Using Mixed Universal and Rigid E-Unification to Handle Equality in Universal Formula Semantic Tableaux

— Extended Abstract —

Bernhard Beckert

University of Karlsruhe
Institute for Logic, Complexity und Deduction Systems
76128 Karlsruhe, Germany
beckert@ira.uka.de

Abstract. In this paper we describe how a combination of the classical "universal" E-unification and "rigid" E-unification, called "mixed" E-unification, can be used to efficiently handle equality in universal formula semantic tableaux, that are an extension of free variable tableaux.

1 Introduction

In this paper we describe how "mixed" E-unification [2], a combination of the classical "universal" E-unification and "rigid" E-unification [8], can be used to efficiently handle equality in universal formula semantic tableaux [4], that are an extension of free variable tableaux [7].

There are two techniques for handling equality in semantic tableaux: The first and more straightforward method is to define additional tableau rules for expanding branches by all the formulae valid in the canonical models[1] they (partially) define; then very simple additional closure rules can be used [11, 12, 7]. The second possibility is to use a more complicated notion of closed tableaux: E-*unification* is used to decide whether a tableau branch is unsatisfiable in canonical models and, therefore, closed. Then, no additional expansion rules are needed.

The common problem of all the methods for handling equality, that are based on additional tableau expansion rules, is that there are virtually no restrictions on the "application" of equalities. This leads to a very large search space; even very simple problems cannot be solved in reasonable time. Unfortunately, it is difficult to transform completion-based methods into (sufficiently) simple tableau expansion rules.[2]

Contrary to that, arbitrary algorithms can be used, if the handling of equality is reduced to solving E-unification problems. In [4] it has been shown that methods based on E-unification are much more efficient than that based on additional rules—even if the comparatively inefficient algorithm from [4] is used to solve E-unification problems.

In the next section, we present the different versions of E-unification that are important for adding equality to semantic tableaux. Section 3 gives a short introduction into universal formula tableaux. In Section 4 we describe the E-unification problems that are extracted from tableaux and have to be solved; and, finally, in Section 5 different methods for solving these E-unification problems and their efficient implementation are discussed.

2 Mixed E-Unification

The intention of defining different versions of E-unification is to allow equalities to be used differently in a proof: in the universal case the equalities can be "applied" several times with different instantiations for the variables they contain; in the rigid case they can be "applied" more than once but with only one instantiation for each variable they contain; in the mixed case there are both types of variables. To distinguish the different types of variables syntactically, equalities can be explicitly quantified:

Definition 1. A *mixed E-unification problem*

$$\langle E, s, t\rangle$$

consists of a finite set E of equalities of the form

$$(\forall x_1)\cdots(\forall x_n)(l \approx r)$$

and terms s and t.[3]

[1] A model $\mathcal{M} = \langle \mathcal{D}, \mathcal{I}\rangle$ (with domain $\mathcal{D}$ and interpretation $\mathcal{I}$) is called *normal* iff $\mathcal{I}(\approx)$ is the identity relation on $\mathcal{D}$ (the binary predicate symbol $\approx$ denotes equality such that no confusion with the meta-level equality predicate $=$ can arise). A model is called *canonical* iff, moreover, for every $d \in \mathcal{D}$ there is a term t such that $\mathcal{I}(t) = d$.

[2] R. J. Browne [6] describes a completion-based method for handling equality, that uses additional expansion rules. It is, however, only applicable to the ground version of tableaux and cannot be extended to free variable tableaux.

[3] Without making a real restriction, we require the sets of bound and free variables in the problem to be disjoint.

A substitution σ is a *solution* to the problem, iff[4]

$$E\sigma \models^{\circ} (s\sigma \approx t\sigma) \enspace .$$

The major differences between this definition and that generally given in the (extensive) literature on (universal) E-unification are:

1. The equalities in E are *explicitly* quantified (instead of considering all the variables in E to be *implicitly* universally quantified).
2. Instead of the "normal" notion of logical consequence, the strong consequence relation is used, i.e., free variables in $E\sigma$ are "held rigid".
3. The substitution σ is applied not only to the terms s und t but as well to the set E.

Example 1. All substitutions are solutions to the purely universal problem

$$\langle \{(\forall x)(f(x) \approx x)\},\ g(f(a), f(b)),\ g(a, b)\rangle \enspace .$$

The (very similar) purely rigid problem

$$\langle \{(f(x) \approx x)\},\ g(f(a), f(b)),\ g(a, b)\rangle$$

has no solution.
$\{y/b\}$ is a solution to the mixed problem

$$\langle \{(\forall x)(f(x, y) \approx f(y, x))\},\ f(a, b),\ f(b, a)\rangle \enspace ;$$

since the variable x is quantified, it does not have to be instantiated by the unifier.

