} \exists x Q \text{ and } c \text{ fresh constant}$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} \{t/x\}Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} P, \Delta}(Q, x, t)\ \exists\text{-r} \quad \text{if } P =_{\mathcal{RE}} \exists x\ Q$$

Figure 1. The sequent calculus modulo

$-P =_{\mathcal{RE}} Q$ *implies* $\mathcal{K} \vdash P \iff Q$.

–for every proposition $P \in \mathcal{K}$*, we have* $\vdash_{\mathcal{RE}} P$.

PROPOSITION 4. *For every class rewrite system* $\mathcal{RE}$*, there is a set of axioms* $\mathcal{K}$ *such that* $\mathcal{K}$ *and* $\mathcal{RE}$ *are compatible.*

PROPOSITION 5. (Equivalence). *If the set of axioms* $\mathcal{K}$ *and the class rewrite system* $\mathcal{RE}$ *are compatible then we have:*

$$\mathcal{K}, \Gamma \vdash \Delta \text{ if and only if } \Gamma \vdash_{\mathcal{RE}} \Delta$$

Proof: See (Dowek et al., 2003). $\diamondsuit$

The latter proposition entails that the two formalisms can be used to deduce the same theorems. Of course, a proof of the same theorem may be of different size depending on the formalism used. Actually, proof are generally smaller in the sequent calculus modulo.

2. The TaMeD method

This section extends the classical tableau method for first-order classical logic as defined in (Smullyan, 1968) and (Fitting, 1996) to a tableau method where congruences are built-in. *In the rest of this paper, we assume the relation* $\longrightarrow^*_{\mathcal{RE}}$ *to be confluent.*

2.1. Labels

The usual first step of a tableau based proof search method is to transform the proposition to be proved (or rather refuted) into a set of branches that involves skolemization. In fact, several skolemized forms are possible for a proposition. Take for example the closed formula $\forall x\ \exists y\ P(0, y)$ where the variable x does not occur: the Skolem constant f could be as well be nullary as unary, yielding respectively $P(0, f)$ or $P(0, f(x))$. The latter is chosen in this paper because of the following fact: if we had an equation $x * 0 = 0$ leading to an $\mathcal{E}$-equivalence between $\forall x\ \exists y\ P(0, y)$ and $\forall x\ \exists y\ P(x * 0, y)$, the Skolem symbols would have the same arity in both cases. This choice is implemented by memorizing the universal quantifier scope of each subformula during the tableau form computation by associating a *label*.

DEFINITION 5. (Labeled proposition). *A* labeled proposition *is a pair P^l formed by a proposition P and a finite set l of variables containing all the free variables of P called its* label.

DEFINITION 6. (Substitution in a labeled proposition). *When we apply a substitution Θ to a labeled proposition, each variable x of the label is replaced by the free variables of Θx. Two labeled propositions P^l and $Q^{l'}$ are $\mathcal{E}$-equivalent if $P =_{\mathcal{RE}} Q$ and $l = l'$. The labeled proposition P^l $\mathcal{R}$-rewrites to $Q^{l'}$ if P $\mathcal{R}$-rewrites to Q (definition 3) and $l = l'$.*

2.2. From formulas to tableaus

The notations used throughout the paper to represent transformations of a tableau are inspired by the ones that can be found either in (Dowek et al., 2003) for clausal form transformations and in (Degtyarev and Voronkov, 1998; Degtyarev and Voronkov, 2001) for tableaus. The more classical tree-like presentation of (Smullyan, 1968; Fitting, 1996) is therefore not use, mostly does not give at first glance a global view of the tableau transformation in the expansion rules. The names $\alpha-, \beta-, \gamma-, \delta-$formulas are those commonly used in tableau-related literature.

2.2.1. *Definitions and notations*

Let us introduce some basic definitions regarding tableaus.

- A *branch* is a multiset $\{Q_1, \ldots, Q_n\}$ of formulas.

- A *fully expanded branch* (fxp-branch) of tableau is a multiset $\{P_1, \ldots, P_n\}$ of formulas such that every P_i is a literal, i.e. either an atomic proposition or the negation of an atomic proposition.

- A *tableau* is a multiset $\{\mathcal{B}_1 \mid \ldots \mid \mathcal{B}_p\}$ of branches.

- A branch is said *closed* if a contradiction can be derived from the formulas composing the branch. In the case of propositionnal classical logic, it means that P and $\neg P$ are on the same branch.

- A tableau is then said *closed* if every branch of the tableau is closed.
- The closed tableau is denoted $\odot$ in this paper.

The notations used throughout this paper are briefly sumed up thereafter. Local changes will be explicitly stated as they arise.

- Terms are denoted $l, r, s, t, \ldots$;
- Propositions are denoted $L, P, Q, R, \ldots$;
- Branches are denoted by $\mathcal{B}, \Gamma, \Phi, \Psi, \ldots$ and $\mathcal{B}, P$ is a notation for the set $\mathcal{B} \cup \{P\}$;
- Tableaus are denoted $\mathcal{T}, \mathcal{U}, \mathcal{V}, \ldots$ and $\mathcal{T} \mid \mathcal{B}$ is a notation for $\mathcal{T} \cup \mathcal{B}$;
- fxp-branches are denoted by the same symbols as sets of propositions and sets of fxp-branches are denoted as sets of sets of propositions;
- When $\mathcal{B} = \{P_1, \ldots, P_n\}$ is a set of propositions then $\mathrm{tnf}(\mathcal{B})$ or $\mathrm{tnf}(P_1, \ldots, P_n)$ is taken to be the same as $\mathrm{tnf}(P_1 \wedge \ldots \wedge P_n)$.
- Labeled propositions are denoted P^l where l is the label of the proposition P. Notations for branches and tableaus are naturally extended with labeled propositions.

2.2.2. *Tableau expansions*

A presentation of the tableau expansion calculus is given below which is used to reduce a formula to what we will call its tableau normal form.

DEFINITION 7. (Tableau normal form: tnf). *To put a set of non-sentences in tableau form, we first label them with an empty set. The universally quantified propositions are also labeled with a given integer n_x denoting the allowed number of γ-expansions per γ-formula (otherwise the computation may be infinite), where x is the universally quantified variable bound in the γ-formula. We consider the following transformations on multisets of multisets of labeled propositions.*

$\beta \quad \mathcal{T} \mid (\mathcal{B}, (P \vee Q)^l)) \xrightarrow{tnf} \mathcal{T} \mid \mathcal{B}, P^l \mid \mathcal{B}, Q^l$

$\beta \quad \mathcal{T} \mid (\mathcal{B}, \neg(P \wedge Q)^l)) \xrightarrow{tnf} \mathcal{T} \mid \mathcal{B}, \neg P^l \mid \mathcal{B}, \neg Q^l$

$\beta \quad \mathcal{T} \mid (\mathcal{B}, (P \Rightarrow Q)^l)) \xrightarrow{tnf} \mathcal{T} \mid \mathcal{B}, \neg P^l \mid \mathcal{B}, Q^l$

$\alpha \quad \mathcal{T} \mid (\mathcal{B}, (P \wedge Q)^l) \xrightarrow{tnf} \mathcal{T} \mid \mathcal{B}, P^l, Q^l$

$\alpha \quad \mathcal{T} \mid (\mathcal{B}, \neg(P \vee Q)^l) \xrightarrow{tnf} \mathcal{T} \mid \mathcal{B}, \neg P^l, \neg Q^l$

$\alpha \quad \mathcal{T} \mid (\mathcal{B}, \neg(P \Rightarrow Q)^l) \xrightarrow{tnf} \mathcal{T} \mid \mathcal{B}, P^l, \neg Q^l$

$\alpha \quad \mathcal{T} \mid (\mathcal{B}, \neg\neg P^l) \xrightarrow{tnf} \mathcal{T} \mid \mathcal{B}, P^l$

$\gamma \quad \mathcal{T} \mid (\mathcal{B}, (\forall x P)^l_{n_x}) \xrightarrow{tnf} \mathcal{T} \mid (\mathcal{B}, (\forall x P)^l_{n_x-1}, P^{l,x}$ *where* x *is a fresh variable and* $n_x > 1$

$\gamma \quad \mathcal{T} \mid (\mathcal{B}, (\forall x P)^l_1) \xrightarrow{tnf} \mathcal{T} \mid \mathcal{B}, P^{l,x}$ *where* x *is a fresh variable*

$\gamma \quad \mathcal{T} \mid (\mathcal{B}, \neg(\exists x P)^l_{n_x}) \xrightarrow{tnf} \mathcal{T} \mid \mathcal{B}, \neg(\exists x P)^l_{n_x-1}, \neg P^{l,x}$ *where* x *is a fresh variable and* $n_x > 1$

$\gamma \quad \mathcal{T} \mid (\mathcal{B}, \neg(\exists x P)^l_1) \xrightarrow{tnf} \mathcal{T} \mid \mathcal{B}, \neg P^{l,x}$ *where* x *is a fresh variable*

$\delta \quad \mathcal{T} \mid (\mathcal{B}, (\exists x P)^{y_1,\ldots,y_n}) \xrightarrow{tnf} \mathcal{T} \mid \mathcal{B}, (P\{x := f(y_1, \ldots, y_n)\})^{y_1,\ldots,y_n}$ *where* f *is a fresh Skolem symbol*

$\delta \quad \mathcal{T} \mid (\mathcal{B}, \neg(\forall x P)^{y_1,\ldots,y_n})) \xrightarrow{tnf} \mathcal{T} \mid \mathcal{B}, (\neg P[x := f(y_1, \ldots, y_n)])^{y_1,\ldots,y_n}$ *where* f *is a fresh Skolem symbol*

$-\mathcal{T} \mid (\mathcal{B}, \bot^l) \xrightarrow{tnf} \mathcal{T}$

$-\mathcal{T} \mid (\mathcal{B}, \neg \bot^l) \xrightarrow{tnf} \mathcal{T} \mid \mathcal{B}$

Remark. [On free variables, tnf-equality]

Free variables created during γ-expansions are always fresh and the notation for them in tnf is an abused shortened notation for:

$\mathcal{T} \mid (\mathcal{B}, (\forall x P)^l_1) \xrightarrow{tnf} \mathcal{T} \mid \mathcal{B}, P^{l,z}$ where z is a fresh variable and x is replaced by z in P.

tnf($\mathcal{T}$) = tnf($\mathcal{U}$) means that tnf yields the same result for both tableaus modulo an appropriate renaming of free variables. In particular it means that either $\mathcal{T} \overset{tnf}{\rightarrow^\star} \mathcal{U}$, $\mathcal{U} \overset{tnf}{\rightarrow^\star} \mathcal{T}$ or $\mathcal{T}$ and $\mathcal{U}$ are identical up to some renaming of free variables.

The tnf transformation deals with $\forall$-quantifiers by adding an integer to annotate the allowed number of γ-expansions. If tnf for a given integer does not give a fully expanded tableau that can be closed, it is possible to go on by *iterative deepening*. Allowing multiple γ-expansions is required for the completeness of the whole calculus.

2.2.3. *Properties*

This section briefly proves the termination and soudness properties of tnf. An ordering is defined on the tnf calculus of definition 7 as follows:

DEFINITION 8. (tnf-ordering). *The* tnf-ordering *is defined as a lexical order on tableaus using the following pair:*

> *−As first component, we take the multiset ordering of the n_x's associated with the representation of the universally quantified variables in the labels of the propositions of each branch.*
>
> *−As second component , we take the multiset of pairs (a, b) where a is the number of occurrences of the symbols $\wedge, \vee, \Rightarrow$ $, \bot, \forall, \exists$ and b the number of occurrences of the symbol $\neg$ for each branch in $\mathcal{T}$.*

This ordering is used to prove the termination of tnf.

PROPOSITION 6. (Termination). *The tnf transformation terminates for any given allowed number n of γ-expansion per γ-proposition. The* tableau n-normal form *is defined as the result of tnf for a given formula with a given γ-expansions allowed n. This result will also be called* tableau normal form.

Proof: Each rule decreases the complexity of the tnf-ordering defined on a tableau $\mathcal{T}$ in definition 8. $\Diamond$

Before adding any deduction modulo ability to tableaus, one needs to ensure that tnf transformations preserve satisfiability. Some more definitions are needed before proving this property.

DEFINITION 9. (Free variables of a branch). *Let* $\mathcal{B} = \{P_1, \ldots P_n\}$ *. The free variables of* $\mathcal{B}$ *are defined as the union of the sets of free variables of* $P_1, \ldots, P_n$*.*

DEFINITION 10. ($\overline{\forall}$ notation). *Let the* Γ_i *be multisets of propositions. Let* $x_1, \ldots, x_n$ *be the union of the free variables of the* Γ_i*s. We will use the following notation:*

$$\overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n) = \forall x_1, \ldots, \forall x_n(\Gamma_1 \vee \ldots \vee \Gamma_n)$$

The tnf transformations are a restricted case of the general free-variable tableau rules of (Fitting, 1996) and (Letz, 1998) and therefore the reader is referred to the soundness proof of (Letz, 1998) (pp. 165-167).

