A Decidable Case of the Semi-Unification Problem (Draft Version)

Hans Leiß
Centrum für Informations- und Sprachverarbeitung
Universität München
Leopoldstraße 139
D-8000 München 40
Germany

Fritz Henglein
DIKU Dept. of Computer Science,
University of Kopenhagen
Universitetsparken 1
DK-2100 Copenhagen Ø
Denmark
Internet: henglein@diku.dk

January 10, 1991

Abstract

Semi-unification is a common generalization of unification and matching. The semi-unification problem is to decide solvability of finite sets of equations $s = t$ and inequations $\tilde{s} \leq_i \tilde{t}$ between first-order terms, with different inequality relations $\leq_i$, $i \in I$. A solution consists of a substitution T_0 and residual substitutions T_i, $i \in I$, such that , respectively, $T_0(s) = T_0(t)$ and $T_i(T_0(\tilde{s})) = T_0(\tilde{t})$.

The semi-unification problem has recently been shown to be undecidable [9]. We present a new subclass of decidable semi-unification problems, properly containing those over monadic languages. In our 'quasi-monadic' problems, function symbols may be of arity > 1, but only terms with at most one free variable are admitted.

1 Introduction

The semi-unification problem, in its simplest form, is the problem to find out whether, given two first-order terms s and t, there is a substitution T such that the refined term $T(s)$ *matches* the refined term $T(t)$. That is, an *in*equation $s \leq t$ is to be solved by finding two substitutions T and R such that $R(T(s)) = T(t)$, or detecting that no such solution exists.

In this simple form, the problem was introduced by Kapur e.a.[6] and shown to be decidable in polynomial time. Kapur e.a. were interested in a criterion for nontermination of the Knuth-Bendix completion algorithm.

In a more general form, the same problem arose in efforts to improve the type inference algorithm of the programming language ML, independently in work of Henglein[2], Leiß[11, 13], and Kfoury, Tiuryn, and Urcyczyn[8, 9]. In this case, several inequations $s_1 \leq_1 t_1, \ldots, s_k \leq_k t_k$ have to be solved simultaneously, with possibly different matching substitutions $T_1, \ldots, T_k$, one for each inequation.

A third source of the semi-unification problem was the work of Baaz (see[15]) on Kreisel's problem on the length of proofs in Peano Arithmetic (cf. Footnote 3 in the Appendix to Takeuti[18]). In this context, Pudlák[15] showed that the semi-unification problem for more than two inequality relations can be reduced to the problem with two inequality relations.

In the second and third case, the point was to find a most general type scheme for a recursive function, and a most general proof, respectively.

While it was soon realized that solvable instances of the problem always do have most general solutions [3, 15, 9], the solvability question itself remained open for a while. Contrary to several conjectures to the opposite, Kfoury e.a.[9] recently showed that the semi-unification problem in general is undecidable. Subsequently, Dörre and Rounds[1] have shown that the semi-unification problem for feature structures, where infinite rational terms are admitted in the solutions, is also undecidable.

For the above-mentioned application on type inference, the undecidability of the semi-unification problem implies the undecidability of the typability of recursive function definitions with respect to a 'polymorphic recursion' typing scheme [14, 2, 13, 8]. On the other hand, the 'polymorphic unification' problem of Kannelakis and Mitchell[5], which arose in characterizing typability of ML-programs, translates to a decidable case of the semi-unification problem, where a certain acyclicity condition on dependencies between variables holds[10].

For the applications in proof theory, the solvability of the relevant instances is given in advance, but information on the form of the most general solution is very much needed, as Kreisel [7] pointed out. Finding parameters of the problem instances that determine properties of the most general solutions is a relevant issue here.

A few decidable cases of the semi-unification problem have been presented so far: *uniform* semi-unification[6, 16], where only one inequality relation occurs, *left-linear* semi-unification[9, 4] (see Section 3 below), the above mentioned *acyclic* semi-unification[10], and semi-unification in at most two variables[13].

In the present paper we present a new class of decidable semi-unification problems[1], including the semi-unification problem for monadic languages. In our 'quasi-monadic' problems, function symbols may be of arity > 1, but terms may have at most one free variable (though they may be non-linear). Another solvable class, which extends the semi-unification problem for monadic languages in a different dimension, is left-linear semi-unification. Presently, the methods to decide solvability for instances of these two classes are very different, but we hope that a uniform treatment is possible. (See Section 3 for the relation between left-linear and quasi-monadic problems.)

From the decidability result of Kapur e.a. it follows that typability for polymorphic *linear* recursive functions - which occur only once in their defining term - is decidable. We do not claim that our decidability result for semi-unification also gives a nice class of polymorphic recursive function definitions with decidable typability problem. Instead, we view the semi-unification problem as an important foundational problem. Isolating decidable subproblems therefore is an effort to understand better the structure of this problem.

[1] extending a partial result announced in Theorem 2, (19), of [13]

2 The Semi-Unification Problem

Let L be a first-order functional language, with a countably infinite set Var of variables. Let $R = \{\leq_i \mid i \in I\}$ be a finite set of binary relation symbols, fixing $I = \{1, \ldots, k\}$ for the rest of the paper. A finite set S of equations and inequations in the language $L \cup R$ is an *instance of the semi-unification problem over L and R.* A *solution* for S is a sequence $T = (T_0, \ldots, T_k)$ of substitutions such that $T_0(s) \equiv T_0(t)$ for each equation $s = t$ in S, and $T_i(T_0(s)) \equiv T_0(t)$ for each inequation $s \leq_i t$ in S.

