Semi-Unification

Rolf Socher-Ambrosius

Max-Planck-Institut für Informatik, Im Stadtwald,
D-W-6600 Saarbrücken, Germany
email: socher@mpi-sb.mpg.de

Abstract

Semi-unifiability is a generalization of both unification and matching. It is used to check nontermination of rewrite rules. In this paper an inference system is presented that decides semi-unifiability of two terms s and t and computes a semi-unifier. In contrast to an algorithm by Kapur, Musser et al, this inference system comes very close to the one for ordinary unification.

Keywords: Automatic Theorem Proving, Termination, Unification

1 Introduction

It is a well-known fact that termination is an undecidable property of rewrite systems, even in the case of a single rewrite rule [2]. For practical purposes, a number of rewrite orderings have been proposed as sufficient conditions for termination. In addition to such termination proofs, the ability to prove nontermination of a rule seems to be useful for Knuth-Bendix completion. For instance, if a newly generated rule can be easily shown to be nonterminating, then it is not necessary to try to orient it with the given termination ordering. Purdom [9] gives some practical results that demonstrate the practical value of non-termination proofs. One such sufficient condition [6] for nontermination is based on semi-unification. A similar idea based on nontermination is proposed by Plaisted [8].

More recently, another application of semi-unification has emerged in the area of type inference in extensions of the Milner Calculus. In [4], it is shown that the polymorphic type inference problem for recursive functions can be reduced to a generalized semi-unification problem.

The first algorithm for semi-unification [9] was shown to be incorrect by Kapur et al, [5]. In their paper, a correct algorithm is presented, which, however, seems to be unnecessarily complicated, both with respect to the formulation and the correctness proof. Their algorithm essentially tries to construct a terminating rewrite system. A refined version of this algorithm is shown to be of polynomial time complexity. Leiß [7] extends Kapur's algorithm to the *generalized* semi-unification problem. Unfortunately, he is not able to prove termination of the extended algorithm. According to [7], it is even an open question whether the generalized semi-unification problem is decidable.

In this paper, an inference system for semi-unification is presented that bears considerable similarity to the inference system for ordinary unification. The proofs of termination and correctness are straightforward. The inference system can be canonically transformed into an inference system on graphs, thus yielding an algorithm of polynomial time-complexity.

We use the standard notions on terms and substitutions: Let $\mathcal{T}=\mathcal{T}(\mathcal{F},\mathcal{V})$ denote the set of terms over the signature $\mathcal{F}$ and the set of variables $\mathcal{V}$. For any $t\in\mathcal{T}\backslash\mathcal{V}$, $\mathcal{H}ead(t)$ denotes the function symbol heading t. We use $\le$ ($<$) for the (proper) subsumption ordering on $\mathcal{T}$, i.e., $s\le t$, if there exists a substitution σ, such that $t=s\sigma$.

A substitution σ is an endomorphism on the term algebra, such that $x\sigma=x$ for all but a finite number of variables. By $\mathcal{D}om(\sigma)$ and $\mathcal{C}od(\sigma)$, we denote the sets $\{x\in\mathcal{V} \mid x\sigma \neq x\}$ and $\{x\sigma \mid x\in\mathcal{D}om(\sigma)\}$, respectively. A substitution σ is usually denoted by the set of termpairs $\{x\to x\sigma \mid x\in\mathcal{D}om(\sigma)\}$. Σ denotes the set of substitutions. A renaming is an injective substitution with $\mathcal{C}od(\rho)\subseteq\mathcal{V}$. We extend the quasi-ordering $\le$ to Σ by $\sigma\le\tau$ iff there exists $\theta\in\Sigma$ with $\tau=\sigma\theta$.

A substitution σ is a unifier of the pair (equation) $s\approx t$ of terms, iff $s\sigma=t\sigma$ holds, and it is a unifier of an equational system E, iff it is a unifier for each $e\in E$. The set of all idempotent unifiers of E is denoted by u(E). A most general unifier of E is a minimum of u(E) w.r.t. the subsumption quasi-ordering, and mgu(E) denotes the set of all most general unifiers of E.

We use the notation $E*\sigma$, where E is an equational system and $\sigma\in\Sigma$, to denote the system $E\sigma\cup\{y\approx y\sigma \mid y\in\mathcal{D}om(\sigma)\}$.