LEMMA 1. (tnf soundness). *Let* $\Gamma_1, \ldots, \Gamma_m$ *and* $\mathcal{B}_1, \ldots, \mathcal{B}_n$ *be sets of labeled propositions. If*

$$\{\Gamma_1 \mid \ldots \mid \Gamma_m\} \xrightarrow{tnf} \{\mathcal{B}_1 \mid \ldots \mid \mathcal{B}_n\}$$

Then

$$\overline{\forall}(\mathcal{B}_1 \vee \ldots \vee \mathcal{B}_n) \vdash \quad \Longrightarrow \quad \overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_m) \vdash$$

Proof: The tableau normal form algorithm is a restricted case of the general satisfiability-preserving free-variable tableau expansion rules. $\Diamond$

2.3. Rules for TaMeD

This section defines constrained tableaus and gives the rules of TaMeD .

DEFINITION 11. *Equations and substitutions For some equational theory* $\mathcal{E}$*, an equation modulo* $\mathcal{E}$ *is a pair of terms or of atomic propositions denoted* $t =^?_{\mathcal{E}} t'$*. A substitution* σ *is a* $\mathcal{E}$*-solution of* $t =^?_{\mathcal{E}} t'$ *when* $\sigma t =_{\mathcal{E}} \sigma t'$*. It is a* $\mathcal{E}$*-solution of an equation system* C *when it is a solution of all the equations in* C*.*

DEFINITION 12. (Constrained tableau).
A constrained tableau *is a pair* $\mathcal{T}[C]$ *such that* $\mathcal{T}$ *is a tableau and* C *is a set of equations called* constraints*.*

Using these definitions, TaMeD can be introduced.

DEFINITION 13. *Let* $\mathcal{RE}$ *be a class rewrite system and* $\mathcal{T}[C]$ *a constrained tableau, we write*

$$\mathcal{T}[C] \overset{T}{\mapsto} \mathcal{T}'[C']$$

if the constrained tableau $\mathcal{T}'[C']$ *can be deduced from the constrained tableau* $\mathcal{T}[C]$ *using finitely many applications of the* ***Extended Narrowing*** *and* ***Extended Branch Closure*** *rules described in figure 2. This means there is a derivation of the tableau* $\mathcal{T}'[C']$ *under the assumptions* $\mathcal{T}[C]$*, i.e. a sequence* $\mathcal{T}_1[C_1], \ldots, \mathcal{T}_n[C_n]$ *such that either* $n = 0$ *and* $\mathcal{T}'[C'] = \mathcal{T}[C]$ *or* $n > 0, \mathcal{T}_0[C_0] = \mathcal{T}[C], \mathcal{T}_n[C_n] = \mathcal{T}'[C']$ *and each* $\mathcal{T}_i[C_i]$ *is produced by the application of a rule in TaMeD to* $\mathcal{T}_{i-1}[C_{i-1}]$*.*

The first rule, **Extended Branch Closure** is a simple extension of the usual branch closure rule for first-order equational tableau, where similarly to the equational constrained tableau method in (Degtyarev and Voronkov, 2001), the $\mathcal{E}$-unification constraints are not solved but stored in the constraint part. Although propositions are labeled with variables, these play no role when applying the **Extended Branch Closure** rule. In particular, they are removed from the constraints part of the tableau. Moreover, we also have here a major difference between resolution and tableau methods: the **Extended Branch Closure** is only binary, whereas the *Extended Resolution* rule for ENAR in (Dowek et al., 2003) needs to be applied to all relevant propositions in order to be complete.

The **Extended Narrowing** rule is much the same as the one proposed for resolution and the narrowing is only applied to atomic propositions and not directly to terms. As atomic propositions may of course be rewritten to non-atomic ones, it must be ensured that they are transformed back in disjunctive normal form.

When $\mathcal{R}$ is empty the **Extended Narrowing** rule is never used and we get a method for equational tableau. When both $\mathcal{R}$ and

Extended Branch Closure

$$\frac{\Gamma_1, P, \neg Q \mid \Gamma_2 \mid \ \ldots \ \mid \Gamma_n \ [C]}{\Gamma_2 \mid \ \ldots \ \mid \Gamma_n \ [C \cup \{P \overset{?}{\underset{\mathcal{E}}{=}} Q\}]}$$

Extended Narrowing

$$\frac{\Gamma_1, U \mid \Gamma_2 \mid \ \ldots \ \mid \Gamma_n \ [C]}{\mathcal{B}_1 \mid \ \ldots \ \mid \mathcal{B}_p \mid \Gamma_2 \mid \ \ldots \ \mid \Gamma_n \ [C \cup \{U_{|\omega} \overset{?}{\underset{\mathcal{E}}{=}} l\}]}$$

if $l \rightarrow r \in \mathcal{R}$, $U_{|\omega}$ is an atomic proposition and $\mathcal{B}_1 \mid \ \ldots \ \mid \mathcal{B}_p = \mathrm{tnf}(\Gamma_1, U[r]_\omega)$

Figure 2. TaMeD rules

$\mathcal{E}$ are empty, then we get back a first-order free variable tableau method.

3. Main theorem

The main result of this paper states the soundness and completeness of the TaMeD method with respect to the sequent calculus modulo.

THEOREM 1. (Main theorem). *Let $\mathcal{RE}$ be a class rewrite system such that $\longrightarrow_{\mathcal{RE}}$ is confluent. For every $\mathcal{B}$ and Γ sets of closed formulas, if C is a $\mathcal{E}$-unifiable set of constraints, then we have the following implications:*

$$Tab(\mathcal{B} \wedge \neg\Gamma)[\emptyset] \overset{T}{\underset{\mathcal{RE}}{\mapsto}} \odot[C] \Rightarrow \mathcal{B} \vdash_{\mathcal{RE}} \Gamma$$

where $\neg\Gamma = \{\neg P | P \in \Gamma\}$.

If the sequent $\mathcal{B} \vdash_{\mathcal{RE}} \Gamma$ has a cut-free proof then there exists a derivation

$$Tab(\mathcal{B} \wedge \neg\Gamma)[\emptyset] \overset{T}{\underset{\mathcal{RE}}{\mapsto}} \odot[C]$$

The following corollary is deduced from the same hypothesis when the cut-elimination property holds

$$Tab(\mathcal{B} \wedge \neg\Gamma)[\emptyset] \overset{T}{\underset{\mathcal{RE}}{\longmapsto}} \odot[C] \quad \Leftrightarrow \quad \mathcal{B} \vdash_{\mathcal{RE}} \Gamma$$

The proof of the main theorem is detailed in the next sections. Its main steps are summed up as a scheme below and require the IC-TaMeD method given in section 4:

$$\begin{array}{c} \mathcal{K}, \mathcal{B}, \neg\Gamma \vdash \\ \overset{Lem.5}{\Longleftrightarrow} \\ \mathcal{B}, \neg\Gamma \vdash_{\mathcal{RE}} \\ \underset{Prop.8}{\overset{Prop.7}{\Longleftrightarrow}} \\ \mathcal{B}, \neg\Gamma \overset{\text{IcT}}{\hookrightarrow} \odot \\ \underset{Prop.10}{\overset{Prop.9}{\Longleftrightarrow}} \\ \mathcal{B}, \neg\Gamma\ [\emptyset] \overset{T}{\longmapsto} \odot\,[C] \end{array}$$

- The second one (propositions 7 and 8) states that the IC-TaMeD method is sound and complete with respect to provability in the sequent calculus modulo. The completeness proof requires that the cut rule is eliminable in the sequent calculus modulo $\mathcal{RE}$.

- The third and last part of the proof (propositions 9 and 10) is the lifting of the proofs of IC-TaMeD to the TaMeD method.

Finally, this series of small steps show that the tableau tnf$(\Gamma \wedge \neg\Delta)$ can be refuted if and only if the sequent $\mathcal{K}, \Gamma \vdash \Delta$ is provable in the sequent calculus.

4. Soundness and completeness of the IC-TaMeD method

In order to prove the theorem stated in section 3, we first define an intermediate calculus simply called *Intermediate Calculus for TaMeD* (IC-TaMeD). The allowed rules in this calculus are described in figure 3 .

Let us define the notion of IC-TaMeD derivation.

$$\frac{\Gamma_1 \mid \ldots \mid \Gamma_n}{(\Gamma_1 \mid \ldots \mid \Gamma_n)[x := t]} \textbf{ Instantiation}$$

$$\frac{\Gamma_1, P \mid \Gamma_2 \mid \ldots \mid \Gamma_n}{\Gamma_1, P' \mid \Gamma_2 \mid \ldots \mid \Gamma_n} \textbf{ Conversion} \text{ if } P =_{\mathcal{E}} P'$$

$$\frac{\Gamma_1, P \mid \Gamma_2 \mid \ldots \mid \Gamma_n}{\mathcal{B}_1 \mid \ldots \mid \mathcal{B}_p \mid \Gamma_2 \mid \ldots \mid \Gamma_n} \textbf{ Reduction}$$

if $P \xrightarrow{\mathcal{R}} Q$ and $\mathcal{B}_1 \mid \ldots \mid \mathcal{B}_p = tnf(\Gamma_1, Q)$

$$\frac{\Gamma 1, P^{l_1}, \neg P^{l_2} \mid \Gamma_2 \mid \ldots \mid \Gamma_n}{\Gamma_2 \mid \ldots \mid \Gamma_n} \textbf{ Identical Branch Closure}$$

Figure 3. IC-TaMeD

DEFINITION 14. (IC-TaMeD derivation). *Let $\mathcal{RE}$ be a class rewrite system and $\mathcal{T}$ a tableau, we note:*

$$\mathcal{T} \overset{IcT}{\hookrightarrow} \mathcal{T}'$$

if the tableau $\mathcal{T}'$ can be obtained from $\mathcal{T}$ using finitely many applications of the IC-TaMeD rules described in Fig. 3. This means there is a sequence $\mathcal{T}_1, \ldots, \mathcal{T}_n$ such that either $n = 0$ and $\mathcal{T} = \mathcal{T}'$ or $n > 0, \mathcal{T} = \mathcal{T}_0, \mathcal{T}' = \mathcal{T}_n$ and each $\mathcal{T}_i$ is produced by the application of a rule of IC-TaMeD to the tableau $\mathcal{T}_{i-1}$.

Some remarks about the rules of figure 3 are necessary:

– In the **Instantiation** rule, the instantiated variable is replaced in the label by the free variables of the substituted term.

– In the **Conversion** rule, ,labels are kept by the transformed propositions because of the definition of $\mathcal{E}$-equivalent labeled propositions (definition 6), thus forbidding in particular to introduce free variables in Γ' that were not present in the labels of Γ.

- In the **Reduction** rule, labels are extended by the disjunctive normal form transformation algorithm.
- In the **Identical Branch Closure**, eliminated propositions need not have the same label.

4.1. IC-TaMeD soundness

This sectios concentrates on the specific IC-TaMeD soundness proof. One main intermediate lemma is needed to prove the correctness of IC-TaMeD with respect to the sequent calculus modulo of figure 1.

LEMMA 2. *Let* $(\Gamma_1 \mid \ldots \mid \Gamma_n)$ *be a multiset of fxp-branches . If*

$$(\Gamma_1 \mid \ldots \mid \Gamma_n) \overset{IcT}{\hookrightarrow} \odot$$

then

$$\overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}}$$

.

Proof: This proof is made by induction on the structure of the IC-TaMeD derivation $\mathcal{T} = (\Gamma_1 \mid \ldots \mid \Gamma_n) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$.

If the IC-TaMeD derivation is empty, then no rule of IC-TaMeD can by definition be applied to $\mathcal{T}$. Hence the tableau $\mathcal{T}$ has been emptied by the tnf algorithm ; i.e. we actually have $\mathcal{T} \iff \bot$.

Otherwise the IC-TaMeD derivation $\Gamma_1 \mid \ldots \mid \Gamma_n \overset{\mathrm{IcT}}{\hookrightarrow} \odot$ starts by producing a new tableau $\mathcal{T}'$ and there is a shorter derivation of $\mathcal{T}' \overset{\mathrm{IcT}}{\hookrightarrow} \odot$. Let $\mathcal{T} = \Gamma_1 \mid \ldots \mid \Gamma_n$ be the tableau before application of the considered IC-TaMeD rule.

- **Case Identical Branch Closure:**
  There is a branch, say Γ_1, from $\mathcal{T}$ that contains a literal and its opposite (i.e. $\Gamma_1 = \mathcal{B} \wedge P_1 \wedge \neg P_1$). Thus we have $\mathcal{T}' = \Gamma_2 \mid \ldots \mid \Gamma_n$ and $\Gamma_1 \iff \bot$.