If $T = (T_0, \ldots, T_k)$ is a solution of S, we call its *main* substitution T_0 a *semi-unifier of* S, and $T_1, \ldots, T_k$ its *residual* or *matching* substitutions.

Another solution $\tilde{T} = (\tilde{T}_0, \ldots, \tilde{T}_k)$ is *more general* than T, if there is a substitution U such that $T_0(x) = U(\tilde{T}_0(x))$ for each variable x occcurring in S. A *most general* solution is a solution that is more general that any other solution. A *most general semi-unifier* for S is the main substitution of a most general solution for S.

Lemma 1 *(Pudlák[15], Henglein[2], Kfoury e.a.[9]) A solvable instance of the semi-unification problem has a most general solution.*

Example 1 [12] Let S be $\{x \leq_i y,\ f(z) \leq_i x\}$. Then $T = (T_0, T_i) = ([f(z)/x, f(z)/y], Id)$ is a solution of S, since $T(x) = f(z) = T(y)$ and $T(f(z)) = f(z) = T(x)$, but it is not the most general one. A more general solution, and in fact the most general one, is $\tilde{T} = (\tilde{T}_0, \tilde{T}_i) = ([f(z_1)/x, f(z_2)/y],\ [z_2/z_1, z_1/z])$, since

$$\tilde{T}(x) = \tilde{T}_i(\tilde{T}_0(x)) = \tilde{T}_i(f(z_1)) = f(z_2) = \tilde{T}_0(y) = \tilde{T}(y),$$

$$\tilde{T}(f(z)) = \tilde{T}_i(\tilde{T}_0(f(z))) = \tilde{T}_i(f(z)) = f(z_1) = \tilde{T}_0(x) = \tilde{T}(x),$$

and $T_0 = [z/z_1, z/z_2] \circ \tilde{T}_0$ can be factored through $\tilde{T}_0$.

2.1 An Algebraic Form of the Semi-Unification Problem

The semi-unification problem can also be stated in an equational language with unary function variables, rather than using inequality relations. In this formulation, a simple transformation calculus for semi-unification problems can be given that is very close to, and a generalization of, the transformations sometimes used to present unification algorithms. (For example, see Snyder and Gallier[17])

Instead of extending L by the relation symbols of R, we augment L by the set I of unary function symbols, which act as L-homomorphisms, and are written as right exponents. That is, for each i and j in I we consider additional compound terms

$$f(t_1, \ldots, t_m)^i := f(t_1^i, \ldots, t_m^i)$$

and additional compound 'variables' x^i, $(x^i)^j$ etc. A peculiarity will be that we have to consider x^i to be free in $(x^i)^j$, since we want to be able to substitute any term for x^i in $(x^i)^j$.

Definition 1 *Let I^* be the set of finite sequences of elements from I, and Var be the denumerably infinite set of variables of L. Let Var^* be $Var \times I^*$, and L^* be the language with function symbols of L and Var^* as its variables. A pair (x, v) will be written as x^v in the following, and if v is the empty word ϵ, x^v will be identified with x. The set of free variables of a term t is defined via $free(x^\epsilon) = \{x\}$ and $free(x^{vi}) = \{x^{vi}\} \cup free(x^v)$.*

Definition 2 *For a sequence $T = (T_0, \ldots, T_k)$ of substitutions $T_j : Var \to L$-terms and each L^*-term t, define $T(t)$ inductively by*

$$T(x) = T_0(x), \quad T(x^{vi}) = T_i(T(x^v)), \quad T(f(t_1, \ldots, t_m)) = f(T(t_1), \ldots, T(t_m)).$$

Proposition 1 *If $T = (T_0, \ldots, T_k)$ is a sequence of substitutions, then for each variable x^v and L^*-terms s and t, we have*

i) $T(t^i) = T_i(T(t))$, for each $i \in I$.

ii) $T(x^v) = T(t)$ implies $T(s) = T(s[t/x^v])$.

Proof: By induction on t, s, and the length of v. □

Definition 3 *A* solution *of a set S of L^*-equations is a sequence $T = (T_0, \ldots, T_k)$ of substitutions $T_j : Var \to L$-terms such that $T(s) \equiv T(t)$ for each equation $s = t$ in S.*

The notions of *most general solution* and *semi-unifier* extend naturally to sets of L^*-equations.

Proposition 2 *1. For any instance S of the semi-unification problem there is a set S^* of L^*-equations such that S has a (most general) solution if and only if S^* has.*

2. For any set S^* of L^*-equations there is an instance S of the semi-unification problem such that S^* has a (most general) solution if and only if S has.

Proof: 1. Let S^* be $\{s^i = t \mid s \leq_i t \in S\}$. Obviously, T solves S if and only if it solves S^*. 2. First, replace $S^*(x_1^{v_1}, \ldots, x_n^{v_n})$ by

$$S^*[y_1/x_1^{v_1}, \ldots, y_n/x_n^{v_n}] \cup \{x_1^{v_1} = y_1, \ldots, x_n^{v_n} = y_n\}, \tag{1}$$

with fresh variables $y_1, \ldots, y_n$.[2] Then, repeatedly replace an equation $x^{vi} = y$, with $i \in I$ and $v \in I^*$, $v \neq \epsilon$, by the equations $x^v = z,\ z^i = y$, with fresh variable z. Finally, replace $z^i = y$ by an inequation $z \leq_i y$.