2 An Algorithm for Semi-unification

1 Definition *Let s and t be terms. We say s* ***semi-unifies*** *with t, iff there exist substitutions μ and σ, such that $s\sigma\mu = t\sigma$. In this case we call the pair (σ,μ) a semi-unifier of the pair $s\approx t$. We call (σ,μ) a semi-unifier of an equational system E, iff it is a semi-unifier for each $e\in E$. Let U, E be equational systems. We call (σ,μ) a semi-unifier of (E,U), iff σ is a unifier of U and (σ,μ) is a semi-unifier of E. The set of all semi-unifiers of E or (E,U) is denoted by su(E) or su(E,U), respectively.*

The problem to find a semi-unifier for a system $(E_1,\ldots,E_n,U)$ with arbitrary $n\in\mathbb{N}$ is referred to as generalized semi-unification problem, while the problem for n=1 is simply called semi-unification problem. In this section we give an inference system for semi-unification, which is shown to be correct and terminating under a particular strategy (i.e., a priority on the inference rules).

Semi-unification can be used as the basis of a sufficient condition for nontermination, as the following theorem ([6], [5]) shows:

2 Theorem *Let $l\to r$ be a rewrite rule. If there exists a subterm r' of r, such that l semi-unifies with r', then $l\to r$ is nonterminating.* □

Of course, this criterion can not be necessary for nontermination. A simple counterexample is given in [3]: Consider the rule fgx→ggffx, which is nonterminating (the term fggx, for instance, issues an infinite derivation), even though the left hand side fails to semi-unify with any subterm of the right hand side.

3 Example *Let $s=f(hy,x)$ and let $t=f(x,h^2y)$. We are looking for a pair (σ,μ), such that $s\sigma\mu = t\sigma$. It is easy to see that the pair $s\approx t$ has neither a unifier nor a matcher. The basic idea of the semi-unification algorithm is the following: The subproblem $(hy)\sigma\mu = x\sigma$ implies that $x\sigma$ must be of the form hu for some term u. We try $\sigma=\{x\to hu\}$ and obtain the subproblem $f(hy,hu) \approx f(hu, h^2y)$, which has the matcher $\mu=\{y\to u, u\to hy\}$.*

In the following we consider the semi-unification problem $s_0\approx t_0$. Let $\mathcal{V}_0 = \mathcal{V}ar(s_0,t_0)$. The rules operate on pairs (E,U), where E and U are equational systems. The initial system is given by (E_0,U_0), where $E_0=\{s_0\approx t_0\}$ and $U_0=\emptyset$. We assume given an infinite set $\mathcal{V}$ of variables with $\mathcal{V}_0\subseteq\mathcal{V}$, and a renaming ρ on $\mathcal{V}$, such that $\mathcal{V} = \bigcup_{n\in\mathbb{N}} \mathcal{V}_0\rho^n$. In other words, for each $x\in\mathcal{V}$, we have an infinite set of copies $\{x\rho^n \mid n\in\mathbb{N}\}$, which will be abbreviated by X.

As a matter of technicality, we shall consider only those semi-unifiers (σ, μ) that satisfy $\rho\sigma = \sigma\mu$. This is no loss of generality, since (σ,μ) is a semi-unifier of $s_0\approx t_0$, iff $(\sigma|_{\mathcal{V}_0}, \mu)$ is a semi-unifier of $s_0\approx t_0$, i.e., the value of σ on $\mathcal{V}\rho$ is irrelevant.

If $U = \{x_1\approx t_1,\ldots,x_n\approx t_n\}$ and $V\subseteq\mathcal{V}$, then $U|_V$ denotes the set $\{x_i\approx t_i \mid i=1,\ldots,n;\ x_i\in V\}$.