  The following equivalences can be easily deduced:

$$\begin{array}{ll} & \overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}} \overline{\forall}(\Gamma_2 \vee \ldots \vee \Gamma_n) \\ \iff & \overline{\forall}((\mathcal{B} \wedge \neg P_1 \wedge P_1) \vee \Gamma_2 \vee \Gamma_n) \vdash_{\mathcal{RE}} \overline{\forall}(\Gamma_2 \vee \ldots \vee \Gamma_n) \\ \iff & \overline{\forall}(\bot \vee \Gamma_2 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}} \overline{\forall}(\Gamma_2 \vee \ldots \vee \Gamma_n) \\ \iff & \overline{\forall}(\Gamma_2 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}} \overline{\forall}(\Gamma_2 \vee \ldots \vee \Gamma_n) \end{array}$$

By induction hypothesis $\mathcal{T}' \vdash_{\mathcal{RE}}$, i.e. $\forall(\Gamma_2 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}}$, hence, with the help of the cut rule $\overline{\forall}\Gamma_1 \vee \Gamma_2 \vee \ldots \vee \Gamma_n \vdash_{\mathcal{RE}}$.

– **Case Instantiation:**
$\mathcal{T}' = (\Gamma_1 \vee \ldots \vee \Gamma_n)[x := t]$ therefore, by induction hypothesis,

$$\overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n)[x := t] \vdash_{\mathcal{RE}}$$

The result is produced by the following sequent derivation:

$$\dfrac{\overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n)[x := t] \vdash_{\mathcal{RE}}}{\forall x\, \overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}}}\ \forall - r(\Gamma_1 \vee \ldots \vee \Gamma_n, x, t)$$

The definition of $\overline{\forall}$ ensures $\overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}}$.

– **Case Reduction:**
There is a branch of $\mathcal{T}$, say Γ_1, which contains an atomic proposition P that reduces to Q. Let $\Gamma_1' = \Gamma_1 \backslash \{P\}, Q$ and $tnf(\Gamma_1') = \mathcal{B}_1 \mid \ldots \mid \mathcal{B}_p$.

The proof will consist of three parts. First, we build the proof of the sequent:

$$\overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}} \overline{\forall}(\Gamma_1' \vee \Gamma_2 \vee \ldots \vee \Gamma_n) \tag{1}$$

Then, we will prove

$$\overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n), \overline{\forall}(\Gamma_1' \vee \Gamma_2 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}} \tag{2}$$

These results will entail the conclusion $\overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}}$ by using the cut-rule.

Let us start and prove the sequent (1) in the sequent calculus modulo of figure 1. As $\Gamma_1 =_{\mathcal{RE}} \Gamma_1'$, and by definition of the axiom sequent rule, $\Gamma_1 \vdash_{\mathcal{RE}} \Gamma_1'$ is provable. Therefore we can build the following derivation:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{}{\Gamma_1 \vdash_{\mathcal{RE}} \Gamma_1'}\ axiom}{\Gamma_1 \vdash_{\mathcal{RE}} \Gamma_1' \vee \Gamma_2}\ \vee - l \\ \vdots\end{array}}{\Gamma_1 \vdash_{\mathcal{RE}} \Gamma_1' \vee \ldots \vee \Gamma_n}\ \vee - l \qquad \begin{array}{c}\dfrac{\dfrac{}{\Gamma_2 \vdash_{\mathcal{RE}} \Gamma_2}\ axiom}{\Gamma_2 \vdash_{\mathcal{RE}} \Gamma_1' \vee \ldots \vee \Gamma_n}\ \vee - r* \\ \vdots\end{array}}{\Gamma_1 \vee \Gamma_2 \vdash_{\mathcal{RE}} \Gamma_1' \vee \ldots \vee \Gamma_n}\ \vee - l \qquad \begin{array}{c}(\Gamma_3) \\ \vdots\end{array}}{\begin{array}{c}\vdots \\ \Gamma_1 \vee \ldots \vee \Gamma_{n-1} \vdash_{\mathcal{RE}} \Gamma_1' \vee \Gamma_2 \vee \ldots \vee \Gamma_n\end{array}}\ \vee - l$$

And we use it to continue as follows:

$$
\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ \overline{\overline{\mathcal{V}}(\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}} \Gamma'_1 \vee \ldots \vee \Gamma_n}\end{array}}{\overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}} \forall x \; (\Gamma'_1 \vee \ldots \vee \Gamma_n)}\ \forall - r}{\vdots}}{\overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}} \overline{\forall}(\Gamma'_1 \vee \ldots \vee \Gamma_n)}\ \forall - r
$$

where the top of the derivation is:

$$
\begin{array}{c}
\dfrac{\dfrac{\text{above derivation} \quad \vdots \qquad \dfrac{\vdots}{\Gamma_n \vdash_{\mathcal{RE}} \Gamma'_1 \vee \ldots \vee \Gamma_n}}{\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}} \Gamma'_1 \vee \ldots \vee \Gamma_n}}{\forall x \, (\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}} \Gamma'_1 \vee \ldots \vee \Gamma_n} \\
\vdots \\
\overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}} \Gamma'_1 \vee \ldots \vee \Gamma_n
\end{array}
$$

Sequent (1) has been obtained, let us now prove sequent (2). Let $\mathcal{K}$ be a set of compatible axioms with $\mathcal{RE}$ as in definition 4.

As $\mathcal{K}$ is a set of sentences, it can be moved within or out the scope of universal quantifiers and still keeping equivalent first-order formulas. In particular:

$$
\begin{array}{c}
\mathcal{K} \wedge \overline{\forall}(\mathcal{B}_1 \vee \ldots \vee \mathcal{B}_p \vee \Gamma_2 \vee \ldots \vee \Gamma_n) \\
\Longleftrightarrow \\
\overline{\forall}(\mathcal{K} \wedge \mathcal{B}_1 \vee \ldots \vee \mathcal{B}_p \vee \Gamma_2 \vee \ldots \vee \Gamma_n)
\end{array}
$$

By induction hypothesis, with $\mathcal{T}' = \mathcal{B}_1 \mid \ldots \mid \mathcal{B}_p \mid \Gamma_2 \mid \ldots \mid \Gamma_n$, we have:

$$\overline{\forall}(\mathcal{B}_1 \vee \ldots \vee \mathcal{B}_p \vee \Gamma_2 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}}$$

i.e., by lemma 5

$$\mathcal{K}, \overline{\forall}(\mathcal{B}_1 \vee \ldots \vee \mathcal{B}_p \vee \Gamma_2 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}} \tag{3}$$

Moreover, we have by tnf:

$$\mathcal{K}, \Gamma'_1 \mid \mathcal{K}, \Gamma_2 \mid \ldots \mid \mathcal{K}, \Gamma_n \xrightarrow{tnf} \mathcal{K}, \mathcal{B}_1 \mid \ldots \mid \mathcal{K}, \mathcal{B}_p \mid \mathcal{K}, \Gamma_2 \mid \ldots \mid \mathcal{K}, \Gamma_n$$

Using the correction of tnf by lemma 1 and the result of sequent (3) we conclude that:

$$\mathcal{K}, \overline{\forall}(\Gamma'_1 \vee \Gamma_2 \vee \ldots \vee \Gamma_n) \vdash$$

i.e
$$\overline{\forall}(\Gamma'_1 \vee \Gamma_2 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}}$$

And by weakening the left side of this sequent, we get the sequent 2 given above:

$$\overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n), \overline{\forall}(\Gamma'_1 \vee \Gamma_2 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}}$$

We already proved: $\overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}} \overline{\forall}(\Gamma'_1 \vee \Gamma_2 \vee \ldots \vee \Gamma_n)$
Applying the cut rule of definition 1 to these two premisse yields the conclusion:

$$\overline{\forall}(\Gamma_1 \vee \Gamma_2 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}}$$

- **Case Conversion:**
  It suffices to follow the same steps as for the first part of the **Reduction** rule, as we have the same $\mathcal{RE}$-congruence between the original and the resulting branches.

$\Diamond$

PROPOSITION 7. (IC-TaMeD soundness). *Let $P_1, \ldots, P_n, Q_1, \ldots, Q_m$ be closed formulas. If*

$$tnf(P_1 \wedge \ldots \wedge P_n \wedge \neg Q_1 \wedge \ldots \wedge \neg Q_m) \overset{IcT}{\hookrightarrow} \odot$$

then
$$P_1, \ldots, P_n \vdash_{\mathcal{RE}} Q_1, \ldots, Q_m$$

Proof: Let $\{\Gamma_1 \mid \ldots \mid \Gamma_n\} = tnf(P_1 \wedge \ldots \wedge P_n \wedge \neg Q_1 \wedge \ldots \wedge \neg Q_m)$.
We have $(\Gamma_1 \mid \ldots \mid \Gamma_n) \overset{\text{IcT}}{\hookrightarrow} \odot$.
Hence, by using lemma 2: $\overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}}$ and by weakening $\mathcal{K}, \overline{\forall}(\Gamma_1 \vee \ldots \vee \Gamma_n) \vdash_{\mathcal{RE}}$, where $\mathcal{K}$ is a set of compatible axioms with $\mathcal{RE}$. We have

$$\{\mathcal{K}, P_1, \ldots, P_n, \neg Q_1, \ldots \neg Q_m\} \xrightarrow{tnf} \{(\mathcal{K}, \Gamma_1) \mid \ldots \mid (\mathcal{K}, \Gamma_n)\}$$

The tableau expansion soundness (lemma 1) entails:

$$\mathcal{K}, P_1, \ldots, P_n, \neg Q_1, \ldots \neg Q_m \vdash$$

Then we get by proposition 5

$$P_1, \dots, P_n, \neg Q_1, \dots \neg Q_m \vdash_{\mathcal{RE}}$$

or, stated in another way

$$P_1, \dots, P_n \vdash_{\mathcal{RE}} Q_1, \dots, Q_m$$

$\Diamond$

4.2. IC-TaMeD completeness

DEFINITION 15. (Function symbol transformation). *Let* t *be a term (resp. a proposition),* f *a function symbol of arity* n *and* u *a term whose free variables are among* $x_1, \dots, x_n$*. The individual transformation of symbol* f *into* u *is denoted by* $(x_1, \dots, x_n)u/f$*.* $\{(x_1, \dots, x_n)u/f\}t$ *denotes its application on a term (resp. a proposition)* t *and is obtained by replacing in* t *any subterm of the form* $f(v_1, \dots, v_n)$*, where* $v_1, \dots, v_n$ *are arbitrary terms by the term* $u[x_1 := v_1, \dots, x_n := v_n]$*.*

Given a finite set of indexes I*, the result of the application of a transformation of function symbols* $\rho = \{(x_1^i, \dots, x_n^i)u^i/f^i\}_{i \in I}$ *to a term (resp. a proposition)* t *is defined as the simultaneous application of the individual symbol transformations on* t*.*

Note that labels are not affected by such transformations.

LEMMA 3. *Let* $\mathcal{T}$ *be a tableau and* ρ *a transformation of function symbols. The Skolem symbols introduced when putting* $\mathcal{T}$ *in tableau normal form are assumed to be fresh, i.e. not transformed by* ρ*. Then* $tnf(\rho\mathcal{T}) = \rho tnf(\mathcal{T})$ *up to some renaming.*

Proof: Let us check that if we have two tableaus $\mathcal{T}$ and $\mathcal{T}'$ such that $\mathcal{T} \xrightarrow{tnf} \mathcal{T}'$, then $\rho\mathcal{T} = \rho\mathcal{T}'$. Let for example

$$\mathcal{T} = \mathcal{U} \mid \Gamma, \exists x\ P^{y_1, \dots, y_n}$$

and

$$\mathcal{T}' = \mathcal{U} \mid \Gamma, (P[x := f(y_1, \dots, y_n)])^{y_1, \dots, y_n}$$

then the set

$$\rho\mathcal{T} = \rho\mathcal{U} \mid \rho\Gamma, \exists x\ \rho P^{y_1, \dots, y_n}$$

transforms to

$$\rho\mathcal{U} \mid \rho\Gamma, (\rho P[x := f(y_1, \ldots, y_n)])^{y_1,\ldots,y_n} = \rho\mathcal{T}'$$

since f is assumed to be a fresh function symbol.

The complete result follows by induction on the length of the transformation of $\mathcal{T}$ to its tableau form. $\Diamond$

LEMMA 4. *Let $\mathcal{T}$ be a tableau and ρ a transformation of function symbols not appearing in $\mathcal{RE}$. If tnf($\mathcal{T}$) $\overset{IcT}{\hookrightarrow} \odot$ then tnf($\rho\mathcal{T}$) $\overset{IcT}{\hookrightarrow} \odot$ and the derivations have the same length.*

Proof: Lemma 3 yields that it suffices to prove that $\rho\mathrm{tnf}(\mathcal{T}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$. The proof proceeds by induction on the structure of the derivation to show more generally that for any fully expanded tableau if $\mathcal{T} \overset{\mathrm{IcT}}{\hookrightarrow} \odot$ then $\rho\mathcal{T} \overset{\mathrm{IcT}}{\hookrightarrow} \odot$

- For the **Instantiation** rule, we have $\rho(\mathcal{T}[x \mapsto t]) = (\rho\mathcal{T})[x \mapsto \rho t]$.
- For the rules **Conversion** and **Reduction**, if $\mathcal{B} =_{\mathcal{E}} \Gamma$ then, $\rho\mathcal{B} =_{\mathcal{E}} \rho\Gamma$ and if $\mathcal{B} \overset{\mathcal{RE}}{\to} \mathcal{B}'$, then $\rho\mathcal{B} \overset{\mathcal{RE}}{\to} \rho\mathcal{B}'$ as the symbols transformed by ρ do not appear in $\mathcal{RE}$.
- For the last rule, **Identical Branch Closure**, the case is obvious.