The resulting instance S of the semi-unification problem is solvable if and only if S^* is: If $T = (T_0, \ldots, T_k)$ is a solution of S^*, modify T_0 on the additional variables by putting $T_0(y_j) := T(x_j^{v_j})$ and $T_0(z) := T(x^v)$, respectively. Note that T is a solution of $x^{vi} = y$ iff $T(x^{vi}) =$

[2] If, as in Kfoury e.a.[8], an instance of the semi-unification problem must not contain equations between first-order terms, we replace $s = t$ by $z \leq_i s,\ z \leq_i t$, with some fresh variable z (obtaining an instance that is not left-linear).

$T(y)$ iff $T_i(T(x^v)) = T_0(y)$ iff, for the appropriate z and the modified T, $T(x^v) = T_0(z)$ and $T_i(T_0(z)) = T_0(y)$, iff T is a solution of $x^v = z,\ z \leq_i y$. It follows that T is a solution of S. Conversely, if T solves S then it solves the above system (1). By Proposition 1, ii), T then solves $S^*(x_1^{v_1}, \ldots, x_n^{v_n})$. □

Note that the unification problem asks for a system $S(x_1, \ldots, x_n)$ of L-equations whether the first-order sentence $\exists x_1 \ldots \exists x_n.\ \bigwedge\ S(x_1, \ldots, x_n)$ holds in the term algebra for L.

The semi-unification problem asks for a system $S^*(x_1, \ldots, x_n, i_1, \ldots, i_k)$ of $L \cup \{i_1, \ldots, i_k\}$-equations, whether the *second-order* condition

$$\exists i_1 \ldots i_k.(\bigwedge_j\ i_j \text{ is an } L\text{-endomorphism } \wedge \exists x_1 \ldots x_n.\bigwedge\ S^*(x_1, \ldots, x_n, i_1, \ldots, i_k))$$

holds in the term algebra for L. (If L is finite,"i is an L-endomorphism" is a first-order condition.) It seems natural to study the semi-unification problem over algebras other than the term algebra, but we cannot say to what extend this has been done by algebraists. Moreover, we are not aware of any work in second-order logic "L_{hom}", i.e. first-order logic over the language L extended by unary function variables that range over L-endomorphism only, or L-homomorphism, in the many-sorted case.

2.2 A Transformation Calculus for Semi-Unification Problems

To avoid discussing symmetry of equality, in the following we consider an *equation* between terms s and t of L^*, written as $s = t$, to be a multiset of two L^*-terms. An equation in a system S is *solved in* S, if it has the form $X^v = t$, where x^v occurs only once in S and, moreover, if t is a variabel yw, then $\mid v \mid \geq \mid w \mid$. In this case, we call s the *solved variable* of the equation, choosing one if both are solved. An occurrence of a variable x^v in S is *maximal*, if it is not inside some x^{vi}, $i \in I$. The *maximal variables of* S are those having a maximal occurrence in S.

Proposition 3 *The number of solved equations of a system is bounded by the number of its maximal variables.*

Proof: Each solved equation of S contains at least one solved variable, and hence a maximal variable with exactly one occurrence in S. □

Transformation Rules

For all L^*-terms $s,\ s_1, \ldots, s_m,\ t,\ t_1, \ldots, t_m$, and variables x^v, we use the following rules to transform systems:

Decomposition of terms
$S\ \dot\cup\ \{f(s_1, \ldots, s_m) = f(t_1, \ldots, t_m)\} \longrightarrow S \cup \{s_1 = t_1, \ldots, s_m = t_m\}$

Substitution of variables

$S \dot{\cup} \{x^v = t\} \longrightarrow S[t/x^v] \cup \{x^v = t\}$,
if $x^v \notin \text{free}(t)$, $x^v = t$ is not solved in $S \cup \{x^v = t\}$, and $\mid w \mid \leq \mid v \mid$ if $t \equiv y^w$.

Lemma 2 *If S_1 can be transformed into S_2, then the sets of solutions of S_1 and S_2 are the same.*

Proof: ¿By induction on the length of the transformation sequnece. If the last transformation is a term decomposition, the claim is obvious. If the last transformation is a substitution

$$S \dot{\cup} \{x^v = t\} \longrightarrow S[t/x^v] \cup \{x^v = t\},$$

then any solution T of $S \dot{\cup} \{x^v = t\}$ or $S[t/x^v] \cup \{x^v = t\}$ satisfies $T(x^v) = T(t)$, hence T solves S if and only if it solves $S[t/x^v]$, by Proposition 1. □

It is useful to admit further transformation rules, in order to remove trivial equations and to stop the transformation process in case an unsolvable equation is found:

Simplification Rules

Elimination of trivial equations

$S \cup \{x^v = x^v\} \longrightarrow S$

Function clash

$S \cup \{f(s_1, \ldots, s_m) = g(t_1, \ldots, t_n)\} \longrightarrow$ *fail*

Extended 'occurs check'

$S \cup \{x^v = t\} \longrightarrow$ *fail*, if $x^v \in \text{free}(t)$ and t is not a variable.