For handling equality in semantic tableaux, several E-unification problems have to be solved simultaneously (one for each branch):

Definition 2. A finite set

$$\{\langle E_1, s_1, t_1\rangle, \ldots, \langle E_n, s_n, t_n\rangle\} \qquad (n \geq 1)$$

of mixed E-unification problems is called *simultaneous* E-unification problem.

A substitution σ is a solution to the simultaneous problem iff it is a solution to every component $\langle E_k, s_k, t_k\rangle$ $(1 \leq k \leq n)$.

Since purely universal E-unification is already undecidable, (simultaneous) *mixed* E-unification is—in general—undecidable as well. Is is, however, possible to enumerate a complete set of most general unifiers. (Simultaneous) *purely rigid* E-unification is decidable [8, 10].[5]

[4] $\models^{\circ}$ denotes the strong consequence relation, i.e., $F \models^{\circ} G$ iff for all interpretations $\mathcal{I}$ and for all variable assignments β: if *val*$_{\mathcal{I},\beta}(F) =$ *true* then *val*$_{\mathcal{I},\beta}(G) =$ *true*.

[5] Purely rigid E-unification is NP-complete [8]; simultaneous purely rigid E-unification is DEXPTIME-complete [10].

3 Universal Formula Tableaux

We use the signed version of semantic tableaux, i.e., the formulae in tableaux are prefixed with one of the signs T (true) and F (false). There is no restriction on where equalities can occur in formulae.

There is a tableau rule for each combination of sign and logical connective (resp. quantifier); thus, to every signed formula that is not a literal exactly one rule can be applied. We do not list all the rules but only the schemata: α-rules (conjunctive type rules), β-rules (disjunctive), γ-rules (universally quantified), and δ-rules (existentially quantified):[6]

$$\frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad\qquad \frac{\beta}{\beta_1 \mid \beta_2} \qquad\qquad \frac{\gamma}{\gamma_1(y)}$$

y is a free variable.

$$\frac{\delta}{\delta_1(f(x_1, \ldots, x_n))}$$

f is a new Skolem function symbol, and
$x_1, \ldots, x_n$ are the free variables in δ.

Using free variable quantifier rules [7, 5] is crucial for efficient implementation—even more if equality has to be handled. When γ-rules are applied, a new free variable is substituted for the quantified variable, instead of replacing it by a ground term, that has to be "guessed" (as in the ground version of semantic tableaux [13]). Free variables can later be instantiated "on demand", when a tableau branch is closed (with or without using equality).

To prove a formula G to be a tautology, we apply the above rules starting from the initial tableau that consists of the single formula $\mathsf{F}\,G$. A proof is found, if all branches of the constructed tableau are closed simultaneously. We identify a branch with the set of the formulae it contains.

Free variable semantic tableaux can be further improved by using the concept of universal formulae [4]: Often, γ-formulae—in particular equalities—have to be used multiply in a tableau proof, with different instantiations for the free variables they contain. A typical example is the associativity axiom from group theory. Usually, it has to be applied several times to prove even very simple theorems from group theory. Therefore, in semantic tableaux the γ-rule has to be applied repeatedly to generate several instances of the axiom each with different free variables. This, however, enlarges the search space for a proof.

[6] For example, if $\alpha = \mathsf{T}\,(F \wedge G)$ then $\alpha_1 = \mathsf{T}\,F$ and $\alpha_2 = \mathsf{T}\,G$; if $\beta = \mathsf{F}\,(F \wedge G)$ then $\beta_1 = \mathsf{F}\,F$ and $\beta_2 = \mathsf{F}\,G$; if $\gamma = \mathsf{T}\,(\forall x)F(x)$ then $\gamma_1(t) = \mathsf{T}\,F(t)$; if $\delta = \mathsf{F}\,(\forall x)F(x)$ then $\delta_1(t) = \mathsf{F}\,F(t)$.

This problem can at least partly be avoided by recognizing formulae (including equalities) that are "universal", i.e. that can be used multiply in a tableau proof with different substitutions for the variables they contain (without affecting soundness):

Definition 3. Let ϕ be a signed formula on some tableau branch B and F_ϕ the "unsigned version" of ϕ, i.e., if $\phi = \mathsf{T}\ G$ for some G then $F_\phi = G$, else if $\phi = \mathsf{F}\ G$ then $F_\phi = \neg G$.