$\Diamond$

A new operator on tableaus ($\bigoplus$) is used in the next proofs and is defined as follows.

DEFINITION 16. ($\bigoplus$ operator). *Let $\mathcal{T} = \{\Gamma_1 \mid \ldots \mid \Gamma_n\}$ and $\mathcal{U} = \{\mathcal{B}_1 \mid \ldots \mid \mathcal{B}_p\}$. Then, the $\bigoplus$ operator will denote the following operation:*

$$\mathcal{T} \bigoplus \mathcal{U} = \bigcup_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}} \Gamma_i \cup \mathcal{B}_j$$

LEMMA 5. *Let t be a closed term, $\mathcal{T}$ a tableau, x a variable not occurring in $\mathcal{T}$, $\mathcal{B}$ a branch with labeled propositions then*

$$\mathcal{T} \bigoplus tnf(\mathcal{B}[x := t]) \overset{IcT}{\hookrightarrow} \odot \;\Rightarrow\; \mathcal{T} \bigoplus tnf(\mathcal{B}) \overset{IcT}{\hookrightarrow} \odot$$

Proof: We proceed by induction on the tnf-ordering following the structure of our tableau form transformation.

If $\mathcal{B}$ is a fxp-branch then $\mathrm{tnf}(\mathcal{B}[x := t]) = \mathcal{B}[x := t]$. Using the **Instantiation** rules, $\mathcal{B}[x := t]$ can be derived from $\mathcal{B}$, as x does not appear free in the rest of the tableau. Hence, if $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}[x := t]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$ then $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$.

Otherwise, there is a proposition $P \in \mathcal{B}$ which is not a literal. Let us detail the different possible cases, using the notation $\mathcal{B} = \mathcal{B}', P$.

- If $P = Q_1 \wedge Q_2$ then $P[x := t] = Q_1[x := t] \wedge Q_2[x := t]$. As

$$\mathrm{tnf}(\mathcal{B}) = \mathrm{tnf}(\mathcal{B}', Q_1, Q_2)$$

  and

$$\mathrm{tnf}(\mathcal{B}[x := t]) = \mathrm{tnf}(\mathcal{B}'[x := t], Q_1[x := t], Q_2[x := t])$$

  if $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}[x := t]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$, then

$$\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}'[x := t], Q_1[x := t], Q_2[x := t]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

  We therefore have by induction hypothesis

$$\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}', Q_1, Q_2) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

  i.e. $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$

- If $P = \neg \perp$ then $\mathrm{tnf}(\mathcal{B}) = \mathrm{tnf}(\mathcal{B}')$ and $\mathrm{tnf}(\mathcal{B}[x := t]) = \mathrm{tnf}(\mathcal{B}'[x := t])$. Thus, if $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}[x := t]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$, then

$$\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}'[x := t]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

  , which implies by induction hypothesis:

$$\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}') \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

  i.e. $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$

– If $P = \perp$ then $\mathrm{tnf}(\mathcal{B}) = \emptyset = \mathrm{tnf}(\mathcal{B}[x := t])$ Thus, if

$$\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}[x := t]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

then, as $\mathrm{tnf}(\mathcal{B}[x := t]) = \emptyset$,

$$\mathcal{T} \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

therefore, with definition 16, we obviously get $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$.

– If $P = \neg\neg Q$ then $P[x := t] = \neg\neg Q[x := t]$ and

$$\mathrm{tnf}(\mathcal{B}) = \mathrm{tnf}(\mathcal{B}', Q)$$

and also

$$\mathrm{tnf}(\mathcal{B}[x := t]) = \mathrm{tnf}(\mathcal{B}'[x := t], Q[x := t])$$

. Thus, if $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}[x := t]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$, then

$$\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}'[x := t], Q[x := t]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

Using the induction hypothesis, we get

$$\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}', Q) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

i.e. $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$.

– If $P = Q_1 \vee Q_2$ then $P[x := t] = Q_1[x := t] \vee Q_2[x := t]$.

$$\mathrm{tnf}(\mathcal{B}) = \mathrm{tnf}(\mathcal{B}', Q_1) \mid \mathrm{tnf}(\mathcal{B}', Q_2)$$

therefore

$$\mathrm{tnf}(\mathcal{B}[x := t]) = \mathrm{tnf}(\mathcal{B}'[x := t], Q_1[x := t]) \mid \mathrm{tnf}(\mathcal{B}'[x := t], Q_2[x := t])$$

Thus if $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}[x := t]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$, then

$$\mathcal{T} \bigoplus (\mathrm{tnf}(\mathcal{B}'[x := t], Q_1[x := t]) \mid \mathrm{tnf}(\mathcal{B}'[x := t], Q_2[x := t])) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

which can be rewritten as

$$\mathcal{T} \bigoplus (\mathrm{tnf}(\mathcal{B}', Q_1) \mid \mathrm{tnf}(\mathcal{B}', Q_2))[x := t] \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

and we can obtain, using the induction hypothesis

$$\mathcal{T} \bigoplus (\mathrm{tnf}(\mathcal{B}', Q_1) \mid \mathrm{tnf}(\mathcal{B}', Q_2])) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

i.e. $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$.

- If $P = \forall y\ Q$ then $P[x := t] = \forall y\ Q[x := t]$. There are two cases, following the value of n_y^γ
  - If $n_y^\gamma = 1$, $\mathrm{tnf}(\mathcal{B}) = \mathrm{tnf}(\mathcal{B}', Q)$ and

    $$\mathrm{tnf}(\mathcal{B}[x := t]) = \mathrm{tnf}(\mathcal{B}'[x := t], Q[x := t])$$

    Thus, if $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}[x := t]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$, then

    $$\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}'[x := t], Q[x := t]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

    Applying the induction hypothesis we get $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}', Q) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$, i.e. $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B})$.
  - Otherwise, if $n_y^\gamma > 1$, $\mathrm{tnf}(\mathcal{B}) = \mathrm{tnf}(\mathcal{B}, Q)$. Then, in the same way as before:

    $$\mathrm{tnf}(\mathcal{B}[x := t]) = \mathrm{tnf}(\mathcal{B}[x := t], Q[x := t])$$

    Thus, if $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}[x := t]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$, then

    $$\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}[x := t], Q[x := t]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

    Applying the induction hypothesis (n has indeed decreased by 1) we get $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}, Q) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$, i.e. $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$.

- If $P = \exists z\ Q$, then $P[x := t] = (\exists z\ Q)[x := t]$ If x does not appear in the label of P then it is not free in P and the case is obvious. Otherwise let $y_1, \ldots, y_n, x$ be the label of P and we therefore have:

  $$\mathrm{tnf}(\mathcal{B}[x := t]) = \mathrm{tnf}(\mathcal{B}'[x := t], Q[x := t][z := g(y_1, \ldots, y_n, x)])$$

where g is a fresh Skolem symbol. By induction hypothesis

$$\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}'[x := t], Q[x := t][z := g(y_1, \ldots, y_n)]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

Let $\rho = \{(y_1, \ldots, y_n) f(y_1, \ldots, y_n, t)/g\}$, then by Lemmas 3 and 4, we get:

$$\begin{array}{rl} & \mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}'[x := t], Q[x := t][z := g(y_1, \ldots, y_n)]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot \\ \Longrightarrow_\rho & \mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}'[x := t], Q[x := t][z := f(y_1, \ldots, y_n, t)]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot \\ \Longleftrightarrow & \mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}', Q[z := f(y_1, \ldots, y_n, x)])[x := t] \overset{\mathrm{IcT}}{\hookrightarrow} \odot \end{array}$$

Then, we have by applying the induction hypothesis to the last line above.

$$\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}', Q[z := f(y_1, \ldots, y_n, x)]) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

i.e. $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$.

– The cases $\neg(Q_1 \vee Q_2)$ and $\neg(Q_1 \Rightarrow Q_2)$ are similar to the case $Q_1 \wedge Q_2$. The cases $\neg(Q_1 \wedge Q_2)$ and $Q_1 \Rightarrow Q_2$ are similar to the case $Q_1 \vee Q_2$. The case $\neg(\forall z\, Q)$ is similar to the case $\exists z\, Q$ and the case $\neg(\exists z\, Q)$ to $\forall z\, Q$.

$\Diamond$

LEMMA 6. *Let $\mathcal{B} = \{P_1, \ldots, P_n\}$ and $\Gamma = \{Q_1, \ldots, Q_n\}$ be two branches of labeled propositions such that, for every i, $P_i \longrightarrow^*_{\mathcal{RE}} Q_i$. If $\mathcal{T} \bigoplus tnf(\Gamma) \hookrightarrow \odot$, then $\mathcal{T} \bigoplus tnf(\mathcal{B}) \hookrightarrow \odot$*

Proof: This proof is done by induction on $\mathcal{B}$ using tnf-ordering.

First, if $P \overset{\mathcal{R}}{\to} Q$ then, for every branch of tnf($\mathcal{B}$), we can find a branch of tnf(Γ) from which we can derive the branch from tnf($\mathcal{B}$) using the **Reduction** rule.

Else if $P =_{\mathcal{E}} Q$, then for every branch of tnf($\mathcal{B}$), we can find a branch of tnf(Γ) from which we can derive the branch of tnf($\mathcal{B}$) using the **Conversion** rule. Hence, if all the propositions of $\mathcal{B}$ are literals, tnf($\{\mathcal{B}\}$) $= \mathcal{B}$ and we can derive from $\mathcal{B}$ all the branches of tnf(Γ) using the **Conversion** and **Reduction** rules. In this case, the result is direct: as $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}) \overset{\mathrm{IcT}}{\hookrightarrow}_{Conv.,Red.} \mathcal{T} \bigoplus \mathrm{tnf}(\Gamma) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$ if $\mathcal{T} \bigoplus \mathrm{tnf}(\Gamma) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$ then $\mathcal{T} \bigoplus \mathrm{tnf}(\mathcal{B}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$.

Let us recall that if we want to use $P \longrightarrow_{\mathcal{RE}} Q$, we need the same labels for P and Q. We will now assume that there is a non-literal proposition P in $\mathcal{B}$ which can be reduced to Q. Let $\mathcal{B} = \mathcal{B}', P$ and $\Gamma = \Gamma', Q$ and let us now detail the different cases.

- If $P = R_1 \wedge R_2$, then $Q = R_1' \wedge R_2', R_1 \longrightarrow_{\mathcal{RE}} R_1', R_2 \longrightarrow_{\mathcal{RE}} R_2'$. We have: $\mathrm{tnf}(\{\mathcal{B}\}) = \mathrm{tnf}(\{\mathcal{B}', R_1, R_2\})$ and $\mathrm{tnf}(\{\Gamma\}) = \mathrm{tnf}(\{\Gamma', R_1', R_2'\})$. Thus, if

  $$\mathcal{T} \bigoplus \mathrm{tnf}(\{\Gamma', R_1', R_2'\}) \stackrel{\mathrm{IcT}}{\hookrightarrow} \odot$$

  then by induction hypothesis

  $$\mathcal{T} \bigoplus \mathrm{tnf}(\{\mathcal{B}', R_1, R_2\}) \stackrel{\mathrm{IcT}}{\hookrightarrow} \odot$$

  i.e. $\mathcal{T} \bigoplus \mathrm{tnf}(\{\mathcal{B}\}) \stackrel{\mathrm{IcT}}{\hookrightarrow} \odot$.

- If $P =\perp$, then $Q =\perp$. Furthermore, $\mathrm{tnf}(\{\mathcal{B}\}) = \mathrm{tnf}(\{\Gamma\}) = \emptyset$.

  Hence if

  $$\mathcal{T} \bigoplus \mathrm{tnf}(\{\Gamma\}) \stackrel{\mathrm{IcT}}{\hookrightarrow} \odot$$

  then $\mathcal{T} \stackrel{\mathrm{IcT}}{\hookrightarrow} \odot$ and therefore $\mathcal{T} \bigoplus \mathrm{tnf}(\{\mathcal{B}\})$.

