UPMAIL Technical Report No. 109
June 1, 1995

Reduction of Second-Order Unification to Simultaneous Rigid E-Unification

Anatoli Degtyarev
Andrei Voronkov*

Computing Science Department
Uppsala University
Box 311, S-751 05 Uppsala,
Sweden

email {anatoli,voronkov}@csd.uu.se

*Supported by a TFR grant

Abstract

The simultaneous rigid E-unification problem is used in automated reasoning with equality. In our previous paper we proved the undecidability of this problem by reduction of monadic semi-unification. Here we give a simpler and more intuitive proof of the undecidability by reduction of second-order unification.

1 Introduction

Simultaneous rigid E-unification plays a crucial role in extending to first order languages with equality automatic proof methods based on sequent calculi, such as semantic tableaux [Fitting 88], the connection method [Bibel 82] (also known as the mating method [Andrews 81]), model elimination [Loveland 68] and a dozen other procedures. Simultaneous rigid E-unification was defined in [GaRaSn 87] (see also [GNRS 92]) and witnessed several faulty proofs of the decidability.

The undecidability of simultaneous rigid E-unification was proven in [DeVo 95b] by reduction of monadic semi-unification [Baaz 93]. Here we give a more elementary and more intuitive proof using reduction of second-order unification. For a short survey of other known results on simultaneous rigid E-unification see [DeMaVo 95, DeVo 95b].

2 Second-order unification

We define second-order unification mainly following [Goldfarb 81].

Definition 1 *A* second-order language $\mathcal{L}^2$ *is a triple* $\langle \Sigma, \mathcal{X}, \mathcal{W} \rangle$ *of disjoint sets, where* Σ *is finite and* $\mathcal{X}, \mathcal{W}$ *are countable. For any such second-order language* Σ *is the set of* function symbols *of* $\mathcal{L}^2$, $\mathcal{X}$ *is the set of* free variables *of* $\mathcal{L}^2$ *and* $\mathcal{W}$ *is the set of* bound variables *of* $\mathcal{L}^2$. *A* variable *of a second-order language* $\mathcal{L}^2$ *is either a free variable or a bound variable of* $\mathcal{L}^2$. *For any such second-order language we assume that the sets* Σ *and* $\mathcal{X}$ *be the unions of disjoint sets* Σ_n *and* $\mathcal{X}_n$ *of function symbols and variables of* arity n, $n \in \{0, 1, \ldots\}$. *Elements of* $\mathcal{W}$ *have arity* 0. *Elements of* Σ_0 *are called* constants.

Unlike [Goldfarb 81] we do not introduce individual variables, their role is played by free variables of arity 0.

For the rest of this section we assume that $\mathcal{L}^2$ stands for a fixed second-order language $\langle \Sigma, \mathcal{X}, \mathcal{W} \rangle$.

Definition 2 *The set* $T(\mathcal{L}^2)$ *of* second order terms *of the language* $\mathcal{L}^2$ *is defined inductively as follows: for every* $f \in \Sigma \bigcup \mathcal{X} \bigcup \mathcal{W}$ *of arity* m *and any* $t_1, \ldots, t_m \in T(\mathcal{L}^2)$, $m \geq 0$, *we have* $f(t_1, \ldots, t_m) \in T(\mathcal{L}^2)$. *A second-order term in* $T(\mathcal{L}^2)$ *is called* ground *iff it uses no variables.*

As usual, for any symbol f of arity 0 we write f instead of $f()$. The equality predicate will be denoted by $=$, whereas $\approx$ means the literal identity of two expressions. We assume $=$ be symmetric, i.e. we do not distinguish $s = t$ from $t = s$. The notation $\rightleftharpoons$ stands for "equal by definition". The set of all variables occurring in a second-order term t is denoted by $var(t)$. Instead of $T(\langle \Sigma, \mathcal{X}, \mathcal{W} \rangle)$ we shall simply write $T(\Sigma, \mathcal{X}, \mathcal{W})$.

Definition 3 *A* second-order substitution *in the language* $\mathcal{L}^2$ *is any expression of the form* $\{t_1/x_1, \ldots, t_n/x_n\}$ *where* $x_i \in \mathcal{X} \bigcup \mathcal{W}$ *are pairwise different variables and if* $x_i \in \mathcal{X}_m$, *then* $t_i \in T(\Sigma, \mathcal{X}, \{w_1, \ldots, w_m\})$, *and if* $x_i \in \mathcal{W}$, *then* $t_i \in T(\Sigma, \mathcal{X}, \emptyset)$. *The* domain $dom(\theta)$ *of such a substitution is the set* $\{x_1, \ldots, x_n\}$, *the* range $ran(\theta)$ *of* θ *is the set* $\{t_1, \ldots, t_n\}$.