4 Definition *For any (finitely based) substitution σ we define (an infinitely based substitution) σ^* by*

$$\sigma^* = \{x\rho^n \to (x\sigma)\rho^n \mid x\in\mathcal{D}om(\sigma), n\in\mathbb{N}\}$$

5 Definition *The system $\Rightarrow_{SU}$ consists of the following inference rules:*

Decomposition:

$(E\cup\{ft_1\ldots t_n \approx fs_1\ldots s_n\},U) \quad \Rightarrow_D \quad (E\cup\{t_1\approx s_1,\ldots,t_n\approx s_n\},U)$

Merge left:

$(E\cup\{t\approx x\},U) \quad \Rightarrow_{ML} \quad (E\sigma^*\cup\{y\approx y\rho \mid y\in\mathcal{V}ar(t)\}, U*\sigma^*)$, if $X\cap\mathcal{V}ar(t\rho)=\emptyset$, $t\notin\mathcal{V}$, and $\sigma=\{x\to t\rho\}$

Merge right:

$(E\cup\{x\approx s, x\approx t\}, U) \quad \Rightarrow_{MR} \quad ((E\cup\{x\approx s\})\sigma^*, U*\sigma^*)$, if $\sigma\in mgu(s,t)$

*The following **failure rules** enhance the efficiency of the procedure:*

$(E\cup\{ft_1\ldots t_n \approx gs_1\ldots s_m\},U) \quad \Rightarrow_F \quad$ failure, if $f\neq g$

$(E\cup\{t\approx x\},U) \quad \Rightarrow_F \quad$ failure, if $X\cap\mathcal{V}ar(t\rho)\neq\emptyset$

$(E\cup\{x\approx s, x\approx t\}, U) \quad \Rightarrow_F \quad$ failure, if $mgu(s,t)=\emptyset$

An inference step $(E\cup\{t\approx x\},U) \Rightarrow_{ML} (E\sigma^*\cup\{y\approx y\rho \mid y\in\mathcal{V}ar(t)\}, U*\sigma^*)$ is called an application of ML **on x in** $E\cup\{t\approx x\}$. An ML step on a variable x is denoted by $\Rightarrow_{ML(x)}$.

Example *Let $E_0=\{fy\approx z, fx\approx y, z\approx f^3x\}$, $U_0=\emptyset$. As a matter of convenience, we write $v' = v\rho$ and $v'' = v\rho^2$ for any variable v. As a first step, we can apply the ML rule to $fy\approx z$:*

$$(\{fy\approx z, fx\approx y, z\approx f^3x\},\emptyset) \Rightarrow (\{y\approx y', fx\approx y, fy'\approx f^3x\},\{z\approx fy'\})$$

Now we apply the ML rule to $fx\approx y$:

$(\{y{\approx}y', fx{\approx}y, fy'{\approx}f^3x\},\{z{\approx}fy'\}) \Rightarrow (\{fx'{\approx}fx'', x{\approx}x', ffx''{\approx}f^3x\},\{z{\approx}ffx'', y{\approx}fx', y'{\approx}fx''\})$

Note that it is not necessary to include explicitely into U all equations $y{\approx}fx'$, $y'{\approx}fx''$, $y''{\approx}fx'''$, …, rather, it would be sufficient to include only the first one. After several decomposition steps, we obtain the solution

$(\{x'{\approx}x'', x{\approx}x', x''{\approx}fx\},\{z{\approx}ffx'', y{\approx}fx', y'{\approx}fx''\})$

which indeed is a semi-unifier for the original system.

In the following we first prove that the inference rules are correct, and then we show that the inference system SU is terminating under a particular priority for the rules.

In the following we shall prove that each terminating SU inference $(E_0,U_0) \Rightarrow (E_1,U_1) \Rightarrow \ldots \Rightarrow (E_n,U_n)$ yields a system (E_n,U_n) that either represents a solution to the system (E_0,U_0), if this system is solvable, or it terminates with failure, if the original system is unsolvable.

In the next lemmata, $(E_0,U_0) \Rightarrow (E_1,U_1) \Rightarrow \ldots \Rightarrow (E_n,U_n)$ denotes an SU-derivation with $E_0=\{s_0{\approx}t_0\}$ and $U_0=\emptyset$.

6 Lemma *Let $U=U_i$ for some $i=1,\ldots,n$. Then for each $e\in U$, there are $x\in\mathcal{V}_0$ and $t\in\mathcal{T}$ with $X\cap\mathcal{V}ar(t)=\emptyset$, such that $e = x{\approx}t$. Moreover, the mapping $\sigma_U := \{x\to t \mid x{\approx}t\in U\}$ is a well-defined and idempotent substitution and $\sigma_U \in mgu(U)$ holds.*

Proof: This is obvious by the construction of the derivation system. □

7 Lemma $su(E_{i+1}, U_{i+1}) = su(E_i, U_i)$ *holds for any* $i=0,\ldots,n$.

Proof: First, we remark that $\mathcal{D}om(\sigma_{U_i}) \cap \mathcal{V}ar(E_i) = \emptyset$ holds for $i=0,\ldots,n$. Let $(E,U)=(E_i,U_i)$ and $(E',U')=(E_{i+1},U_{i+1})$. Note that we consider only semi-unifiers (σ,μ) with $\rho\sigma = \sigma\mu$.