ϕ is *universal* with respect to the variable x iff:

$$B \models^{\circ} (\forall x) F_\phi \ .$$

A *method Υ for recognizing universal formulae* assigns to a tableau branch B and a signed formula ϕ a set $\Upsilon(B, \phi)$ of variables such that:
if

$$x \in \Upsilon(B, \phi)$$

then

1. $\phi \in B$,
2. ϕ is universal w.r.t. x.

An important class of universal formulae can be recognized easily (and the method is easy to implement):

Example 2. Υ_1 is a method for recognizing universal formulae where $\Upsilon_1(B, \phi)$ contains exactly the variables x such that the formula $\phi \in B$ has been added to B

1. by applying a γ-rule, and x is the free variable that has been introduced; or
2. by applying an α-, δ- or γ-rule to a formula ϕ' where $x \in \Upsilon_1(B, \phi')$, i.e., ϕ' is universal w.r.t. x.

A formula $\phi(x)$ is recognized as being universal w.r.t. x by this method, if new instances $\phi(x')$, $\phi(x'')$, ... can be added to the branch without affecting other branches or generating new ones.

A free variable tableau T (without universal formulae) is closed if there is a single substitution σ such that each branch of $T\sigma$ contains complementary formulae. Once formulae are recognized as being universal, this knowledge can be taken advantage of to make it easier to find such a substitution σ: instantiations of variables w.r.t. which the formulae used to close a branch are universal are not taken into consideration. Soundness is not affected if this notion of closed tableau is used [4]; completeness is not affected anyway.

The following is a formal definition of closed universal formula tableaux without equality; in which way this definition has to be changed to handle equality is described in the next section.

Definition 4. Let Υ be a method for recognizing universal formulae. A free variable tableau T with branches $B_1, \ldots, B_k$ is *closed* iff there are

1. a grounding substitution σ, and
2. for $1 \leq i \leq k$
   (a) formulae $\phi_i, \psi_i \in B_i$,
   (b) grounding substitutions σ_i,

such that

1. $\phi_i\sigma_i$ and $\psi_i\sigma_i$ are complementary[7];
2. σ_i differs from σ only on the set U of variables with respect to which both ϕ_i and ψ_i are universal, i.e.[8]

$$\sigma_{i|(\mathrm{Var}\setminus U)} = \sigma_{|(\mathrm{Var}\setminus U)} \ ,$$

   where

$$U = \Upsilon(B_i, \phi_i) \cap \Upsilon(B_i, \psi_i) \ .$$

4 Extracting E-Unification Problems from Tableaux

The equality theory defined by a tableau branch B consists of the equalities on B; they are (explicitly) quantified w.r.t. to the variables w.r.t. which they can be recognized as being universal:

Definition 5. Let B be a tableau branch and Υ a method for recognizing universal formulae (Definition 3). Then the *set $E(B)$ of equalities* consists of the equalities

$$(\forall x_1) \cdots (\forall x_n)(s \approx t)$$

such that

1. $\mathsf{T}\ (s \approx t)$ is formula on B,
2. $\{x_1, \ldots, x_n\} = \Upsilon(B, \mathsf{T}\ (s \approx t))$.

There are unification problems for each inequality on a branch B and each pair of atoms that potentially close B, i.e., atoms with the same predicate sign and complementary truth value signs:

Definition 6. Let B be a tableau branch and Υ a method for recognizing universal formulae. Then the *set $\mathcal{P}(B)$ of unification problems* consists exactly of the sets of term pairs:

$$\{\langle s_1\nu, t_1\nu\rangle, \ldots, \langle s_n\nu, t_n\nu\rangle\}$$

[7] Signed formulae are called complementary iff they are of the form $\mathsf{T}\ G$ and $\mathsf{F}\ G$

[8] $\sigma_{|V}$ denotes the restriction of a substitution σ to a set V of variables.

for each pair

$$\mathsf{T}\ P(s_1,\ldots,s_n), \mathsf{F}\ P(t_1,\ldots,t_n) \in B$$

of (potentially closing) atoms such that $P \neq \approx$, and

$$\{\langle s\nu, t\nu\rangle\}$$

for each inequality

$$\mathsf{F}\ (s \approx t) \in B\ .$$

The substitution $\nu = \{x_1/y_1, \ldots, x_m/y_m\}$ renames all the variables $x_1, \ldots, x_m$ in

$$\Upsilon(B, \mathsf{T}\ P(s_1,\ldots,s_n)) \cap \Upsilon(B, \mathsf{F}\ P(t_1,\ldots,t_n))$$

(resp. in $\Upsilon(A, \mathsf{F}\ (s \approx t))$); $y_1, \ldots, y_m$ are new variables.