The occurs check is called *extended* because a variable x^v also "occurs" in the variable x^{vw}. Note that $x^v = f(x^{vw})$ cannot be solved, since $\mid T(x^v) \mid < \mid T(f(x^{vw})) \mid$ for any T.

Lemma 3 *[13] If S is irreducible under the Transformation and Simplification rules, then it has a most general solution. If S can be reduced to fail, it has no solution.*

In order to decide solvability of a system S, we want to transform it into an irreducible one. The above two lemmata establish partial correctness of this method. However, by the undecidability result of Kfoury e.a., there can be no reduction strategy which terminates on all instances S. Clearly, non-termination depends on the Transformation rules only.

Let us recall why the transformations ensure termination in the case of unification, when all variables have exponent ϵ. In this case, termination obviously follows from three facts:

- Term decompositions decrease the sum of the sizes of the equations in the system, while they do not decrease the number of solved equations.

- Substitutions of variables increase the number of solved equations[3], which is bounded by the number of variables occurring in the system.
- The number of variables is not increased by the transformations.

In the case of semi-unification, however, the second and third properties no longer hold:

- Substitutions of variables need not increase the number of solved equations, even for monadic languages:

$$\{\underline{x^{ii} = y},\ x^i = f(x)\} \longrightarrow \{\underline{f(x^i) = y},\ x^i = f(x)\}.$$

  The number of solved equations may even go down:

$$\{\underline{y^j = g(z)}, \underline{x^{ijk} = z}, x^i = f(y)\} \longrightarrow \{y^j = g(z), f(y^{jk}) = z, \underline{x^i = f(y)}\}$$

- The number of maximal L^*-variables may be increased by substitutions of variables:

$$\{\underline{x^i} = c,\ \underline{x} = f(\underline{y}, \underline{z})\} \longrightarrow \{f(\underline{y^i}, \underline{z^i}) = c,\ \underline{x} = f(\underline{y}, \underline{z})\}.$$

Thus, substitution transformations do not make a system simpler in an obvious sense. In the rest of this Section, we will show that any system can be transformed into one with *incomparable* dependent variables, where x^v and y^w are incomparable when $x \not\equiv y$ or none of v and w is a prefix of the other. In the next section we will impose an additional restriction on such systems which will imply that substitution sequences must terminate. Finally, we will then show that decomposition transformations can be allowed without reintroducing nontermination.

Let us first look at substitutions with equations between variables only.

Lemma 4 *Each system can be transformed into one where all equations between variables are trivial or solved, without increasing the size of the system.*

Proof: Assume that the claim is true for all systems whose size is smaller than that of S. Let $x^v = y^w \in S$ be an unsolved and nontrivial equation, and $S = S_0 \dot{\cup} \{x^v = y^w\}$. By symmetry, we can assume that $x^v \notin \text{free}(y^w)$.

Case 1: $\mid v \mid > \mid w \mid$. By $S_0 \cup \{x^v = y^w\} \longrightarrow S_0[y^w/x^v] \cup \{x^v = y^w\}$ we obtain a system S' which is strictly smaller than S, since x^v had at least two occurrences. By induction, S' can be transformed to a smaller system all of whose equations between variables are solved or trivial.

Case 2: $\mid v \mid = \mid w \mid =: n$. By induction on n, all equations between variables of size $< n$ are solved or trivial. The substitution $S_0 \cup \{x^v = y^w\} \longrightarrow S_0[y^w/x^v] \cup \{x^v = y^w\}$ turns $x^v = y^w$

[3] An equation $x = t$ is not solved before, but is solved after application of the substitution

$$S \cup \{x = t\} \longrightarrow S[t/x] \cup \{x = t\}.$$

In addition, if $z = s \in S$ is solved in $S \cup \{x = t\}$, with z as solved variable, then $z \not\equiv x$ and z is not free in t and s, hence $(z = s)[t/x] \equiv (z = s[t/x])$ is solved in $S[t/x] \cup \{x = t\}$. Note also that no two solved equations may collapse into one.

into a solved equation, with x^v as solved variable. Note that all solved variables of size $\leq n$ remain unchanged (which is not true for solved variables like x^{viw}), and that no new occurrences of these variables are created. Hence the number of solved equations between variables of size $\leq n$ does not go down. Since the number and size of equations remains unchanged, by iteration we can make all equations between variables of size $\leq n$ solved or trivial. Clearly, the size of the system does not grow. $\square$

Except for equations between compound terms, and equations between variables of the same size, each equation in a system has an asymmetry that we can use to define the *dependent* variable of the equation. It will be sufficient to define this notion for equations that could possibly be used to perform a substitution transformation. In the following, the dot-notation $\dot{s}$ is used to indicate that s is a *compound term*, that is, not in Var*. (A *compound variable* x^v, with nonempty v, is not considered a compound term.)