Case 1: (E', U') is derived from (E, U) by a decomposition step. Then obviously su(E', U') = su(E, U) holds.

Case 2: $(E, U) \Rightarrow_{ML} (E', U')$ holds. Let $E = \{t{\approx}x, s_1{\approx}t_1,\ldots, s_n{\approx}t_n\}$ and $U = \{x_1{\approx}r_1,\ldots, x_m{\approx}r_m\}$, with $\mathcal{V}ar(t)=\{y_1,\ldots,y_k\}$, such that $E' = \{y_1{\approx}y_1\rho, \ldots, y_k{\approx}y_k\rho, s_1\theta{\approx}t_1\theta, \ldots, s_n\theta{\approx}t_n\theta\}$ and $U' = \{x_1{\approx}r_1\theta, \ldots, x_m{\approx}r_m\theta, x{\approx}t\rho\}$ hold for $\theta = \{x\to t\rho\}^* = \{x\rho^i \to t\rho^{i+1} \mid i\in\mathbb{N}\}$.

Let $(\sigma,\mu)\in su(E, U)$. We have $x\sigma = t\sigma\mu = t\rho\sigma = x\theta\sigma$, and $x\rho^i\sigma = x\sigma\mu^i = t\sigma\mu^{i+1} = t\rho^{i+1}\sigma = x\rho^i\theta\sigma$ for any $i\in\mathbb{N}$, and $y\theta\sigma = y\sigma$ for $y\notin X$, which shows that $\theta\sigma = \sigma$. From $x_i\sigma = r_i\sigma = r_i\theta\sigma$ now follows $\sigma\in u(U')$. Finally, from $y_i\sigma\mu = y_i\rho\sigma$, $i=1,\ldots,k$; and $s_i\theta\sigma\mu = s_i\sigma\mu = t_i\sigma = t_i\theta\sigma$, $i=1,\ldots,n$, follows $(\sigma,\mu) \in su(E')$.

Conversely, assume that $(\sigma,\mu)\in su(E',U')$. This implies $x\sigma = t\rho\sigma = x\theta\sigma$, hence $\theta\sigma = \sigma$. We have $t\sigma\mu = t\rho\sigma = x\sigma$, $x_j\sigma = r_j\sigma$ for $j=1,\ldots,m$ and $s_i\sigma\mu = t_i\sigma$ for $i=1,\ldots,n$. Hence $(\sigma,\mu)\in su(E,U)$.

Case 3: $(E, U) \Rightarrow_{MR} (E', U')$ holds. Let $\{x{\approx}t, x{\approx}t'\} \subseteq E$, and let $\tau_0\in mgu(t,t')$, $\tau = \tau_0{}^*$, such that $E'= E\tau$ and $U' = U{*}\tau$. First, suppose that $(\sigma,\mu)\in su(E,U)$.

We have $x\sigma\mu = t\sigma$ and $x\sigma\mu = t'\sigma$. Hence $\sigma\in u(t,t')$, and with $\tau_0\in mgu(t,t')$ follows $\tau_0\leq\sigma$, that is, $\tau_0\sigma = \sigma$. Let $y\in\mathcal{D}om(\tau_0)$. Then for any $i\in\mathbb{N}$, $y\rho^i\tau\sigma = y\rho^i\tau_0{}^*\sigma = y\tau_0\rho^i\sigma = y\tau_0\sigma\mu^i = y\sigma\mu^i = y\rho^i\sigma$, which shows that $\tau\sigma = \sigma$.

Moreover, from $\sigma\in u(U)$ and $\sigma_U = mgu(U)$ follows $\sigma_U\leq\sigma$. This yields $\sigma_{U'} = \sigma_U\tau \leq \sigma$, hence $\sigma\in u(U')$. Moreover, from $\tau\sigma = \sigma$ follows $(\tau\sigma,\mu) = (\sigma,\mu)\in su(E)$, which yields $(\sigma,\mu) \in su(E\tau) = su(E')$.