Let θ *be a second-order substitution* $\{t_1/x_1, \ldots, t_n/x_n\}$, t *be a term. The result* $t\theta$ *of* application *of* θ *to* t *is defined inductively as follows:*

1. *For every* $f \in \Sigma_m$ *we have* $f(s_1, \ldots, s_m)\theta \rightleftharpoons f(s_1\theta, \ldots, s_m\theta)$.

2. *For every $x \in \mathcal{X} \bigcup \mathcal{W}$ of arity m*

$$x(s_1,\dots,s_m)\theta \rightleftharpoons \begin{cases} t_i\{s_1\theta/w_1,\dots,s_m\theta/w_m\} & \text{if } x \approx x_i,\ 1 \le i \le n \\ x(s_1\theta,\dots,s_m\theta) & \text{if } x \notin \{x_1,\dots,x_n\} \end{cases}$$

For purely technical reasons we extend the notation $x\theta$ to variables x of arity > 0 as follows. Let θ be as in Definition 3. Then

$$x\theta \rightleftharpoons \begin{cases} t_i & \text{if } x \approx x_i,\ 1 \le i \le n \\ x & \text{if } x \notin \{x_1,\dots,x_n\} \end{cases}$$

Definition 4 *A* second-order equation *in the language $\mathcal{L}^2$ is an expression of the form $s_1 = s_2$, where $s_1, s_2 \in T(\Sigma, \mathcal{X}, \emptyset)$. A* unifier *for $s_1 = s_2$ is a second-order substitution θ such that $s_1\theta \approx s_2\theta$. A substitution θ is a* unifier *for a set of equations S iff it is a unifier for every $E \in S$. A set of second-order equations possessing a unifier is called* unifiable.

For a set of second-order equations S, *var*(S) is the set of all variables occurring in S. Note that second order equations do not contain bound variables. The *second order unification problem* is the problem of determining whether any finite set of second order equations is unifiable.

By a routine inspection of the definitions we obtain

Lemma 1 *Let S be a unifiable set of equations in the language $\mathcal{L}^2$. Then there is a unifier θ for S such that*

1. *var(S) = dom(θ);*

2. *Variables in $\mathcal{X}$ do not occur in ran(θ).*

Substitutions θ satisfying these two conditions will be called *ground for S*. Note that for every equation $(s = t) \in S$ and for every substitution θ ground for S the terms $s\theta, t\theta$ are ground.

The following definition is technical:

Definition 5 *A set S of second order equations is called* reduced *iff all equations in S have either the form $s = t$ where s, t are terms not using variables of arity > 0 or the form $x(t_1,\dots,t_m) = t$, where x is a variable, $m > 0$ and $t, t_1,\dots,t_m$ are terms not using variables of arity > 0.*

Lemma 2 *For any set S of second-order equations one can effectively find a reduced set S' of second order equations such that S is unifiable iff S' is unifiable.*

Proof. Assume that S contains an equation $s = t$ such that s has an occurrence of a term $x(t_1,\dots,t_m)$. Let $y \in \mathcal{X}_0$ be a variable foreign to S. Then there is a term s' such that $s \approx s'\{x(t_1,\dots,t_m)/y\}$. Let S_1 be obtained from S by the replacement of $s = t$ by two equations $s' = t, x(t_1,\dots,t_m) = y$. By routine inspection of the definitions one can see that S is unifiable iff S_1 is unifiable.

With the help of such replacements one can obtain a system S' satisfying the claim.

□

3 Simultaneous rigid E-unification

Definition 6 *A* first-order language $\mathcal{L}^1$ *is a pair $\langle \Sigma, \mathcal{X} \rangle$ of disjoint sets, where Σ is finite and $\mathcal{X}$ is countable. For any such language Σ is the set of* function symbols *of $\mathcal{L}^1$ and $\mathcal{X}$ is the set of* variables *of $\mathcal{L}^1$. We assume that the set Σ is the union of disjoint sets Σ_n of function symbols* of arity n, $n \in \{0, 1, \ldots\}$. *Elements of Σ_0 are called* constants.

For the rest of this section we assume that $\mathcal{L}^1$ stands for a fixed first-order language $\langle \Sigma, \mathcal{X} \rangle$.