Definition 4 *The* dependent variable *of an equation $x^v = \dot{s}$ in S is x^v, if $x^v \notin free(\dot{s})$. The dependent variable of a nontrivial equation $x^v = y^w$ in S is the variable with larger size, if there is one; if both variables are of the same size, it is the solved variable (resp. one of the two solved variables) in case the equation is solved in S, and taken from $\{x^v, y^w\}$ by some choice function, otherwise. No further equation has a dependent variable. Two variables x^v and y^w are* incomparable, *if neither $x^v \in free(y^w)$ nor $y^w \in free(x^v)$.*

Lemma 5 *Each system can be transformed, by the substitution and the occurs check rules only, to fail or into a system S such that*

$\Delta(S)$ *each dependent variable of S has only one occurrence that is not inside a compound term.*

In particular, the dependent variables of any two equations in S are incomparable.

Proof: By induction on the number $n(S)$ of equations between variables. Suppose the claim is true for all systems S' with $n(S') < n(S)$. We may assume that all equations in S between variables are solved or trivial, which, by the proof of Lemma 4, can be achieved without increasing n. In particular, we can assume:

$\Phi(S)$ In S, each dependent variable of an equation between variables has only one occurrence that is not in a compound term.

$\Psi(S)$ S does not contain an equation $x^v = \dot{s}(\dot{x}^{vw})$.

If $\Psi(S)$ does not hold, S can be reduced to *fail* by the occurs check rule. Let $m(S)$ be the number of equations $x^v = \dot{s}$, such that $x^v \notin \text{free}(\dot{s})$ and x^v has another occurrence in S that is not inside a compound term.

Case 1: $m(S) = 0$. The claim is true by assumption $\Phi(S)$ on equations between variables.

Case 2: $m(S) > 0$. Let S be $S_0 \cup \{x^v = \dot{s}\}$, such that $x^v \notin \text{free}(\dot{s})$ and x^v has another occurrence in some left or right variable side of an equation in S_0. Apply the substitution

$$S = \quad S_0 \cup \{x^v = \dot{s}\} \longrightarrow S_0[\dot{s}/x^v] \cup \{x^v = \dot{s}\} \quad =: S'.$$

Subcase 2.1: $n(S') < n(S)$. The claim holds by induction on n.

Subcase 2.2: $n(S') = n(S)$. In S there must be an equation of the form $x^{vw} = \dot{t}$ different from $x^v = \dot{s}$. We will show that $m(S') < m(S)$ and S' satisfies $\Phi(S')$, whence the claim follows by induction on m.

Each occurrence of some x^{vu} in S_0 has turned into a compound term $\dot{s}^u$ in S'. Hence, a variable side of an equation in S either has not changed or has become a compound term. This implies, first, that x^v has only one occurrence in S' that is not in a compound term, and, second, that every dependent and every solved variable in S of an equation between variables has at most one occurrence in S' that is not inside a compound term.

By $n(S') = n(S)$, $\Phi(S)$ and $\Psi(S)$, each dependent variable in S' of an equation of the form $z^u = \dot{r}'$ is the dependent variable of an equation $z^u = \dot{r}$ in S. Hence we obtain $m(S') < m(S)$. If $\Psi(S')$ does not hold, we can reduce S' to *fail*.

We have not quite shown that S' satisfies $\Phi(S')$, because the dependent variables of S' are dependent variables of S except for two cases only: a) An equation in S between variables of the same size may be solved in S with *two* solved variables, but solved in S' with *one* solved variable. By our definition of dependent variables, the dependent variable of the equation in S and S' may be different. However, as this variable is solved in S', it does not violate $\Phi(S')$. b) An equation between variables may be unsolved in S', and if the variables are of the same size, there is freedom in choosing the dependent variable. We let the dependent variable of the equation in S' be the same as in S. It then follows that this also does not violate $\Phi(S')$. Thus, using an approptrioate choice function in defining dependent variables of S', Φ is inherited from S to S'. □

Example 2 *Let S_0 be $\{f(x, g(x, y)) \leq_1 f(h(y), z),\ g(z, h(x)) \leq_2 g(g(z, x), y)\}$. Transorming this according to Proposition 2 gives the following set of L^*-equations:*

$$S_1 = \{f(x^1, g(x^1, y^1)) = f(h(y), z),\ g(z^2, h(x^2)) = g(g(z, x), y)\}.$$

While $\Delta(S_1)$ trivially holds, we loose property Δ when applying decompositions, obtaining

$$S_2 = \{x^1 = h(y),\ z = g(x^1, y^1),\ z^2 = g(z, x),\ y = h(x^2)\},$$

with $n(S_2) = 0$, but $m(S_2) = 1$, as z and z^2 are comparable dependent variables. By Case 2 of the lemma, using $[g(x^1, y^1)/z]$, this reduces to

$$S_3 = \{x^1 = h(y),\ z = g(x^1, y^1),\ g(x^{12}, y^{12}) = g(g(x^1, y^1), x),\ y = h(x^2)\},$$

with $n(S_3) = n(S_2) = 0$ and $m(S_3) = 0$. Thus $\Delta(S_3)$ holds, and all dependent variables are incomparable.

3 Quasi-Monadic Semi-Unification

We have seen that in the case of the extended language L^*, termination of transformations is no longer obvious: substitution of variables may both increase the number of L^*-variables and decrease the number of solved equations in a system.