- If $P = \neg \perp$ then $Q \neg \perp$. Hence, $\mathrm{tnf}(\{\mathcal{B}\}) = \mathrm{tnf}(\{\mathcal{B}'\})$ and $\mathrm{tnf}(\{\Gamma\}) = \mathrm{tnf}(\{\Gamma'\})$. Thus if

  $$\mathcal{T} \bigoplus \mathrm{tnf}(\{\Gamma\}) \stackrel{\mathrm{IcT}}{\hookrightarrow} \odot$$

  , then

  $$\mathcal{T} \bigoplus \mathrm{tnf}(\{\Gamma'\}) \stackrel{\mathrm{IcT}}{\hookrightarrow} \odot$$

  and by induction hypothesis $\mathcal{T} \bigoplus \mathrm{tnf}(\{\mathcal{B}'\}) \stackrel{\mathrm{IcT}}{\hookrightarrow} \odot$. i.e. $\mathcal{T} \bigoplus \mathrm{tnf}(\{\mathcal{B}\}) \stackrel{\mathrm{IcT}}{\hookrightarrow} \odot$.

- If $P = \neg\neg R$ then $Q = \neg\neg R'$ and $R \longrightarrow_{\mathcal{RE}} R'$. Moreover $\mathrm{tnf}(\{\mathcal{B}\}) = \mathrm{tnf}(\{\mathcal{B}', R\})$ and $\mathrm{tnf}(\{\Gamma\}) = \mathrm{tnf}(\{\Gamma', R'\})$ Hence if

  $$\mathcal{T} \bigoplus \mathrm{tnf}(\{\Gamma\}) \stackrel{\mathrm{IcT}}{\hookrightarrow} \odot$$

i.e.
$$\mathcal{T} \bigoplus \mathrm{tnf}(\{\Gamma', R'\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$
and by induction hypothesis
$$\mathcal{T} \bigoplus \mathrm{tnf}(\{\mathcal{B}', R\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$
i.e. $\mathcal{T} \bigoplus \mathrm{tnf}(\{\mathcal{B}\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$.

– If $P = R_1 \vee R_2$, then $Q = R'_1 \vee R'_2$, $R_1 \longrightarrow_{\mathcal{RE}} R'_1$, $R_2 \longrightarrow_{\mathcal{RE}} R'_2$. We have:
$$\mathrm{tnf}(\{\mathcal{B}\}) = \mathrm{tnf}(\{\mathcal{B}', R_1\}) \mid \mathrm{tnf}(\{\mathcal{B}', R_2\})$$
and
$$\mathrm{tnf}(\{\Gamma\}) = \mathrm{tnf}(\{\Gamma', R'_1\}) \mid \mathrm{tnf}(\{\Gamma', R'_2\})$$
.Thus if
$$\mathcal{T} \bigoplus \mathrm{tnf}(\{\Gamma', R'_1\}) \mid \mathrm{tnf}(\{\Gamma', R'_2\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$
then by induction hypothesis
$$\mathcal{T} \bigoplus (\mathrm{tnf}(\{\Gamma', R'_1\}) \mid \mathrm{tnf}(\{\mathcal{B}', R_2\})) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$
and
$$\mathcal{T} \bigoplus (\mathrm{tnf}(\{\mathcal{B}', R_1\}) \mid \mathrm{tnf}(\{\mathcal{B}', R_2\})) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$
i.e. $\mathcal{T} \bigoplus \mathrm{tnf}(\{\mathcal{B}\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$.

– If $P = \forall y\ R$, then $Q = \forall y\ R'$ with $R \longrightarrow_{\mathcal{RE}} R'$. Moreover R and R' have the same labels. We have two cases, depending on the value of n_y.

  - If $n_y = 1$, then $\mathrm{tnf}(\{\mathcal{B}\}) = \mathrm{tnf}(\{\mathcal{B}', R\})$ and also $\mathrm{tnf}(\{\Gamma\}) = \mathrm{tnf}(\{\Gamma', R'\})$. In this case
$$\mathrm{tnf}(\{\Gamma\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$
    rewritten as
$$\mathrm{tnf}(\{\Gamma', R'\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

then by induction hypothesis

$$\mathrm{tnf}(\{\mathcal{B}', R\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

i.e $\mathrm{tnf}(\{\mathcal{B}\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$.

- If $n_y > 1$ then $\mathrm{tnf}(\{\mathcal{B}\}) = \mathrm{tnf}(\{\mathcal{B}, R\})$ and also $\mathrm{tnf}(\{\Gamma\}) = \mathrm{tnf}(\{\Gamma, R'\})$. In this case

$$\mathrm{tnf}(\{\Gamma\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

i.e.

$$\mathrm{tnf}(\{\Gamma, R'\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

and by induction hypothesis (n has decreased)

$$\mathrm{tnf}(\{\mathcal{B}, R\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

i.e. $\mathrm{tnf}(\{\mathcal{B}\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$.

– If $P = \exists z\ R$, then $Q = \exists z\ R'$ and $R \longrightarrow_{\mathcal{RE}} R'$. Let $y_1, \ldots, y_n$ be the common label of P and Q. The two refutations are independent so we choose the same Skolem symbol without loss of generality. We have

$$\mathrm{tnf}(\{\mathcal{B}\}) = \mathrm{tnf}(\{\mathcal{B}', R[x := f(y_1, \ldots, y_n)]\})$$

and

$$\mathrm{tnf}(\{\Gamma\}) = \mathrm{tnf}(\{\Gamma', R'[x := f(y_1, \ldots, y_n)]\})$$

. Hence if

$$\mathcal{T} \bigoplus \mathrm{tnf}(\{\Gamma\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

i.e.

$$\mathcal{T} \bigoplus \mathrm{tnf}(\{\Gamma', R'[x := f(y_1, \ldots, y_n)]\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

then by induction hypothesis

$$\mathcal{T} \bigoplus \mathrm{tnf}(\{\mathcal{B}', R[x := f(y_1, \ldots, y_n)]\}) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

i.e. $\mathcal{T} \bigoplus \mathrm{tnf}(\{\mathcal{B}\})$.

– The cases $\neg(R_1 \vee R_2)$ and $\neg(R_1 \Rightarrow R_2)$ are treated like the case $R_1 \wedge R_2$. The cases $\neg(R_1 \wedge R_2)$ and $\mathcal{R}_1 \Rightarrow R_2$ are similar to the case $R_1 \vee R_2$. Finally, the case $\neg(\forall z\ R)$ is similar to the case $\exists z\ R$ and the case $\neg(\exists y\ R)$ similar to $\forall y\ R$.

$\Diamond$

LEMMA 7. *Let R, S be closed formulas and Γ, Δ branches of closed formulas. If*

$$\mathit{tnf}(R, \Gamma, \neg\Delta) \overset{IcT}{\hookrightarrow} \odot \quad \textit{and} \quad \mathit{tnf}(S, \Gamma, \neg\Delta) \overset{IcT}{\hookrightarrow} \odot$$

Then we can build a derivation of:

$$\mathit{tnf}(R \vee S, \Gamma, \neg\Delta) \overset{IcT}{\hookrightarrow} \odot$$

Proof: A tableau is closed (here $\overset{\text{IcT}}{\hookrightarrow} \odot$) if every branch of this tableau is closed (also $\overset{\text{IcT}}{\hookrightarrow} \odot$). We know that $\text{tnf}(R \vee S, \Gamma, \neg\Delta) = \text{tnf}((R, \Gamma, \neg\Delta)|(S, \Gamma, \neg\Delta))$ from definition 7 and from lemma 1, that this calculus is sound. Moreover,

$$\text{tnf}(R, \Gamma, \neg\Delta) \mid \text{tnf}(S, \Gamma, \neg\Delta)) = \text{tnf}((R, \Gamma, \neg\Delta)|(S, \Gamma, \neg\Delta))$$

.

Hence, if

$$\text{tnf}(R, \Gamma, \neg\Delta) \overset{\text{IcT}}{\hookrightarrow} \odot$$

and

$$\text{tnf}(S, \Gamma, \neg\Delta) \overset{\text{IcT}}{\hookrightarrow} \odot$$

then, providing that all formulas are closed.

$$\text{tnf}(R, \Gamma, \neg\Delta) \mid \text{tnf}(S, \Gamma, \neg\Delta)) = \text{tnf}((R, \Gamma, \neg\Delta) \mid (S, \Gamma, \neg\Delta)) \overset{\text{IcT}}{\hookrightarrow} \odot$$

and finally

$$\text{tnf}(R \vee S, \Gamma, \neg\Delta) \overset{\text{IcT}}{\hookrightarrow} \odot$$

$\Diamond$

The following lemma allows the restriction of the use of the congruence to reductions in proofs and comes directily from (Dowek et al., 2003).

LEMMA 8. *If the relation* $\longrightarrow_{\mathcal{RE}}$ *is confluent, we have:*

- *If* P *and* $Q \wedge R$ *are sentences such that* $P =_{\mathcal{RE}} Q \wedge R$*, then there exists a sentence* $Q' \wedge R'$ *such that* $P \longrightarrow^*_{\mathcal{RE}} Q' \wedge R', Q =_{\mathcal{RE}} Q'$ *and* $P =_{\mathcal{RE}} P'$.

- *If* P *and* $Q \vee R$ *are sentences such that* $P =_{\mathcal{RE}} Q \vee R$*, then there exists a sentence* $Q' \vee R'$ *such that* $P \longrightarrow^*_{\mathcal{RE}} Q' \vee R', Q =_{\mathcal{RE}} Q'$ *and* $P =_{\mathcal{RE}} P'$.

- *If* P *and* $Q \Rightarrow R$ *are sentences such that* $P =_{\mathcal{RE}} Q \Rightarrow R$*, then there exists a sentence* $Q' \Rightarrow R'$ *such that* $P \longrightarrow^*_{\mathcal{RE}} Q' \Rightarrow R', Q =_{\mathcal{RE}} Q'$ *and* $P =_{\mathcal{RE}} P'$.

- *If* P *and* $\neg Q$ *are sentences such that* $P =_{\mathcal{RE}} \neg Q$*, then there exists a sentence* $\neg Q'$ *such that* $P \longrightarrow^*_{\mathcal{RE}} \neg Q'$*and* $Q =_{\mathcal{RE}} Q'$.

- *If* P *is a sentence such that* $P =_{\mathcal{RE}} \bot$*, alors* $P \longrightarrow^*_{\mathcal{RE}} \bot$.

- *If* P *and* $\forall x Q$ *are sentences such that* $P =_{\mathcal{RE}} \forall x Q$*, then there exists a sentence* $\forall x Q'$ *such that* $P \longrightarrow^*_{\mathcal{RE}} \forall x Q'$*and* $Q =_{\mathcal{RE}} Q'$.

- *If* P *and* $\exists x Q$ *are sentences such that* $P =_{\mathcal{RE}} \exists x Q$*, then there exists a sentence* $\exists x Q'$ *such that* $P \longrightarrow^*_{\mathcal{RE}} \exists x Q'$ *and* $Q =_{\mathcal{RE}} Q'$.

Proof: See (Dowek et al., 2003). $\diamondsuit$

PROPOSITION 8. (IC-TaMeD completeness). *Let* $\longrightarrow^*_{\mathcal{RE}}$ *be a confluent relation and* $P_1, \ldots, P_n, Q_1, \ldots, Q_m$ *be closed formulas. If the sequent:*

$$P_1, \ldots, P_n \vdash_{\mathcal{RE}} Q_1, \ldots, Q_m$$

has a cut-free proof then:

$$tnf(P_1, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{IcT}{\hookrightarrow} \odot$$

Proof: By induction on the size of a closed cut-free proof of:

$$P_1, \ldots, P_n \vdash_{\mathcal{RE}} Q_1, \ldots, Q_m$$

– If the last rule is the axiom rule, then $n = m = 1$ and $P_1 =_{\mathcal{RE}} Q_1$. By confluence, there exist sentences R and R' such that $P_1 \longrightarrow^*_{\mathcal{RE}} R, Q_1 \longrightarrow^*_{\mathcal{RE}} R'$ and $R =_{\mathcal{RE}} R'$. By induction on the structure of R and using the rules **Conversion**, **Instantiation**, and **Identical Branch Closure** , we prove $\mathrm{tnf}(R, \neg R') \overset{\mathrm{IcT}}{\hookrightarrow} \odot$ and by lemma 6 we get:

$$\mathrm{tnf}(P_1, \neg Q_1) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

– If the last rule is $contr - l$ or $contr - r$, then the tnf of the antecedent and succedent are the same, thus we simply apply the induction hypothesis.

– If the last rule is $weak - l$ or $weak - r$, then the tnf of the antecedent is a subset of the tnf of the succedent, thus we simply apply the induction hypothesis. In the case of $weak - l$ (resp. $weak - r$) and the duplication of a universally quantified formula (resp. existentally quantified), it is needed to allow one more γ-expansion, therefore n is increased.