Definition 7 *The set $T(\mathcal{L}^1)$ of* first order terms *of the language $\mathcal{L}^1$ is defined inductively as follows: every $x \in \mathcal{X}$ belongs to $T(\mathcal{L}^1)$; for every $f \in \Sigma_n$ and $t_1, \ldots, t_n \in T(\mathcal{L}^1)$, $n \geq 0$, we have $f(t_1, \ldots, t_n) \in T(\mathcal{L}^1)$. A first-order term in $T(\mathcal{L}^1)$ is called* ground *iff it uses no variables.*

Instead of $T(\langle \Sigma, \mathcal{X} \rangle)$ we shall simply write $T(\Sigma, \mathcal{X})$.

Definition 8 *A* first-order substitution *in the language $\mathcal{L}^1$ is any expression of the form $\{t_1/x_1, \ldots, t_n/x_n\}$ where $x_i \in \mathcal{X}$ are pairwise different variables and $t_i \in T(\mathcal{L}^1)$. The* domain *$dom(\theta)$ of such a substitution is the set $\{x_1, \ldots, x_n\}$.*

Let θ be a first-order substitution $\{t_1/x_1, \ldots, t_n/x_n\}$, t be a term. The result $t\theta$ of application *of θ to t is defined inductively as follows:*

1. *For every $f \in \Sigma_m$ we have $f(s_1, \ldots, s_m)\theta \rightleftharpoons f(s_1\theta, \ldots, s_m\theta)$.*
2. *For every $x \in \mathcal{X}$*

$$x\theta \rightleftharpoons \begin{cases} t_i & \text{if } x \approx x_i,\ 1 \leq i \leq n \\ x & \text{if } x \notin \{x_1, \ldots, x_n\} \end{cases}$$

A first-order equation *in the language $\mathcal{L}^1$ is an expression of the form $s_1 = s_2$, where $s_1, s_2 \in T(\mathcal{L}^1)$.*

For a first-order equation $s = t$ and a set of equations E we write $E \vdash s = t$ to denote that the formula $\forall((\bigwedge_{e \in E} e) \supset s = t)$ is provable in first order logic with equality. For such formulas provability can be tested by the congruence closure algorithm (see [GNRS 92]).

For a set E of first-order equations and a substitution θ we denote by $E\theta$ the set of first-order equations $\{s\theta = t\theta \mid (s = t) \in E\}$.

Definition 9 *A* rigid equation *is an expression of the form $E \vdash_\forall s = t$, where E is a finite set of first order equations, $s = t$ is a first-order equation. Its* solution *is any first-order substitution θ such that $E\theta \vdash s\theta = t\theta$. A substitution θ is a solution of a set S of rigid equations iff θ is a solution of all rigid equations in S.*

The *simultaneous rigid E-unification problem* is the problem of determining whether any finite set of rigid equations possesses a solution.

We shall introduce one particular kind of a rigid equation that will be used as a technical tool for proofs in this paper. For any first-order language $\langle \Delta, \mathcal{Y} \rangle$ with at least one constant and for any constant $c \in \Delta_0$ introduce the following set of first-order equations:

$$Gr(\Delta, c) \rightleftharpoons \bigcup_{m=0}^{\infty} \{f(a_1, \ldots, a_m) = c \mid f \in \Delta_m \setminus \{c\},\ a_1, \ldots, a_m \in \Delta_0\}$$

We shall use the following two obvious lemmas:

Lemma 3 *Consider the rigid equation*

$$Gr(\Delta, c) \vdash_{\forall} x = c$$

A substitution θ is a solution of this equation iff $x\theta$ is a ground term of the language $\langle \Delta, \mathcal{Y} \rangle$.

We shall use the substitution notation $\{t_1/c_1, \dots, t_n/c_n\}$ where c_i are constants. This will denote the operation of the simultaneous replacement of *all* occurrences of c_i by t_i.

Lemma 4 *Let $c_1, \dots, c_n$ be pairwise different constants and $t_1, \dots, t_n$ be first order terms such that c_i does not occur in t_j for all $i, j \in \{1, \dots, n\}$. Then $c_1 = t_1, \dots, c_n = t_n \vdash s_1 = s_2$ iff $s_1\{t_1/c_1, \dots, t_n/c_n\} \approx s_2\{t_1/c_1, \dots, t_n/c_n\}$.*

4 Undecidability proof

In this section we reduce second order unification to simultaneous rigid E-unification. For the rest of this section we assume that $\mathcal{L}^2$ stands for a fixed second-order language $\langle \Sigma, \mathcal{X}, \mathcal{W} \rangle$.