If one of the problems in the set $\mathcal{P}(B)$ of unification problems of a branch B has a solution σ (w.r.t. the equalities $E(B)$), $B\sigma$ is unsatisfiable in canonical models; therefore the branch B is closed under the substitution σ. The pair of potentially closing atoms corresponding to the solved unification problem has been proven to actually be complementary; or the corresponding inequality has been proven to be inconsistent (provided the unifier is applied to the tableau).

The following is a formal definition of the simultaneous mixed E-unification problems that have to be solved to close a tableau:

Definition 7. A universal formula tableau T with branches $B_1, \ldots, B_k$ is *closed* iff in the sets of unification problems $\mathcal{P}(B_i)$ there are elements

$$\{\langle s_{i1}, t_{i1}\rangle, \ldots, \langle s_{in_i}, t_{in_i}\rangle\} \in \mathcal{P}(B_i)$$

$(1 \leq i \leq k)$ such that there is a solution to the simultaneous mixed E-unification problem

$$\begin{array}{lll} \{\ \langle E(B_1), s_{11}, t_{11}\rangle, & \ldots, & \langle E(B_1), s_{1n_1}, t_{1n_1}\rangle, \\ \quad\vdots & \vdots & \quad\vdots \\ \ \ \langle E(B_k), s_{k1}, t_{k1}\rangle, & \ldots, & \langle E(B_k), s_{kn_k}, t_{kn_k}\rangle\ \} \end{array}$$

(see Definitions 5 and 6).

Using algorithms based on solving *simultaneous* rigid E-unification problems, all branches of a tableau are closed simultaneously. Another possibility, that is easier to implement, is to close the branches one after the other; the first substitution found to close a branch B_i is applied to the whole tableau. If, later on, it is not possible to close a branch B_j $(j > i)$, backtracking is initiated to compute further closing substitutions for B_i. This method leads to a correct and complete calculus, provided the search for further substitutions closing a branch is fair.

(1) $\mathsf{T}\ (\forall x)((g(x) \approx f(x)) \vee \neg(x \approx a))$

(2) $\mathsf{T}\ (\forall x)(g(f(x)) \approx x)$

(3) $\mathsf{T}\ (b \approx c)$

(4) $\mathsf{T}\ P(g(g(a)), b)$

(5) $\mathsf{T}\ \neg P(a, c)$

(6) $\mathsf{F}\ P(a, c)$

(7) $\mathsf{T}\ (g(f(x_1)) \approx x_1)$

(8) $\mathsf{T}\ ((g(x_2) \approx f(x_2)) \vee \neg(x_2 \approx a))$

(9) $\mathsf{T}\ (g(x_2) \approx f(x_2))$ | (10) $\mathsf{T}\ \neg(x_2 \approx a)$

(11) $\mathsf{F}\ (x_2 \approx a)$

Fig. 1. A free variable tableau for the given formulae (1) to (5). By applying the standard free variable tableau rules, formula (6) is derived from (5), (7) from (2), (8) from (1), (9) and (10) from (8), and (11) from (10). Formula (7) is recognized as being universal w.r.t. x by the method from Example 2.

Example 3. As an example we use the tableau from Figure 1. Its left branch is denoted by B_1 and its right branch by B_2. If the method for recognizing universal formulae from Example 2 is used, $E(B_2)$ contains the equalities

$$b \approx c \text{ and } (\forall x)(g(f(x)) \approx x)\ .$$

$E(B_1)$ contains in addition the equality

$$g(x_2) \approx f(x_2)\ .$$

Both $\mathcal{P}(B_1)$ and $\mathcal{P}(B_2)$ contain the set

$$\{\langle g(g(a)), a\rangle, \langle b, c\rangle\}\ .$$

$\mathcal{P}(B_2)$ contains in addition the set

$$\{\langle x_2, a\rangle\}\ .$$

The tableau is closed (Definition 7), because the substitution $\sigma = \{x_2/a\}$ is a solution to the simultaneous mixed E-unification problem

$$\begin{array}{lll} \{\langle E(B_1), & g(g(a)), & a\rangle, \\ \ \langle E(B_1), & b, & c\rangle, \\ \ \langle E(B_2), & x_2, & a\rangle\}\ . \end{array}$$

5 Conclusion

The handling of equality by solving E-unification problems can be combined with virtually all refinements of tableau, such as regularity, links, lemma generation, etc.