It is clear that in a monadic language, no substitution can increase the number of (maximal) variables in a system. However, we want to allow function symbols of arity > 1, as for the application to type inference at least one binary function symbol is needed. Thus, the next best restriction is that each term (in fact, 'each term being substituted' would be sufficient!) may have at most one maximal free variable. [4]

Before we go further into details, a word of motivation may be helpful. Kfoury e.a.[9] showed that the left-linear semi-unification problem is decidable. An instance S of the left-linear semi-unification problem is a system $\{s_i \leq_i t_i \mid i \in I\}$ of inequations (no equations!) between first-order terms, where each variable occurs at most once in each s_i. Suppose we transform such a system using a decomposition rule (working with inequations). By left-linearity, it can never happen that from decomposing $s_i \leq_i t_i$ we obtain two inequations with the same variable on its left hand side, such as

$$x \leq_i s \qquad \text{and} \qquad x \leq_i t. \tag{2}$$

Any solution $T = (T_0, T_i)$ of these two inequations would satisfy

$$T_0(s) = T_i(T_0(x)) = T_0(t),$$

so T_0 would be a unifier of s and t. When s or t contain many variables, this might lead to complicated dependencies between variables and substitutions T_j of any solution of the given instance. In particular, a unification of terms on the right hand side of $\leq_i$ may substitute a variable y on the left of some $\leq_j$ by a term with several occurrences of the same variable, thereby destroying left-linearity. Thus, the key effect of left-linearity is that unifications between terms on the right hand side of an inequality $\leq_i$ do not occur.

The main observation leading to the solvable class of semi-unification presented below was that *some* unifications between terms on the right hand side of inequations can be allowed without destroying decidability of solvability. We simply must be sure that no such unification will increase the number of (maximal) variables in the system.

Definition 5 *The quasi-monadic semi-unification problem is the semi-unification problem restricted to instances in which each term has at most one free maximal variable.*

For example, $\{x \leq_1 f(z, z),\ f(x, g(x, 0, x)) \leq_2 f(g(0, y, y), y)\}$ is quasi-monadic (when translated to L^*), or $\{g(x^2, f(x^2), x^2) = g(0, y^1, y^1)\}$. Clearly, quasi-monadic problems are closed under transformations, and these do not increase the number of maximal variables in a system:

Lemma 6 *If S_1 can be transformed to S_2 and S_1 is quasi-monadic, then S_2 is also quasi-monadic, and*

$$\mid maxvar(S_2) \mid \leq \mid maxvar(S_1) \mid.$$

Proof: By induction on the transformation. It is sufficient to consider $S_1 \longrightarrow S_2$ to be a substitution

$$S \cup \{x^v = t\} \longrightarrow S[t/x^v] \cup \{x^v = t\}.$$

[4] This restriction is so severe that we do not expect our decidability result by itself to have applications in deciding typability for polymorphic recursive functions.

In this case, since S_1 is quasi-monadic, we have $|\ \mathrm{maxvar}(t)\ | \leq 1$, hence for any w,

$$|\ \mathrm{maxvar}(x^{vw}[t/x^v])\ | = |\ \mathrm{maxvar}(t^w)\ | \leq 1 = |\ \mathrm{maxvar}(x^{vw})\ |.$$

From this the claim follow easily. □

Although, by the lemma, the number of maximal variables in transforming a system is bounded, the termination proof for unification still cannot simply be adapted to quasi-monadic semi-unification: since the number of solved equations (and solved variables) may be reduced by substitution transformations, we have to find a substitute for counting the number of solved equations in a system.

Before doing this, we will strengthen Lemma 5 for quasi-monadic systems. The problem is that a substitution with $y^w = f(x)$, say, may turn a solved equation $x^v = s$ into an unsolved one, since new occurrences of variables may arise, as in $y^{wv}[f(x)/y^w] = f(x^v)$. Hence performing a substitution will give rise to further substitutions in general. With quasi-monadic systems at least, we do not run into infinite sequences of substitutions:

Lemma 7 *Let S be a quasi-monadic system satisfying $\Delta(S)$. Every sequence of substitution transformations beginning in S finally produces a system S' that either can be turned to fail by the occurs check rule or is irreducible under substitutions.*

Proof: For each variable x^v, we define the set of exponents of x^v in S by

$$exp(x^v, S) := \{w \mid x^{vw} \text{ occurs in some compound term of } S\}.$$

Note that if $x^v = t$ is a solved equation of S, with x^v as solved variable, then $exp(x^v, S)$ is empty. We perform an induction on

$$e(S) := \sum\{|\ exp(x^v, S)\ | \mid x^v \text{ is a dependent variable of } S\}.$$

Case 1: $e(S) = 0$. Let $x^v = s$ be an equation of S, with x^v as dependent variable. By $\Delta(S)$ and $e(S) = 0$, there is only a single occurrence of x^v in S, which means that $x^v = s$ is solved and hence the substitution transformation using $[s/x^v]$ (or $[x^v/\mathrm{s}]$) cannot be applied. Thus S is irreducible under substitution transformations.

Case 2: $e(S) > 0$. We may assume that $\Psi(S)$ from the proof of Lemma 5 holds, since otherwise we can reduce S to *fail*. Let S' be obtained from S by performing the substitution

$$S := S_0 \cup \{x^v = s\} \longrightarrow S_0[s/x^v] \cup \{x^v = s\} =: S'.$$

Then $x^v \notin \mathrm{free}(s)$, and $exp(x^v, S) \neq \emptyset$. If s is a constant term, then clearly $exp(x^v, S') = \emptyset$ and $e(S') < e(S)$, using $\Psi(S)$ to ensure that no new dependent variables arise. By induction, every sequence of substitution transformations beginning in S' is finite. If s is not a constant term, s contains one maximal variable, y^w say, since S is quasi-monadic. We distinguish two cases.