Theorem 1 *There is an effective method that reduces second order unification to simultaneous rigid E-unification.*

Proof. Let S be a set of second-order equations of the language $\mathcal{L}^2$. By Lemma 2 we can assume that S is reduced. By Lemma 1 we can restrict ourselves to substitutions ground for S. Let $V = var(S)$ be the set of all variables occurring in S and let k be the maximal arity of variables in V. Introduce new constants $c_1, \dots, c_k$ foreign to Σ and define the first-order language $\mathcal{L}^1$ as $\langle \Sigma \bigcup \{c_1, \dots, c_k\}, \mathcal{X} \bigcup \mathcal{W} \rangle$. Let σ be the first-order substitution $\{c_1/w_1, \dots, c_k/w_k\}$ in this language.

Fix a constant $a \in \Sigma_0$. Consider the set R consisting of the following rigid equations in the language $\mathcal{L}^1$:

1. For every $x \in V$ of arity $m \geq 0$ the rigid equation

$$G_x \rightleftharpoons Gr(\Sigma \bigcup \{c_1, \dots, c_m\}, a) \vdash_{\forall} a = x$$

2. For every equation $(s = t) \in S$ where s, t use only variables of arity 0 the rigid equation

$$E_{s=t} \rightleftharpoons \vdash_{\forall} s = t$$

3. For every equation $(x(t_1, \dots, t_m) = t) \in S$ with $m > 0$ the rigid equation

$$C_{x(t_1, \dots, t_m) = t} \rightleftharpoons c_1 = t_1, \dots, c_m = t_m \vdash_{\forall} x = t$$

For any second-order substitution $\theta = \{t_1/x_1, \dots, t_l/x_l\}$ ground for S denote by $\hat{\theta}$ the first-order substitution $\{t_1\sigma/x_1, \dots, t_l\sigma/x_l\}$. Note that for every term t without variables of arity > 0 we have $t\theta \approx t\hat{\theta}$.

We prove the following technical

Lemma 5

1. *Let θ be a substitution ground for S. Then θ is a unifier for S iff $\hat{\theta}$ is a solution of R.*

2. *Every solution τ of R with $dom(\tau) = V$ has the form $\hat{\theta}$ for a suitable θ ground for S.*

Proof.

1. First we prove the "only if" part. Suppose θ is a unifier for S and consider any $x \in V$ of arity m. Since θ is ground for S, we have $x\theta \in T(\Sigma, \emptyset, \{w_1, \ldots, w_m\})$. Hence $x\hat{\theta} \in T(\Sigma \bigcup \{c_1, \ldots, c_m\}, \emptyset)$. By Lemma 3, $\hat{\theta}$ is a solution of G_x.

   Since θ is a unifier for every $(s = t) \in S$ with s, t without variables of arity > 0, we have $s\hat{\theta} \approx s\theta \approx t\theta \approx t\hat{\theta}$, which implies that $\hat{\theta}$ is a solution of every $E_{s=t}$.

   Let $(x(t_1, \ldots, t_m) = t) \in S$ and $m > 0$. Then

   $$\begin{array}{l}
   x\hat{\theta}\{t_1\hat{\theta}/c_1, \ldots, t_m\hat{\theta}/c_m\} \approx \text{ (since } t_i\theta \approx t_i\hat{\theta}) \\
   x\hat{\theta}\{t_1\theta/c_1, \ldots, t_m\theta/c_m\} \approx \text{ (using the definition of } \theta \text{ and } x\theta \in T(\Sigma, \emptyset, \{w_1, \ldots, w_m\})) \\
   x\theta\{t_1\theta/w_1, \ldots, t_m\theta/w_m\} \approx \text{ (by the definition of substitution application)} \\
   x(t_1, \ldots, t_m)\theta \approx \text{ (since } \theta \text{ is a unifier for } x(t_1, \ldots, t_m) = t) \\
   t\theta \approx \text{ (since } t\theta \approx t\hat{\theta}) \\
   t\hat{\theta} \approx \text{ (since } c_i \text{ do not occur in } t\hat{\theta}) \\
   t\hat{\theta}\{t_1\hat{\theta}/c_1, \ldots, t_m\hat{\theta}/c_m\}
   \end{array}$$

   We obtained $x\hat{\theta}\{t_1\hat{\theta}/c_1, \ldots, t_m\hat{\theta}/c_m\} \approx t\hat{\theta}\{t_1\hat{\theta}/c_1, \ldots, t_m\hat{\theta}/c_m\}$. By Lemma 4 this yields $c_1 = t_1\hat{\theta}, \ldots, c_m = t_m\hat{\theta} \vdash x\hat{\theta} = t\hat{\theta}$. Hence, $\hat{\theta}$ is a solution of $C_{x(t_1, \ldots, t_m)=t}$.

   Thus, $\hat{\theta}$ is a solution of R. By the same considerations we can prove the "if" part.