In addition, arbitrary algorithms can be used to solve the mixed E-unification problems that are extracted from tableaux. One possibility to do this, is to compute the equivalence classes of the terms to be unified (w.r.t. to the relation defined by the equalities on the branch) [4].

It is, however, much better to use completion-based methods. Unfortunately, the Unfailing Knuth-Bendix-Algorithm with narrowing, that is generally considered to be the best algorithm for universal E-unification, cannot be used to solve rigid or mixed problems. Completion-based methods for rigid E-unification have been described in [8, 9]; these, however, are non-deterministic and unsuited for implementation, since the "guess" that is part of the algorithm is highly complex.

Deterministic completion-based methods have been introduced recently, both for purely rigid E-unification [1] and mixed E-unification [2].[9] Besides being completion-based, there are several reasons why these methods are well suited for adding equality to free variable semantic tableaux: Firstly, the terms to be unified do not become part of the completion (in contrary to the method in [8]); this is important because the E-unification problems in Definition 7 that share the same set of equalities can, thus, be solved using a single completion. Secondly, simultaneous E-unification problems are solved by searching for common specializations of solutions to its components; this is of advantage, because the different E-unification problems consist of the same components.

If a completion-based method is used to solve the E-unification problems extracted from a tableau, it is advantageous to combine the completion process and the expansion of the tableau. Thus, if a β-rule is applied, the (partial) completion that has been computed up to that point can be shared by the new subbranches and has only to be computed once. The question, whether handling equality using E-unification is really superior to other methods, will remain open until this improvement is added to one of the existing implementations (or a new one).

[9] The method described in [2] has been implemented as part of the tableau-based theorem prover $_3T^AP$ [3]. The implementation is written in Prolog. Besides the possibility to prove theorems from predicate logic with equality, the E-unification module can be used "stand alone" to solve simultaneous mixed E-unification problems. Upon request, the source code is available from the author.

References

1. G. Becher and U. Petermann. Rigid E-unification by completion and rigid paramodulation. Report 93-22, LAIAC, University of Caen, 1993.
2. B. Beckert. A completion based method for mixed universal and rigid E-unification. In *Proceedings, 12th International Conference on Automated Deduction (CADE), Nancy*, LNCS. Springer, 1994.
3. B. Beckert, S. Gerberding, R. Hähnle, and W. Kernig. The tableau-based theorem prover $_3T^AP$ for multiple-valued logics. In *Proceedings, 11th International Conference on Automated Deduction (CADE), Saratoga Springs, NY*, LNCS 607. Springer, 1992.
4. B. Beckert and R. Hähnle. An improved method for adding equality to free variable semantic tableaux. In D. Kapur, editor, *Proceedings, 11th International Conference on Automated Deduction (CADE), Saratoga Springs, NY*, LNCS 607, pages 507–521. Springer, 1992.
5. B. Beckert, R. Hähnle, and P. H. Schmitt. The even more liberalized δ-rule in free variable semantic tableaux. In G. Gottlob, A. Leitsch, and D. Mundici, editors, *Proceedings, 3rd Kurt Gödel Colloquium (KGC), Brno, Czech Republic*, LNCS 713, pages 108–119. Springer, August 1993.
6. R. J. Browne. Ground term rewriting in semantic tableaux systems for first-order logic with equality. Technical Report UMIACS-TR-88-44, College Park, MD, 1988.
7. M. Fitting. *First-Order Logic and Automated Theorem Proving*. Springer, 1990.
8. J. Gallier, P. Narendran, S. Raatz, and W. Snyder. Theorem proving using equational matings and rigid E-unification. *Journal of the ACM*, 39(2):377–429, April 1992.
9. J. Goubault. A rule-based algorithm for rigid E-unification. In G. Gottlob, A. Leitsch, and D. Mundici, editors, *Proceedings, 3rd Kurt Gödel Colloquium (KGC), Brno, Czech Republic*, LNCS 713, pages 202–210. Springer, August 1993.
10. J. Goubault. Personal communication. May 1994.
11. R. C. Jeffrey. *Formal Logic: Its Scope and Limits*. McGraw Hill, 1967.
12. S. Reeves. Adding equality to semantic tableau. *Journal of Automated Reasoning*, 3:225–246, 1987.
13. R. Smullyan. *First-Order Logic*. Springer, 1968.