Case 2 a: There is an equation $y^u = t \in S$, with dependent variable y^u, such that $u \leq w$. Let $(u\backslash w)$ stand for the suffix of w obtained by removing the prefix u, and $(u\backslash w)V = \{(u\backslash w)v \mid v \in V\}$ for any set V of words over I. By $\Delta(S)$, x^v and y^u are incomparable, and we obtain:

$$\begin{array}{rcll} exp(x^v, S') & = & \emptyset, & \\ exp(y^u, S') & = & exp(y^u, S) \ \cup \ (u\backslash w)\, exp(x^v, S), & \\ exp(z^r, S') & = & exp(z^r, S), & \text{for all other dependent variables } z^r. \end{array}$$

(We may assume that S and S' have the same dependent variables.) It follows that

i) $e(S') \leq e(S)$, and $x^v = s$ is solved in S', but unsolved in S.

Any variable occurrence in S' that was not in S is obtained from a substitution $x^{vr}[s(y^w)/x^v] = s(y^{wr})$. Since $y^u = t$ is an unsolved (and applicable !) equation of S and there are no incomparable dependent variables (in S), no solved equation of S has become an unsolved equation of S'. This shows that

ii) the number of solved equations in S' is greater than the number of solved equations in S.

As the number of equations is bounded, we can stay in Case 2 a) finitely often only.

Case 2 b: There is no equation $y^u = t \in S$, with dependent variable y^u, such that $u \leq w$. It is sufficient to show $e(S') < e(S)$.

Suppose $y^w \in \text{free}(x^v)$. Then $y \equiv x$ and $w < v$. For a dependent variable x^r with $w \leq r$,

$$exp(y^r, S') = \begin{cases} exp(y^r, S) \ \cup \ r\backslash(w \cdot exp(x^v, S)), & r \neq v \\ r\backslash(w \cdot exp(x^v, S)), & r = v \end{cases}.$$

For different such r, the $r\backslash(w \cdot exp(x^v, S))$ are pairwise disjoint strict subsets of $exp(x^v, S)$, because the x^r are incomparable, by $\Delta(S)$. Hence

$$\begin{array}{l} \sum\{\mid exp(x^r, S') \mid \mid x^r \text{ a dependent variable (of } S'),\ w \leq r\} \\ \quad < \ \sum\{\mid exp(x^r, S) \mid \mid x^r \text{ a dependent variable (of } S),\ w \leq r\}. \end{array}$$

As $exp(z^r, S') = exp(z^r, S)$ for all remaining dependent variables, we obtain $e(S') < e(S)$.

Suppose $y^w \notin \text{free}(x^v)$. Then, using $0 = \mid exp(x^v, S') \mid$ and the same disjointness argument as above, we obtain

$$\begin{array}{l} \mid exp(x^v, S') \mid + \sum\{\mid exp(y^r, S') \mid \mid y^r \text{ a dependent variable (of } S'),\ w \leq r\} \\ \quad < \ \mid exp(x^v, S) \mid + \sum\{\mid exp(y^r, S) \mid \mid y^r \text{ a dependent variable (of } S),\ w \leq r\}. \end{array}$$

Again, $exp(z^r, S') = exp(z^r, S)$ for all remaining dependent variables, and $e(S') < e(S)$. □

A good measure for a system irreducible under substitution transformations is how many solved equations $x^v = t$ with $\mid v \mid = n$ it has, for each n. The number of such equations for small n is more relevant than that for large n, because a solved equation $x^{vw} = s$ can get transformed by a substitution with some equation $x^v = t$, which for example may arise only after some decomposition of terms.

Formally, to each system S that is irreducible under substitutions of variables, we associate a function f_S that gives the number $f_S(n)$ of variables x^v solved in S such that $\mid v \mid = n$. If S is obtained from S_0 by a sequence of transformations, note that f_S is bounded by the constant function s whose value is the number M of maximal variables in S_0, by Lemma 6. Also note that there are at most M such n where $f_S(n) > 0$. This motivates the following definition:

Definition 6 *For* $s : \omega \to \omega$ *and* $M \in \omega$, *and* $\text{support}(f) := \{n \in \omega \mid f(n) \neq 0\}$, *define*

$$\begin{array}{rcl} F(s, M) & := & \{f : \omega \to \omega \mid \forall n. f(n) \leq s(n), \mid \text{support}(f) \mid < M\}, \qquad \textit{and} \\ f \prec g & :\Leftrightarrow & \exists n. (f(n) > g(n) \wedge \forall k < n. f(k) = g(k)). \end{array}$$

The function f_S generalizes the measure of counting solved equations or variables of S in the simpler case of unification. We omit the proof of the following observation, which depends essentially on the bounds s and M.

Lemma 8 *The relation* $\prec$ *is well-founded on* $F(s, M)$*, for each* s *and* M.