Conversely, let $(\sigma,\mu)\in su(E',U')$. Then $\sigma_U\tau = \sigma_{U'} \le \sigma$ and $\tau \le \sigma$, which implies $\sigma_U \le \sigma$, that is, $\sigma\in u(U)$. Moreover, from $(\sigma,\mu) \in su(E') = su(E\tau)$ follows $(\sigma,\mu) = (\tau\sigma,\mu)\in su(E)$. □

8 Lemma *Let $(E_0,U_0) \Rightarrow \ldots \Rightarrow (E_n,U_n)$ with $U_0=\emptyset$, such that (E_n,U_n) is irreducible by the inference rules $\Rightarrow_D$, $\Rightarrow_{MR}$ and $\Rightarrow_{ML}$.*

a) *If (E_0,U_0) is not semi-unifiable then one of the failure conditions holds.*

b) *If (E_0,U_0) is semi-unifiable, then E_n is of the form $\{x_1\approx t_1,\ldots,x_n\approx t_n\}$, such that $x_i\neq x_j$ for $i\neq j$. Moreover, the pair (σ_{U_n},μ_{E_n}) is a semi-unifier of (E_0,U_0), where $\mu_{E_n} = \{x_1\to t_1, \ldots, x_n\to t_n\}$.*

Proof: Let $(E,U)=(E_n,U_n)$, and let $\sigma=\sigma_U$.

a) Assume that none of the failure conditions holds. From lemma 7 follows $\sigma = mgu(U)$ and $\mathcal{Dom}(\sigma)\cap\mathcal{Var}(E)=\emptyset$. Since E is irreducible, E is of the form $\{x_1\approx t_1,\ldots,x_n\approx t_n\}$, such that $x_i\neq x_j$ for $i\neq j$, which implies that $\mu = \{x_1\to t_1, \ldots, x_n\to t_n\}$ is a well-defined substitution. Finally, we have $x_i\sigma\mu = x_i\mu = t_i = t_i\sigma$ for $i=1,\ldots,n$, that is, $(\sigma,\mu)\in su(E)$. From the previous lemma follows that $(\sigma,\mu)\in su(E_0,U_0)$.

b) If (E_0,U_0) is semi-unifiable, then by the previous lemma, (E,U) is semi-unifiable. Let $(\sigma,\mu)\in su(E,U)$. We show that in this case none of the failure conditions holds.

Suppose, there is $s\approx t\in E$ with $s,t\notin\mathcal{V}$. If $\mathcal{Head}(s)\neq\mathcal{Head}(t)$, then E is not semi-unifiable, which is a contradiction, and if $\mathcal{Head}(s)=\mathcal{Head}(t)$, then the MR rule applies to E, which contradicts the irreducibility of E.

Let $t\approx x\in E$ with $x\rho^m\in\mathcal{Var}(t)$ for some $m\in\mathbb{N}$. Since $x\rho^m\in\mathcal{Var}(t)$, for each $i=1,\ldots,m$, there exists $s(i)\le n$ and an ML-step $(E_{s(i)},U_{s(i)}) \Rightarrow (E_{s(i)+1},U_{s(i)+1})$ such that $x\rho^i \approx x\rho^{i+1} \in E_{s(i)+1}$. Let $\varphi=\sigma_U$. Since the the merge left and decomposition rules do not remove equations from E_i, the final system E contains equations $x\varphi \approx x\rho^1\varphi,\ldots,x\rho^{m-1}\varphi \approx x\rho^m\varphi$, and $t[x\rho^m] \approx x$. Since the variables x and $x\rho^m$ both occur in E, we have $x=x\varphi$ and $x\rho^m=x\rho^m\varphi$. The pair (σ,μ) is a semi-unifier for E, hence we have

$$x\varphi\sigma\mu = x\rho^1\varphi\sigma,\ x\rho^1\varphi\sigma\mu = x\rho^2\varphi\sigma,\ldots,x\rho^{m-1}\varphi\sigma\mu = x\rho^m\varphi\sigma, \text{ and } t[x\rho^m]\sigma\mu = x\sigma$$

Hence we can infer $x\sigma\mu^m = x\varphi\sigma\mu^m = x\rho^m\varphi\sigma = x\rho^m\sigma$ and $t[x\sigma\mu^{m+1}] = x\sigma$, which is a contradiction. If, on the other hand, $x\rho^m\notin\mathcal{Var}(t)$ for all $m\in\mathbb{N}$, then the ML rule applies to E, which is a contradiction.