– If the last rule is $\wedge$-l, then of the P_i (say P_1) is $\mathcal{RE}$-equivalent to a conjunction $R \wedge S$. By lemma 8 $P_1 \longrightarrow^*_{\mathcal{RE}} R' \wedge S'$ with $R =_{\mathcal{RE}} R', S =_{\mathcal{RE}} S'$. By induction hypothesis and proposition 2:

$$\mathrm{tnf}(R', S', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

and

$$\mathrm{tnf}(R' \wedge S', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

With lemma 6

$$\mathrm{tnf}(P_1, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

– If the last rule is $\neg - l$, one of the P_i (say P_1) is $\mathcal{RE}$-equivalent to a negation $\neg R$. By lemma 8 $P \longrightarrow^*_{\mathcal{RE}} \neg R'$ and $R' =_{\mathcal{RE}} R$. By induction hypothesis and proposition 2:

$$\mathrm{tnf}(\neg R', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

i.e. with lemma 6

$$\mathrm{tnf}(P_1, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

- If the last rule is $\bot -l$, one of the P_i (say P_1) is $\mathcal{RE}$-equivalent to $\bot$. By lemma 8 $P \longrightarrow^*_{\mathcal{RE}} \bot$.

$$\mathrm{tnf}(\bot, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

With lemma 6

$$\mathrm{tnf}(P_1, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

- If the last rule is $\vee$-l, one P_i (say P_1) is $\mathcal{RE}$-equivalent to a disjunction $R \vee S$. By lemma 8 $P \longrightarrow^*_{\mathcal{RE}} R' \vee S'$ and $R' =_{\mathcal{RE}} R, S' =_{\mathcal{RE}} S$. By induction hypothesis and proposition 2:

$$\mathrm{tnf}(R', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

and

$$\mathrm{tnf}(S', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

Lemma 7 can be used as instantiation of universally quantified variables are not delayed in the sequent calculus modulo. Therefore rigid free variables do not appear in the branches by this β-expansion step.

$$\mathrm{tnf}(R' \vee S', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

And with lemma 6

$$\mathrm{tnf}(P_1, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

- If the last rule is $\forall - l$, one of the P_i (say P_1) is $\mathcal{RE}$-equivalent to a universal proposition $\forall R$. By lemma 8 $P \longrightarrow^*_{\mathcal{RE}} \forall R'$ and $R' =_{\mathcal{RE}} R$. By induction hypothesis and proposition 2:

$$\mathrm{tnf}(R'[x := t], P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

for a given term t . Labels of $P_1, \forall x\, R', R'[x := t]$ are empty and the one of R' is x (which is a fresh variable). i.e.

$$\mathcal{T} = \mathrm{tnf}(P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m)$$

By lemma 5, $\mathcal{T} \bigoplus \mathrm{tnf}(R') \overset{\mathrm{IcT}}{\hookrightarrow} \odot$ i.e. $\mathcal{T} \bigoplus \mathrm{tnf}(\forall x\, R') \overset{\mathrm{IcT}}{\hookrightarrow} \odot$ and by lemma 6

$$\mathrm{tnf}(P_1, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

– If the last rule is $\exists - l$, one of the P_i (say P_1) is $\mathcal{RE}$-equivalent to an existential proposition $\exists R$. By lemma 8 $P \longrightarrow^*_{\mathcal{RE}} \exists R'$ and $R' =_{\mathcal{RE}} R$. By induction hypothesis and proposition 2:

$$\mathrm{tnf}(R'[x := c], P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

where c is a new constant, i.e.

$$\mathrm{tnf}(\exists x\, R', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

By lemma 6

$$\mathrm{tnf}(P_1, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

– If the last rule is $\wedge - r$ or $\Rightarrow -l$, the proof is like the one of $\vee - l$. If the last rule is $\vee - r$ or $\Rightarrow -r$, the proof is the one of $\wedge - l$. If the last rule is $\neg - r$ then the proof is like the one of $\neg - l$

$\diamondsuit$

5. Soundness and completeness of the TaMeD method

The soundness and completeness results obtained for IC-TaMeD will be lifted to TaMeD in the following subsections. Interactions between tableau forms and substitutions are handled by special lemmas in both cases.

5.1. TaMeD soundness

Proving the soundnes of TaMeD relies on one intermediate lemma.

LEMMA 9. *Let $\mathcal{B}$ be a branch of labeled propositions and Θ a closed substitution such that free variables introduce by putting $\mathcal{B}$ in tnf do not appear in Θ. Then, there exists a transformation ρ of the function symbols introduced by putting $\mathcal{B}$ in tableau normal form such that tnf($\Theta\mathcal{B}$) = $\rho\Theta$tnf($\mathcal{B}$).*

Proof: This proof is done by induction on $\mathcal{B}$ with the tnf-ordering.

If all the propositions of $\mathcal{B}$ are literals then

$$\mathrm{tnf}(\Theta\mathcal{B}) = \Theta\mathcal{B} = \Theta\mathrm{tnf}(\mathcal{B})$$

In this case we only have to take the identity for ρ. Otherwise there is a proposition P in $\mathcal{B}$ that is not a literal. Let $\mathcal{B} = \mathcal{B}' \cup \{P\}$.

- If $P = Q_1 \wedge Q_2$ then by induction hypothesis

  $$\mathrm{tnf}(\{\Theta\mathcal{B}', \Theta Q_1, \Theta Q_2\}) = \rho\Theta\mathrm{tnf}(\{\mathcal{B}', Q_1, Q_2\})$$

  i.e.

  $$\mathrm{tnf}(\Theta\mathcal{B}) = \rho\Theta\mathrm{tnf}(\mathcal{B})$$

- If $P =\perp$ then we have $\mathrm{tnf}(\Theta\mathcal{B}) = \emptyset = \mathrm{tnf}(\mathcal{B})$. Thus, ρ is the identity.

- If $P = \neg\perp$ then by induction hypothesis

  $$\mathrm{tnf}(\Theta\mathcal{B}') = \rho\Theta\mathrm{tnf}(\mathcal{B}')$$

  i.e.

  $$\mathrm{tnf}(\Theta\mathcal{B}) = \rho\Theta\mathrm{tnf}(\mathcal{B})$$

- If $P = \neg\neg Q$ then by induction hypothesis, we have

  $$\mathrm{tnf}(\Theta(\mathcal{B}', Q)) = \rho\Theta\mathrm{tnf}(\mathcal{B}', Q)$$

  i.e.

  $$\mathrm{tnf}(\Theta\mathcal{B}) = \rho\Theta\mathrm{tnf}(\mathcal{B})$$

- If $P = Q_1 \vee Q_2$ then by induction hypothesis

  $$\mathrm{tnf}(\{\Theta\mathcal{B}', \Theta Q_1\}) = \rho\Theta\mathrm{tnf}(\{\mathcal{B}', Q_1\})$$

  and

  $$\mathrm{tnf}(\{\Theta\mathcal{B}', \Theta Q_2\}) = \rho'\Theta\mathrm{tnf}(\{\mathcal{B}', Q_2\})$$

  Since the domains of ρ and ρ' are disjoint (Skolem symbols are assumed to be fresh):

  $$\begin{aligned}\mathrm{tnf}(\Theta\mathcal{B}) &= \mathrm{tnf}(\Theta\mathcal{B}', \Theta Q_1)|\mathrm{tnf}(\Theta\mathcal{B}', \Theta Q_2)\\ &= (\rho\rho')\Theta(\mathrm{tnf}(\mathcal{B}', Q_1)|\mathrm{tnf}(\mathcal{B}', Q_2))\\ &= (\rho\rho')\Theta\mathrm{tnf}(\mathcal{B})\end{aligned}$$

- If $P = \forall x\ Q$, two cases arise:
  - If $n_x = 1$ then as x is not in the domain of Θ and $\mathrm{tnf}(\Theta\mathcal{B}) = \mathrm{tnf}(\Theta\mathcal{B}', \Theta Q) = \rho\Theta\mathrm{tnf}(\mathcal{B}', Q) = \rho\Theta\mathrm{tnf}(\mathcal{B})$
  - Otherwise,$n_x > 1$. x is still not in the domain of Θ and $\mathrm{tnf}(\Theta\mathcal{B}) = \mathrm{tnf}(\Theta\mathcal{B}, \Theta Q)$. We obtain the result by using the induction hypothesis in addition to the precedent equality $\mathrm{tnf}(\Theta\mathcal{B}) = \mathrm{tnf}(\Theta\mathcal{B}, \Theta Q) = \rho\Theta\mathrm{tnf}(\mathcal{B}, Q) = \rho\Theta\mathrm{tnf}(\mathcal{B})$ (recalling that $\mathrm{tnf}(\mathcal{B}) = \mathrm{tnf}(\mathcal{B}, Q)$).

- If $P = \exists x\ Q$ then let $y_1, \ldots, y_p$ be the variables in the label of P that are in the domain of Θ and $z_1, \ldots, z_q$ the others. The label of P is thus $y_1, \ldots, y_p, z_1, \ldots, z_q$ and the label of ΘP is $z_1, \ldots, z_q$ because Θ is closed.

  We have

$$\mathrm{tnf}(\mathcal{B}) = \mathrm{tnf}(\mathcal{B}', Q[x := g(y_1, \ldots, y_p, z_1, \ldots, z_q)]$$

$$\mathrm{tnf}(\Theta\mathcal{B}) = \mathrm{tnf}(\Theta\mathcal{B}', (\Theta Q)[x := f(z_1, \ldots, z_q)]$$

  Moreover

$$\Theta(Q[x := g(y_1, ..., y_p, z_1, ..., z_q)]) = \Theta Q[x := g(\Theta y_1, ..., \Theta y_p, z_1, ..., z_q)]$$

  If we take $\rho = \{(y_1, \ldots, y_p, z_1, \ldots, z_q) f(z_1, \ldots, z_q)/g\}$ we get

$$\begin{aligned} &\rho\Theta Q[x := g(y_1, \ldots, y_p, z_1, \ldots, z_q)] = \\ &(\Theta Q)[x := f(z_1, \ldots, z_q)] = \\ &\Theta(Q[x := f(z_1, \ldots, z_q)]) \end{aligned}$$

  Thus

$$\rho\Theta(\mathcal{B}', Q[x := g(y_1, ..., y_p, z_1, \ldots, z_q)]) = \Theta(\mathcal{B}', Q[x := f(z_1, ..., z_q)])$$

  Hence

$$\begin{aligned} &\mathrm{tnf}(\Theta\mathcal{B}) = \\ &\mathrm{tnf}(\Theta(\mathcal{B}', Q[x := g(y_1, \ldots, y_p, z_1, \ldots, z_q)])) = \\ &\mathrm{tnf}(\rho\Theta(\mathcal{B}', Q[x := g(y_1, \ldots, y_p, z_1, \ldots, z_q)])) \end{aligned}$$

  By lemma 3

$$\mathrm{tnf}(\Theta\mathcal{B}) = \rho\mathrm{tnf}(\Theta(\mathcal{B}', Q[x := g(y_1, \ldots, y_p, z_1, \ldots, z_q)]))$$

And by induction hypothesis

$$\begin{aligned}&\mathrm{tnf}(\Theta(\mathcal{B}', Q[x := g(y_1, \ldots, y_p, z_1, \ldots, z_q)])) = \\ &\rho_0\Theta\mathrm{tnf}(\mathcal{B}', Q[x := g(y_1, \ldots, y_p, z_1, \ldots, z_q)]) = \\ &\rho_0\Theta\mathrm{tnf}(\mathcal{B})\end{aligned}$$

Hence $\mathrm{tnf}(\Theta\mathcal{B}) = \rho\rho_0\Theta\mathrm{tnf}(\mathcal{B})$.

$\Diamond$

LEMMA 10. *Let $\mathcal{T}$ be a tableau composed of globally constrained branches such that:*

$$\mathcal{T} \overset{T}{\mapsto} \odot[C]$$

and Θ be a closed substitution, unifier of C, mapping all the variables of C to a closed term. Then

$$\Theta\mathcal{T} \overset{IcT}{\hookrightarrow} \odot$$

Proof: This proof is done by induction on the structure of the derivation $\mathcal{T} \overset{T}{\mapsto} \odot[C]$. For the rest of the proof, let $\mathcal{T} = \{\mathcal{B}_1 \mid \ldots \mid \mathcal{B}_n\}$.

If the derivation is empty, $\mathcal{T}$ has been emptied by the tnf algorithm and obviously $\Theta\mathcal{T} \overset{\mathrm{IcT}}{\hookrightarrow} \odot$.

Otherwise, the derivation starts by producing a tableau $\mathcal{T}'$ from $\mathcal{T}$ and $\mathcal{T}'$ has a smaller derivation. Furthermore we have by induction hypothesis $\Theta\mathcal{T}' \overset{\mathrm{IcT}}{\hookrightarrow} \odot$. We look at the rule used to produce $\mathcal{T}'$.

- **Case Extended Branch Closure:**
  There is a branch, say $\mathcal{B}_1$ of $\mathcal{T}$ that contains two propositions P and $\neg Q$ such that $P =_{\mathcal{E}} Q$. Since all constraints of $\mathcal{T}$ are constraints of $\mathcal{T}'$, they are unified by Θ. $\Theta\mathcal{T}'$ can be produced from $\Theta\mathcal{T}$ by the rules **Conversion** and **Identical Branch Closure**. By induction hypothesis $\Theta\mathcal{T}' \overset{\mathrm{IcT}}{\hookrightarrow} \odot$, and $\Theta\mathcal{T} \overset{\mathrm{IcT}}{\hookrightarrow} \Theta\mathcal{T}'$, thus we get $\Theta\mathcal{T} \overset{\mathrm{IcT}}{\hookrightarrow} \odot$.