2. Suppose τ is any solution of R with $dom(\tau) = V$. Define the second-order substitution θ with $dom(\theta) = V$ by $x\theta \rightleftharpoons (x\tau)\{w_1/c_1, \ldots, w_m/c_m\}$, for all $x \in V$. Since τ is a solution of all G_x, by Lemma 3 we have $x\tau \in T(\Sigma \bigcup \{c_1, \ldots, c_m\}, \emptyset)$ for x of arity m, which implies $x\theta \in T(\Sigma, \emptyset, \{w_1, \ldots, w_m\})$. Hence, θ is ground for S. It is easy to see that $\hat{\theta} = \tau$.

□

We continue the proof of Theorem 1. Note that if there is a solution of R then there is a solution τ of R with $dom(\tau) = V$. Applying Lemmas 1 and 5 we obtain that S is unifiable iff R has a solution. □

In our construction we used a potentially infinite number of new constants c_i. One can note that the proof can be modified as well so that the language $\mathcal{L}^1$ of rigid equations will be fixed. The modification is as follows. Let $f, g \notin \Sigma$ be unary function symbols and $b \notin \Sigma$ be a constant. Replace all new constants c_i used in the proofs by terms of the form $f(g^i(b))$. The proof remains correct because $f(g^k(b))$ and $f(g^l(b))$ are not subterms of each other when $k \neq l$. This shows that second-order unification in the signature Σ is effectively reducible to simultaneous rigid E-unification in the signature $\Sigma \bigcup \{f, g, b\}$. By the undecidability of second order unification [Goldfarb 81] we obtain

Theorem 2 *Simultaneous rigid E-unification is undecidable.*

5 Conclusion

In [DeVo 95b] we show that simultaneous rigid E-unification is expressible in the $\exists^*$-fragment of intuitionistic logic with equality. Theorem 2 implies that this fragment of intuitionistic logic with equality is undecidable.

Our Theorem 1 also reveals another phenomenon: the proposal of using simultaneous rigid E-unification in matrix methods of theorem proving for classical first-order logic with equality in fact leads to second-order logic with all its inefficiencies. For matrix methods in classical logic with equality there are alternative approaches, e.g. the one proposed in [DeVo 94, DeVo 95a]. However, the problem of handling simultaneous rigid E-unification seems to be unavoidable in proof methods for intuitionistic logic with equality.

Bibliography

[Andrews 81] P.B. Andrews. Theorem proving via general matings. *Journal of the Association for Computing Machinery*, 28(2):193–214, 1981.

[Baaz 93] M. Baaz. Note on the existence of most general semi-unifiers. In *Arithmetic, Proof Theory and Computation Complexity*, volume 23 of *Oxford Logic Guides*, pages 20–29. Oxford University Press, 1993.

[Bibel 82] W. Bibel. *Automated theorem proving.* Vieweg Verlag, 1982.

[DeVo 94] A. Degtyarev and A. Voronkov. Equality elimination for semantic tableaux. UPMAIL Technical Report 90, Uppsala University, Computing Science Department, December 1994.

[DeVo 95a] A. Degtyarev and A. Voronkov. General connections via equality elimination. UPMAIL Technical Report 93, Uppsala University, Computing Science Department, January 1995. (To appear in Proc. WOCFAI'95.)

[DeVo 95b] A. Degtyarev and A. Voronkov. Simultaneous rigid E-unification is undecidable. UPMAIL Technical Report 105, Uppsala University, Computing Science Department, May 1995.

[DeMaVo 95] A. Degtyarev, Yu. Matijasevich, and A. Voronkov. Simultaneous rigid E-unification is not so simple. UPMAIL Technical Report 104, Uppsala University, Computing Science Department, April 1995.

[Fitting 88] M. Fitting. First-order modal tableaux. *Journal of Automated Reasoning*, 4:191–213, 1988.

[GNRS 92] J. Gallier, P. Narendran, S. Raatz, and W. Snyder. Theorem proving using equational matings and rigid E-unification. *Journal of the Association for Computing Machinery*, 39(2):377–429, 1992.

[GaRaSn 87] J.H. Gallier, S. Raatz, and W. Snyder. Theorem proving using rigid E-unification: Equational matings. In *Logic in Computer Science (LICS'87) (Ithaca, N.Y.)*, pages 338–346. IEEE Computer Society Press, 1987.

[Goldfarb 81] W. D. Goldfarb. The undecidability of the second-order unification problem. *Theoretical Computer Science*, 13:225–230, 1981.

[Loveland 68] D.W. Loveland. Mechanical theorem proving by model elimination. *Journal of the Association for Computing Machinery*, 15:236–251, 1968.