So we have $E = \{x_1\approx t_1,\ldots,x_n\approx t_n\}$. If $x_i=x_j$ for $i\neq j$, then the pair $t_i\approx t_j$ is unifiable, because E is semi-unifiable, and thus the MR rule applies to E, which is a contradiction.

Hence μ_E is a well-defined substitution and from $\mathcal{Dom}(\sigma_U)\cap\mathcal{Var}(E)=\emptyset$ follows $x_i\sigma\mu = x_i\mu = t_i = t_i\sigma$ for $i=1,\ldots,n$, that is, $(\sigma_U, \mu_E)\in su(E,U)$, and, by the previous lemma, $(\sigma_U, \mu_E)\in su(E_0,U_0)$. □

Next we show termination of the inference system. We modify the system SU in the following way: Let SU^* be the system consisting of the decomposition rule, the MR rule, and the ML rule together with the following strategy: The decomposition rule obtains highest priority, and the MR rule obtains the second highest one.

In the next lemmata, $(E_0,U_0) \Rightarrow (E_1,U_1) \Rightarrow \ldots \Rightarrow (E_n,U_n)$ denotes an SU^*-derivation with $E_0=\{s_0\approx t_0\}$ and $U_0=\emptyset$.

9 Lemma *If $x\rho \in \mathcal{V}ar(E_i)$ for some $x \in \mathcal{V}$, then $x \approx x\rho \in E_i$.*

Proof. Let j be the least number such that $x\rho \in \mathcal{V}ar(E_j)$. Then, $(E_{j-1},U_{j-1}) \Rightarrow_{ML} (E_j,U_j)$, and by definition of the ML rule, $x \approx x\rho \in E_j$. Now suppose there is some $i>j$ with $x \approx x\rho \notin E_i$. We can assume that i is the least such integer. Consider the step $(E_{i-1},U_{i-1}) \Rightarrow (E_i,U_i)$. This step is either an ML or an MR step, yielding some substitution σ^*. Since $x \approx x\rho \notin E_i$, $x \in \mathcal{D}om(\sigma^*)$ or $x\rho \in \mathcal{D}om(\sigma^*)$. If $x \in \mathcal{D}om(\sigma^*)$, then by definition of the mapping *, $x\rho \in \mathcal{D}om(\sigma^*)$. Hence $x\rho \in \mathcal{D}om(\sigma^*)$, which implies $x\rho \notin \mathcal{V}ar(E_i)$, contradicting the asssumption of the lemma. □

10 Lemma *Let $x \approx t \in E_i$ with $t \notin \mathcal{V}$ and $\mathcal{V}ar(E_i) \cap X\rho = \emptyset$. If there is no $n \geq i$ such that ML applies on x in E_n, then there exists no $j \geq i$, such that the ML rule applies on $x\rho^k$ with $k \geq 1$ in E_j.*

Proof: First, we can conclude from the assumption of the lemma that for each $j' \geq i$, there exists $t' \notin \mathcal{V}$ with $x \approx t' \in E_{j'}$. Without loss of generality, we assume that $x \approx t \in E_{j'}$ for each $j' \geq i$.

Since $x\rho \notin \mathcal{V}ar(E_i)$, the only way to generate the variable $x\rho$ in some step $(E_j,U_j) \Rightarrow (E_{j+1},U_{j+1})$, $j \geq i$, is by an application of the ML rule on some equation $s \approx y$ in E_j with $x \in \mathcal{V}ar(s)$. However, as $x \approx t \in E_j$, we have $\{x \approx t, x \approx x\rho\} \subseteq E_{j+1}$, so the next inference step is an MR step yielding the substitution $\{x\rho \to t\}$, which shows that $x\rho$ does not occur in E_{j+2}. An induction argument now proves the assertion of the lemma. □

11 Theorem *The inference system SU* is terminating.*

Proof: Let $x \in \mathcal{V}_0$ and let n be the least integer such that the ML rule is applied on some equation $t \approx x\rho^n$ in E_i. Let i be the least such number. Without loss of generality, we can assume that n=1. Then we can conclude that $x \approx x\rho \in E_i$, which in turn implies $x \approx t \in E_{i+1}$ and $\mathcal{V}ar(E_{i+1}) \cap X\rho = \emptyset$. Since the ML rule is not applied on x in any E_j, the assumption of the previous lemma is satisfied. Hence we can conclude that there is no $j \geq i$ such that the ML rule applies on any $x \in X$ in E_j.