- **Case Extended Narrowing:**
  There is a branch of $\mathcal{T}$, let us call it Γ , that can be rewritten to Γ' and $\mathcal{U} = \mathrm{tnf}(\Gamma') = \mathcal{B}_1 \mid \ldots \mid \mathcal{B}_p$ such that $\mathcal{T}' = \mathcal{B}_1 \mid \ldots \mid \mathcal{B}_p \mid \Gamma_2 \mid \ldots \mid \Gamma_n$. Let also ω be the occurrence of Γ where the **Extended Narrowing** is applied such that

$\Gamma' = \Gamma[r]_\omega)$. Therefore $\Theta(\Gamma[l]_\omega) \rightarrow_\mathcal{R} \Theta\Gamma'$.
$\mathcal{T}'$ is constrained by $\Gamma_{|\omega} =^?_\mathcal{E} l$, hence $\Theta\Gamma_{|\omega} =_\mathcal{E} \Theta l$, $\Theta\Gamma =_\mathcal{E} \Theta(\Gamma[l]_\omega)$. As Θl is closed, we can put the same label on these propositions and further derive $\Theta\Gamma[l]_\omega$ from $\Theta\Gamma$ by using the rule **Conversion**.
Now, let Θ_1 be the restriction of Θ to the free variables produced by the variables bound in Γ' and Θ_2 its restriction to the other variables.
By lemma 9, there is a transformation ρ of the function symbols introduced by putting Γ' in tnf such that $\mathrm{tnf}(\Theta_2\Gamma') = \rho\Theta_2\mathcal{U}$. Hence, $\mathrm{tnf}(\Theta\Gamma') = \rho\mathrm{tnf}(\Theta_2\Theta_1\Gamma') = \rho\Theta_2\mathcal{U}$ as Θ_1 only applies to variables freed in $\mathcal{U}$
Moreover $\Theta_1\mathrm{tnf}(\mathcal{U}) = \Theta_1\rho\Theta_2\mathcal{U} = \rho\Theta\mathcal{U}$ as $\Theta\rho = \rho\Theta$. The set of branches $\rho\Theta\mathcal{U}$ can be derived from $\Theta\Gamma[\Theta l]_\omega$ with the **Reduction** rule and the **Instantiation** rule.
We have $\Theta\mathcal{T}' \overset{\mathrm{IcT}}{\hookrightarrow} \odot$, hence by lemma 4:

$$\rho\Theta\mathcal{T}' \overset{\mathrm{IcT}}{\hookrightarrow} \odot$$

We produce the following IC-TaMeD derivation from $\Theta\mathcal{T}$

$$\begin{array}{ll}
 & \Theta\mathcal{T} = \Theta\Gamma_1 \mid \ldots \mid \Theta\Gamma_n \\
\overset{\mathrm{IcT}}{\hookrightarrow}_{Conversion} & \Theta\Gamma_1[\Theta l]_{|\omega} \mid \ldots \mid \Theta\Gamma_n \\
\overset{\mathrm{IcT}}{\hookrightarrow}_{Reduction} & \mathrm{tnf}(\Theta\Gamma_1[\Theta r]_{|\omega} \mid \ldots \mid \Theta\Gamma_n) \\
= & \rho\Theta\mathrm{tnf}(\Gamma_1[r]_{|\omega} \mid \ldots \mid \Theta\Gamma_n) \\
= & \rho\Theta\mathcal{B}_1 \mid \ldots \mid \rho\Theta\mathcal{B}_p | \rho\Theta\Gamma_2 \mid \ldots \mid \rho\Theta\Gamma_n \overset{\mathrm{IcT}}{\hookrightarrow} \odot
\end{array}$$

Therefore $\Theta\mathcal{T} \overset{\mathrm{IcT}}{\hookrightarrow} \odot$

$\Diamond$

PROPOSITION 9. (TaMeD soundness). *Let $\mathcal{T}$ be a non-constrained tableau such that*

$$\mathcal{T} \overset{T}{\mapsto} \odot[C]$$

where C is a unifiable set of constraints. Then $\mathcal{T} \overset{IcT}{\hookrightarrow} \odot$.

Proof: The set of constraints C is unifiable and thus it has a unifier Θ mapping all variables of C. By lemma 10 $\Theta\mathcal{T} \overset{\mathrm{IcT}}{\hookrightarrow} \odot$. All the branches of $\Theta\mathcal{T}$ can be derived from those of $\mathcal{T}$ itself with the **Instantiation** rules. Hence $\mathcal{T} \overset{\text{IcT}}{\hookrightarrow} \odot$. $\Diamond$

Since IC-TaMeD is sound, the previous proposition entails immediately TaMeD soundness.

5.2. TaMeD completeness

The completeness result will now be lifted from IC-TaMeD to TaMeD in a way similar to the lifting of the soundness proof. A lemma similar to lemma 10 is also needed.

LEMMA 11. *Let $\mathcal{B}$ and Γ be two branches of labeled propositions. Let Θ be a substitution such that no variable bound in $\mathcal{B}$ is in the domain of Θ. Suppose that $\Theta\mathcal{B} =_{\mathcal{E}} \Gamma$. Then there is a transformation ρ of function symbols introduced by putting Γ in tableau normal form such that $\Theta tnf\mathcal{B}) =_{\mathcal{E}} \rho tnf\Gamma)$*

Proof: The proof is done by induction on $\mathcal{B}$ using tnf-ordering.

If all the propositions are literals then

$$\Theta\text{tnf}\mathcal{B}) = \Theta\mathcal{B} =_{\mathcal{E}} \Gamma = \text{tnf}\Gamma)$$

We only have to take the identity for ρ. Otherwise, there is a proposition P which is not a literal. Let $\mathcal{B} = \{\mathcal{B}', P\}$ and $\Gamma = \{\Gamma, P'\}$ with $\Theta\mathcal{B}' =_{\mathcal{E}} \Gamma'$ and $\Theta P =_{\mathcal{E}} P'$.

– If $P = Q_1 \wedge Q_2$, then $P' = Q_1' \wedge Q_2'$ and $\Theta Q_1 = Q_1', \Theta Q_2 = Q_2'$.

  By induction hypothesis

$$\Theta\text{tnf}\mathcal{B}', Q_1, Q_2) =_{\mathcal{E}} \rho\text{tnf}\Gamma', Q_1', Q_2')$$

  i.e. $\Theta\text{tnf}\mathcal{B}') =_{\mathcal{E}} \text{tnf}\Gamma')$.

– If $P =\perp$ then $P' =\perp$. We have $\Theta\text{tnf}\mathcal{B}) = \emptyset = \Theta\text{tnf}\Gamma)$. We take the identity for ρ.

– If $P = \neg \perp$ then $P' = \neg \perp$.

  By induction hypothesis

$$\Theta\text{tnf}\mathcal{B}') =_{\mathcal{E}} \rho tab; (\Gamma')$$

  i.e. $\Theta\text{tnf}\mathcal{B}) =_{\mathcal{E}} \rho\text{tnf}\Gamma)$.

- If $P = \neg\neg Q$ then $P' = \neg\neg Q'$ and $\Theta Q =_{\mathcal{E}} Q'$.

  By induction hypothesis

  $$\Theta\text{tnf}\mathcal{B}', Q) =_{\mathcal{E}} \rho\text{tnf}\Gamma', Q')$$

  i.e. $\text{tnf}\mathcal{B}) =_{\mathcal{E}} \rho\text{tnf}\Gamma)$.

- If $P = Q_1 \vee Q_2$ then $P' = Q'_1 \vee Q'_2$ and $\Theta Q_1 = Q'_1, \Theta Q_2 = Q'_2$.

  By induction hypothesis

  $$\Theta\text{tnf}\mathcal{B}', Q_1) =_{\mathcal{E}} \rho\text{tnf}\Gamma', Q'_1)$$

  and

  $$\Theta\text{tnf}\mathcal{B}', Q_2) =_{\mathcal{E}} \rho'\text{tnf}\Gamma', Q'_2)$$

  Since the domains of ρ and ρ' are disjoint:

  $$\begin{aligned} \Theta\text{tnf}\mathcal{B}) &= \Theta\text{tnf}\mathcal{B}', Q_1)|\Theta\text{tnf}\mathcal{B}', Q_2) \\ &=_{\mathcal{E}} \rho\text{tnf}\Gamma', Q'_1)|\rho'\text{tnf}\Gamma', Q'_2) \\ &= (\rho \cup \rho')\text{tnf}\Gamma) \end{aligned}$$

- If $P = \exists x Q$ then $P' = \exists x Q'$ and $\Theta Q =_{\mathcal{E}} Q'$. Let $y_1, \ldots, y_n$ be the label of P and $z_1, \ldots, z_q$ be the free variables in $\Theta y_1, \ldots, \Theta y_n$. The label of ΘP and P' is $z_1, \ldots, z_q$.

  We have $\text{tnf}\mathcal{B}) = \text{tnf}\mathcal{B}, Q[x := f(y_1, \ldots, y_n)]$ and $\text{tnf}\Gamma) = \text{tnf}\Gamma', Q'[x := g(z_1, \ldots, z_q)])$.

  As $\Theta Q =_{\mathcal{E}} Q'$,

  $$\begin{aligned} \Theta(Q[x := f(y_1, \ldots, y_n)]) &= (\Theta Q)[x := f(\Theta y_1, \ldots, \Theta y_n)] \\ &=_{\mathcal{E}} Q'[x := f(\Theta y_1, \ldots, \Theta y_n)] \\ &= \rho Q'[x := g(z_1, \ldots, z_q)] \end{aligned}$$

  where $\rho = \{(z_1, \ldots, z_q) f(\Theta y_1, \ldots, \Theta y_n)/g\}$. Thus

  $$\Theta(\mathcal{B}', Q[x := f(y_1, \ldots, y_n)]) =_{\mathcal{E}} \rho(\Gamma', Q'[x := g(z_1, \ldots, z_q)])$$

  and by induction hypothesis

  $$\Theta\text{tnf}\mathcal{B}', Q[x := f(y_1, \ldots, y_n)]) =_{\mathcal{E}} \rho'\text{tnf}\rho(\Gamma', Q'[x := g(z_1, \ldots, z_q))])$$

  and by lemma 3

  $$\Theta\text{tnf}\mathcal{B}', Q[x := f(y_1, \ldots, y_n)]) =_{\mathcal{E}} \rho'\rho\text{tnf}\Gamma', Q'[x := g(z_1, \ldots, z_q)])$$

  i.e. $\Theta\text{tnf}\mathcal{B}) =_{\mathcal{E}} \rho'\rho\text{tnf}\Gamma)$

– If $P = \forall x Q$ then $P' = \forall x Q'$ and $\Theta Q =_{\mathcal{E}} Q'$. We proceed by case on the integer n_x for the considered universally quantified formula.

  - If $n_x = 1$ then
$$\Theta \mathrm{tnf}\mathcal{B}', Q) =_{\mathcal{E}} \rho \mathrm{tnf}\Gamma', Q')$$
    i.e. $\Theta \mathrm{tnf}\mathcal{B}) =_{\mathcal{E}} \rho \mathrm{tnf}\Gamma)$.
  - If $n_x > 1$ then, as $\Theta \mathcal{B}' =_{\mathcal{E}} \Gamma'$, $\Theta P =_{\mathcal{E}} P'$ and $\Theta Q =_{\mathcal{E}} Q'$
$$\Theta tab(\mathcal{B}, Q) =_{\mathcal{E}} \rho \mathrm{tnf}\Gamma, Q)$$

– The cases $\neg(Q_1 \vee Q_2)$ and $\neg(Q_1 \Rightarrow Q_2)$ are treated like the case $Q_1 \wedge Q_2$. The cases $\neg(Q_1 \wedge Q_2)$ and $Q_1 \Rightarrow Q_2$ are similar to the case $Q_1 \vee Q_2$. The case $\neg(\forall z; Q)$ is similar to the case $\exists z; Q$ and the case $\neg(\exists z; Q)$ to $\forall z; Q$.

$\Diamond$

PROPOSITION 10. (TaMeD completeness). *Let $\mathcal{U}$ be a constrained tableau, Θ be a $\mathcal{E}$-unifier of the constraints of $\mathcal{U}$, and $\mathcal{T}$ a non-constrained tableau such that*
$$\Theta \mathcal{U} =_{\mathcal{E}} \mathcal{T}$$
and
$$\mathcal{T} \overset{IcT}{\hookrightarrow} \odot$$
then
$$\mathcal{U} \overset{T}{\mapsto} \odot[C]$$
where C is a unifiable set of constraints .

Proof: This proof is done by induction on the structure of the IC-TaMeD proof of $\mathcal{T}$. For the rest of the proof, let also $\mathcal{T} = \mathcal{B}_1 | \ldots | \mathcal{B}_n$ and $\mathcal{U} = \Gamma_1 | \ldots | \Gamma_p$.

If the IC-TaMeD derivation is empty, then $\mathcal{T}$ is formed of only closed branches and therefore $\mathcal{U} \overset{T}{\mapsto} \odot$ where C is a set on constraints and Θ a $\mathcal{E}$-unifier of C.

Otherwise the derivation starts by producing a tableau $\mathcal{T}'$ from $\mathcal{T}$ using the IC-TaMeD rules and $\mathcal{T}'$ has a smaller derivation to the closed form. We detail the four different rules which can be used to produce $\mathcal{T}'$

– **Case Instantiation:**
Let $\mathcal{T}' = \mathcal{T}[x \mapsto t]$. By hypothesis, $\Theta\mathcal{U} =_{\mathcal{E}} \mathcal{T}$. We now have:

$$(\Theta\mathcal{U})[x \mapsto t] =_{\mathcal{E}} \mathcal{T}[x \mapsto t] = \mathcal{T}'$$

Let $\Theta' = [x \mapsto t]\dot{\Theta}$, we get $\Theta' U =_{\mathcal{E}} \mathcal{T}'$ and Θ' is a unifier of the constraints of the tableau $\mathcal{U}$.

As $\mathcal{T}' \overset{\text{IcT}}{\hookrightarrow} \odot$ and $\Theta'\mathcal{U} =_{\mathcal{E}} \mathcal{T}'$, then we get by induction hypothesis $U \overset{\mathcal{T}}{\mapsto} \odot$.

– **Case Conversion:**
The case is obvious as:

$$\Theta\mathcal{U} =_{\mathcal{E}} \mathcal{T} =_{\mathcal{E}} \mathcal{T}'$$

The induction hypothesis is used on $\Theta\mathcal{U} =_{\mathcal{E}} \mathcal{T}'$ and $\mathcal{T}' \overset{\text{IcT}}{\hookrightarrow} \odot$ to yield $U \overset{\mathcal{T}}{\mapsto} \odot$.

– **Case Identical Branch Closure:**
$\mathcal{T}$ has a branch, say $\mathcal{B}_1$, containing a literal P and its negation$\neg P$. $\Theta\mathcal{U} =_{\mathcal{E}} \mathcal{T}$ therefore there is a branch,say Γ_1, in $\mathcal{U}$ containing two literals P' and $\neg Q'$ such that $\Theta P' =_{\mathcal{E}} P =_{\mathcal{E}} \Theta Q'$.
Let $\mathcal{U}' = \mathcal{U}\backslash\{\Gamma_1\}$ then obviously $\Theta\mathcal{U}' = \mathcal{T}'$. By induction hypothesis $\mathcal{T}' \overset{\text{IcT}}{\hookrightarrow} \odot$ therefore $\mathcal{U}' \overset{\mathcal{T}}{\mapsto} \odot[C]$ and Θ is a unifier of C.
Applying **Extended Branch Closure** to P' and $\neg Q'$ in $\mathcal{U}$ exactly yields the constrained tableau $\mathcal{U}'$. Hence, $\mathcal{U} \overset{\mathcal{T}}{\mapsto}_{\text{EBC}} \mathcal{U}' \overset{\mathcal{T}}{\mapsto} \odot \cdot [C]$

– If the rule is **Reduction**, then there is in $\mathcal{T}$ a branch, say $\mathcal{B}_1 = (\mathcal{B}, P)$, containing a literal P such that $P \rightarrow_{\mathcal{R}} Q$ and $\Gamma'_1 \mid \ldots \mid \Gamma'_p = \text{tnf}\mathcal{B}, Q)$. Moreover, there is a constrained tableau $\mathcal{U}[C_1]$ containing a branch, say $\Gamma_1 = (\Gamma, P')$ such that $\Theta\Gamma =_{\mathcal{E}} \mathcal{B}$ and $\Theta P' =_{\mathcal{E}} P$.

As $\Theta P' =_{\mathcal{E}} P$, $P \rightarrow_{\mathcal{R}} Q$ and $\mathcal{R}$ applies only to atomic propositions and P' is a literal, we have $\Theta P' \rightarrow_{\mathcal{R}\mathcal{E}} Q$. Hence, the proposition $\Theta P'$ contains an occurrence ω such that $\Theta P'_{|\omega} =_{\mathcal{E}} \sigma l$ and $\Theta P'[\sigma r]_{\omega} = Q$ for some substitution σ and rule $l \rightarrow r$. We also have $(\Theta P')_{|\omega} = \Theta(P'_{|\omega})$. Let $Q' = P'[r]_{|\omega}$, $C' =$

$C_1 \cup \{P'_{|\omega} =^?_{\mathcal{E}} l\}$ and $\Theta' = \Theta \cup \sigma$. As the domain of the substitution σ contains only fresh variables, Θ' is a unifier of C', $\Theta'\Gamma = \Theta\Gamma =_{\mathcal{E}} \mathcal{B}$ and

$$\Theta' Q' = \Theta'(P'[r]_\omega) = \Theta'(P')[\Theta' r]_\omega = \Theta P'[\sigma r]_\omega = Q$$

$\Theta'(\Gamma, Q') =_{\mathcal{E}} \mathcal{B}, Q$ and, since the substitution σ only affects the $\{\Gamma, Q'\}$ branch:

$$\Theta'(\Gamma, Q'|\Gamma_2 \mid \ldots \mid \Gamma_p) =_{\mathcal{E}} (\mathcal{B}, Q|\mathcal{B}_2 \mid \ldots \mid \mathcal{B}_n)$$

. Let $\mathcal{U}' = \text{tnf}\Gamma, Q'|\Gamma_2| \ldots |\Gamma_p)$. By lemma 11 there is a transformation of function symbols such that

$$\Theta'\mathcal{U}' =_{\mathcal{E}} \rho\text{tnf}\mathcal{B}, Q|\mathcal{B}_2| \ldots |\mathcal{B}_n) = \rho\mathcal{T}'$$

.
As $\mathcal{T}' \overset{\text{IcT}}{\hookrightarrow} \odot$, $\rho\mathcal{T}' \overset{\text{IcT}}{\hookrightarrow} \odot$ by lemma 4 and the derivation have the same length. Hence, the induction hypothesis can be applied on $\Theta\mathcal{U}'$ and $\rho\mathcal{T}$ yielding $\mathcal{U}'[C'] \overset{T}{\mapsto} \odot[C]$ where C is a unifiable set of constraints.
The **Extended Narrowing** rule applied to $\mathcal{U}$ (precisely to its branch $\mathcal{B}_1$) leads to $\mathcal{U}'$ in several steps. Hence, $\mathcal{U} \overset{T}{\mapsto}_{ExtN} \mathcal{U}' \overset{T}{\mapsto} \odot[C]$ where C is a unifiable set of constraints.

$\Diamond$

6. Conclusion

After introducing the sequent calculus modulo of (Dowek et al., 2003), we have presented a tableau-based proof search method for deduction modulo. Furthermore we have shown that TaMeD is indeed sound and complete with respect to this sequent calculus modulo. TaMeD is built as a simple extension of the usual first-order free-variable tableau method.

The integration of rewrite rules dedicated to computational operations in sequent rules dedicated to deduction permits to underline the essential steps of a given proofs: those where deduction, non-deterministic search, had to be used. Moreover powerful and expressive theories have been expressed as deduction modulo.

Small congruences can be defined such as in the propositional rewriting example given in the introduction. However arithmetic in (Dowek and Werner, 2005) or, more significantly, higher-order logic (both extensional and intentional) in (Dowek et al., 1999) have been expressed as deduction modulo. The latter work gives in particular a presentation of intentional classical higher-order logic as a first-order logic modulo. This work together with this paper yields almost for free a framework for a higher-order tableau method, which would be rather different from the extensional higher-order tableau method described in (Kohlhase, 1998).

We have also tried to underline the differences with the ENAR method, which principally come from the fact that free variables introduced by our tnf-calculus can not be locally dealt with and therefore have to be present globally. This might be a drawback when compared to resolution. However, the tableau closure rule of TaMeD allows binary closure which ENAR does not.

All in all, the calculus presented here is quite basic and separate the process into two steps: tnf and TaMeD . Yet, one could want to introduce some restrictions or extensions. What easily comes to mind would be to allow non-literal branch closure, which would ensure faster closure in many cases (that should not change the soundness and completeness of the method). Furthermore, the skolemization process defined through the use labels could also be refined perhaps by using the framework of (Cantone and Asmundo, 2005), thus improving the whole method. Finally, allowing non-atomic propositional rewrite rules in specific cases seems possible as in (Deplagne, 2002) where first-order classical sequents are presented as a theory modulo.

Strong assumptions have been made in our proofs, for example when relying on the confluence property of the considered $\mathcal{RE}$ rewrite system and cut elimination in the sequent calculus modulo, which could possibly be relaxed using for example the work of (Hermant, 2005).

The method used to solve the constraints has also been omitted. Unification-related problems are indeed not the topic of this paper. It is however expected that the rigidity of free variables introduced by the tableau method will once again be a major problem when dealing at least with equational theories (see for example (Beckert, 1998)), since rigid $\mathcal{E}$-unification has been proved undecidable (Degtyarev and Voronkov, 1998).

Further research could eventually be made to get a model-based completeness proof for TaMeD as is done in (Stuber, 2001) for the resolution-based ENAR. However, the next major step should primarily be to implement TaMeD on top of a first-order tableau prover.

References

Beckert, B.: 1991, 'Konzeption und Implementierung von Gleichheit für einen tableau-basierten Theorembeweiser'. IKBS report 208, Science Center, Institute for Knowledge Based Systems, IBM Germany.

Beckert, B.: 1994, 'Adding Equality to Semantic Tableaux'. In: K. Broda, M. D'Agostino, R. Goré, R. Johnson, and S. Reeves (eds.): *Proceedings, 3rd Workshop on Theorem Proving with Analytic Tableaux and Related Methods, Abingdon.* Imperial College, London, TR-94/5, pp. 29–41.

Beckert, B.: 1998, 'Rigid E-unification'. In: W. Bibel and P. H. Schmidt (eds.): *Automated Deduction: A Basis for Applications. Volume I, Foundations: Calculi and Methods.* Dordrecht: Kluwer Academic Publishers.

Beckert, B. and R. Hähnle: 1992, 'An Improved Method for Adding Equality to Free Variable Semantic Tableaux'. In: D. Kapur (ed.): *Proceedings, 11th International Conference on Automated Deduction (CADE), Saratoga Springs, NY.* pp. 507–521.

Cantone, D. and M. N. Asmundo: 2005, 'A Sound Framework for delta-Rule Variants in Free Variable Semantic Tableaux'. In: *FTP*.

Degtyarev, A. and A. Voronkov: 1998, 'What You Always Wanted to Know about Rigid E-Unification'. *J. Autom. Reason.* **20**(1-2), 47–80.

Degtyarev, A. and A. Voronkov: 2001, *Equality Reasoning in Sequent-based Calculi*, Chapt. 10. Elsevier Science Publishers B.V.

Deplagne, E.: 2002, 'Systme de preuve modulo rcurrence'. Ph.D. thesis, Universit Henri Poincar - Nancy 1.

Dowek, G., T. Hardin, and C. Kirchner: 1999, 'HOL-$\lambda\sigma$: An Intentional First-Order Expression of Higher-Order Logic'. In: *RTA*. pp. 317–331.

Dowek, G., T. Hardin, and C. Kirchner: 2003, 'Theorem Proving Modulo'. *Journal of Automated Reasoning* (31), 33–72.

Dowek, G. and B. Werner: 2005, 'Arithmetic as a Theory Modulo'. In: J. Giesel (ed.): *Term rewriting and applications.*

Fitting, M.: 1996, *First-Order Logic and Automated Theorem Proving.* Springer-Verlag, 2nd edition.

Goubault-Larrecq, J. and I. Mackie: 1997, *Proof Theory and Automated Deduction*, Applied Logic Series. Kluwer Academic Publishers.

Hermant, O.: 2005, 'Méthodes Sémantiques en Déduction Modulo'. Ph.D. thesis, Université Paris 7 - Denis Diderot.

Kohlhase, M.: 1998, 'Higher-Order Automated Theorem Proving'. In: W. Bibel and P. H. Schmidt (eds.): *Automated Deduction: A Basis for Appli-*

cations. Volume I, Foundations: Calculi and Methods. Dordrecht: Kluwer Academic Publishers.

Letz, R.: 1998, 'First-Order Tableau Methods'. In: M. d'Agostino, D. M. Gabbay, R. Hähnle, and J. Posegga (eds.): *Handbook of Tableau Methods.* Dordrecht: Kluwer Academic Publishers.

Smullyan, R.: 1968, *First Order Logic.* Springer.

Stuber, J.: 2001, 'A Model-Based Completeness Proof of Extended Narrowing and Resolution'. *Lecture Notes in Computer Science* **2083**, 195+.