Now we have proved that for any $x \in \mathcal{V}_0$, the number of applications of the ML rule on any $x' \in X$ is finite. As $\mathcal{V}_0$ is finite, we can infer that the ML rule can be applied only finitely many times in any SU^* derivation. The system consisting of the remaining two rules, the decomposition rule and the MR rule, is a subsystem of the inference system for ordinary unification, and hence this system is terminating, too, which proves the assertion of the lemma. □

Lemma 8 and the previous theorem yield the following

12 Theorem *The inference system SU^* describes a sound and complete semi-unification algorithm.*

3 A Polynomial Algorithm

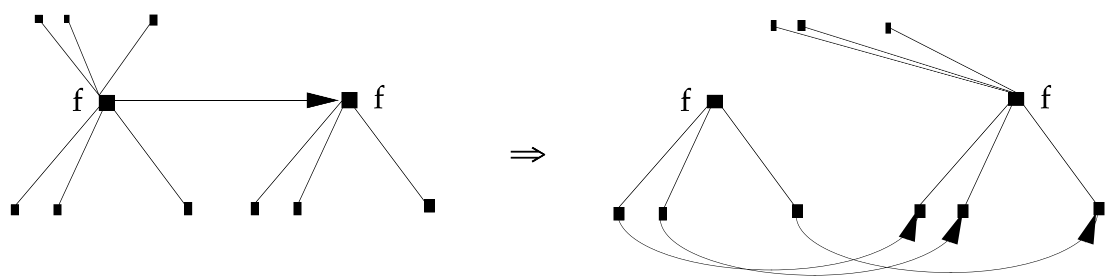
The inference system described in the previous section can easily be modified to operate on directed acyclic graphs in order to obtain a polynomial-time algorithm. The proceeding is very close to an algorithm by Corbin and Bidoit [1], which has time complexity of $\Theta(n^2)$, where n is the number of symbols occurring in the unification problem. In the following we work with a directed acyclic graph G = (N,L). In addition to the regular edges denoting the subterm relation, we use the following edges between two nodes s and t or x and t, respectively:

$s \rightarrow t$	denotes the semi-unification problem $s \approx t$
$s \leftrightarrow t$	denotes the unification problem $s \approx t$
$x \Rightarrow t$	denotes the solved unification problem $x \approx t$

A graph (N,L) thus corresponds to a system (E,U) in the following way: $s \rightarrow t \in L$ denotes that the pair $s \approx t$ is in E, $x \Rightarrow t \in L$ denotes that the pair $x \approx t$ is in U, and finally $s \leftrightarrow t \in L$ denotes that $\sigma \in mgu(s,t)$ has to be applied to U and E according to the merge right rule. The renaming ρ is adapted in an obvious way to the graph representation.

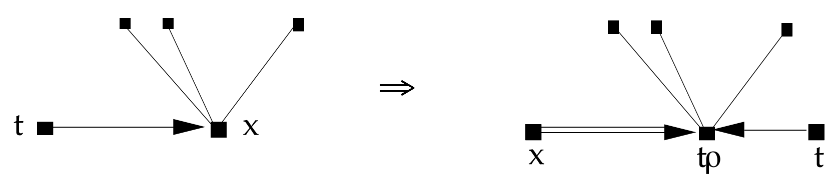
The inference rules correspond to graph rewriting rules in the following way:

Decomposition



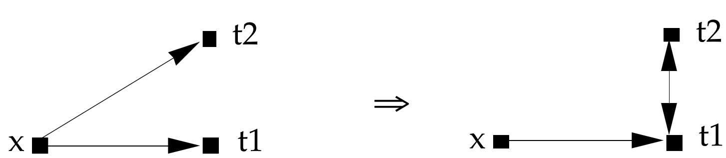
This rule moves the $\rightarrow$ edge between nodes s and t down to the children of s and t, respectively, provided that $\mathcal{Head}(s)=\mathcal{Head}(t)$, and redirects the regular edges connected with s to t.

Merge left



Each regular edge, $\rightarrow$-edge, $\Rightarrow$-edge, or $\leftrightarrow$-edge connected with x is redirected to a renamed copy $t\rho$ of (the subgraph) t, provided that no leaf node of the form x^m is below t. Moreover, the edge $t \rightarrow x$ is replaced by $x \Rightarrow t\rho$.

Merge right



The edge $x \rightarrow t_2$ is replaced by an edge $t_2 \leftrightarrow t_1$.

Moreover, we have the obvious failure rules:

Failure

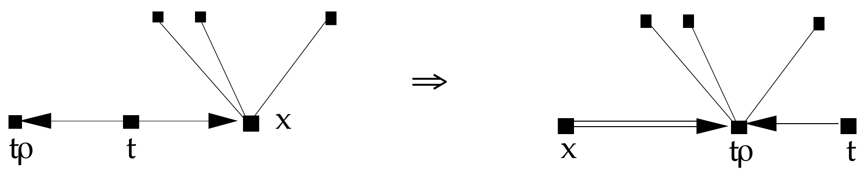
$s \to t \quad \Rightarrow \quad$ failure, if $\mathcal{H}ead(s) \neq \mathcal{H}ead(t)$

$t \to x \quad \Rightarrow \quad$ failure, if there is a leaf node x^m below t

The inference rules for edges of the form $\leftrightarrow$ are the ones given in [1].

We briefly sketch why the algorithm is guaranteed to run in time $\Theta(v^2n^2)$, where $v=|\mathcal{V}|$ and $n=|s|+|t|$. The proof of theorem 11 shows that the maximal number of new nodes generated by the algorithm is vn. We modify the semi-unification algorithm in the following way: each new subtree $t\rho$ that is created during the algorithm, is created at the beginning of the algorithm rather than during a merge left step. So we obtain a new graph $G'=(N',L')$ with $|N'| \leq vn$ nodes. The merge left rule is modified in an obvious way:

Merge left (modified)



Now it is easy to see that (i) the modified semi-unification algorithm applied to the graph G' takes the same number of steps as Corbin and Bidoit's algorithm on G' and (ii) the modified algorithm on G' takes the same number of steps as the original one on G. This proves the claim on the time complexity of the semi-unification algorithm.

Acknowledgement

I would like to thank Harald Ganzinger and Peter Barth for reading an earlier draft of the paper.

References

[1] J. Corbin, M. Bidoit. A Rehabilitation of Robinson's Unification Algorithm. In: R.E.A. Mason (Ed.) *Information Processing 83*. Elsevier 1983, 909-914.

[2] M. Dauchet. Simulation of Turing Machines by a Left-Linear Rewrite Rule. In: N. Dershowitz (Ed.) *Proc. of 3rd International Conference on Rewriting Techniques and Applications*, Chapel Hill (1989). Springer LNCS 355, 109-120.

[3] N. Dershowitz. Termination of Rewriting. In: J.-P. Jouannaud (Ed.). *Rewriting Techniques and Applications*. Academic Press 1987, 69-116.

[4] F. Henglein. Type inference and semi-unification. In: Proc. of ACM Conference on LISP and functional programming. ACM Press, New York, 1988.

[5] D. Kapur, D. Musser, P. Narendran and J. Stillman. Semi-Unification. *Theoretical Computer Science* **81** (1991), 169-187.

[6] D.S. Lankford, D.R. Musser. A Finite Termination Criterion. Unpublishd Draft, USC Information Sciences Institute, Marina del Rey, 1978.

[7] H. Leiß. A Semi-Unification Algorithm? Extended abstract. In: H.-J. Bürckert and W. Nutt (Eds.): *UNIF '89: Extended abstracts of the thirs Int. Workshop on Unification.* SEKI-Report SR-89-17, University of Kaiserslautern, 1989.

[8] D. A. Plaisted. A simple non-termination test for the Knuth-Bendix method. In: J. Siekmann (Ed.): 8th International Conference on Automated Deduction, Oxford (1988). Springer LNCS 230, 79-88.

[9] P. W. Purdom. Detecting Looping Simplifications. In: P. Lescanne (Ed.). *Proc. of 2nd International Conference on Rewriting Techniques and Applications*, Bordeaux (1987). Springer LNCS 256, 54-61.