If a system S is transformed to S' by term decomposition, then clearly $f_{S'} \overset{\preceq}{=} f_S$. For substitutions of variables, this need not be true: the two equations for y^i and z^i in

$$\{u = r(x^{ii}),\ x^i = f(y, z),\ y^i = t,\ z^i = s\} \longrightarrow \{u = r(f(y^i, z^i)),\ x^i = f(y, z),\ y^i = t',\ z^i = s'\},$$

will become unsolved when we turn the equation for x^i into a solved one.

Suppose that S is irreducible under substitutions, a term decomposition is applied, and then S' is obtained by transforming the result to another system irreducible under substitutions. We want to show that with respect to the measure just sketched, the system S' is to a higher degree in solved form than S is. Since the maximal variables are bounded, by this we must reach a system which is irreducible.

Lemma 9 *Let* S *be a system that is irreducible under substitution of variables, and let* S' *be a system obtained from* S *by decompositions of terms. Let* S'' *be obtained by turning* S' *into a system irreducible under substitution transformations, according to Lemma 7. Then* $f_{S''} \prec f_S$.

Proof: (Idea only; explicitly in the final version) Let $x^v = s$ be an equation of S' that arose from the decomposion of an equation between compound terms of S. Suppose x^w is the dependent variable of an equation $x^w = t$ of S. Since S was substitution-irreducible, w is not a prefix of v. Hence either x^v is incomparable with x^w, or v is a *strict* prefix of w. Applying $[s/x^v]$ repeatedly to make $x^v = s$ a solved equation, we increase the number of solved equations with exponents of length $\mid v \mid$. Even if v is a prefix of w and thus $x^w = t$ will no longer belong to the solved equations - in particular, this may happen with several different such w simultaneously, thereby possibly reducing the number of solved equations - we have a gain according to $\prec$ in our measure f_S as defined above. □

Theorem 1 *The quasi-monadic semi-unification problem is decidable.*

Proof: This follows from the above lemmata, using the well-foundednes of $\prec$ on $F(s, M)$ with $s(n) = M$ for all n, and $M = \mid \text{maxvar}(S) \mid$, for each instance S. □

Example 3 Here is a (monadic) example that demonstrates Lemma **??**. The system

$$S_0 = \{x^i = f(y),\ y^j = g(z),\ h(x) = h(f(g(z^k)))\}$$

is irreducible under substitutions and satisfies Δ. By decomposition, we obtain

$$S_1 = \{x^i = f(y),\ y^j = g(z),\ x = f(g(z^k))\},$$

which, according to Lemma **??**, can be made irreducible under substitutions, giving

$$S_2 = \{f(g(z^{ki})) = f(y),\ y^j = g(z),\ x = f(g(z^k))\}.$$

Again, S_2 is irreducible under substitutions and satisfies Δ. Note that $f_{S_0}(0) = 0,\ f_{S_2}(1) = 1$ and hence $f_{S_2} \prec f_{S_0}$. Decomposing S_2 gives

$$S_3 = \{y = g(z^{ki}),\ y^j = g(z),\ x = f(g(z^k))\},$$

which reduces further to

$$S_4 = \{y = g(z^{ki}),\ g(z^{kij}) = g(z),\ x = f(g(z^k))\},$$

with $f_{S_4}(0) = 2,\ f_{S_4}(1) = f_{S_4}(2) = f_{S_4}(3) = 0$. By decomposition, we arrive at an irreducible, hence solvable, system

$$S_5 = \{y = g(z^{ki}),\ z^{kij} = z,\ x = f(g(z^k)\}.$$

Note that $f_{S_5}(0) = 2,\ f_{S_5}(1) = f_{S_5}(2) = 0$, but $f_{S_5}(3) = 1$, thus $f_{S_5} \prec f_{S_4}$.

References

[1] J. Dörre and B. Rounds: On Subsumption and Semiunification in Feature Algebras. Proc. 5th Annual IEEE Symposium on Logic in Computer Science, Pittsburgh 1990

[2] F.Henglein: Type Inference and Semi-Unification. Proc. ACM Conference on Lisp and Functional Programming, Snowbird, Utah, July 25-27, pp. 184-197, 1988.

[3] F.Henglein: Polymorphic Type Inference and Semi-Unification. PhD Thesis, Rutgers University, April 1989. Also available as: New York University, Dept. of Computer Science, Courant Institute of Mathematical Sciences, Tech.Report 443, May 1989

[4] F.Henglein: Fast Left-Linear Semi-Unification. Utrecht University, Dept. of Computer Science, Tech.Report RUU-CS-89-33, 1989. Also in: Proc. International Conference on Computing and Information, Niagara Falls, Canada, May 1990. Springer LNCS, to appear

[5] P.C.Kannelakis, J.C.Mitchell: Polymorphic unification and ML typing. Proc. 16th ACM Symposium on Principles of of Programming Languages, pp 105 - 115, January 1989

[6] D. Kapur and D. Musser and P. Narendran and J. Stillman: Semi-Unification. Proc. of the 8th Conference on Foundations of Software Technology and Theoretical Computer Science, Pune, India, December 21 - 23, 1988, pp. 435-454. Springer LNCS 338

[7] G.Kreisel, personal communication, Nov. 1989

[8] A.J.Kfoury, J.Tiuryn, and P.Urcyczyn: Computational Consequences and Partial Solutions of a Generalized Unification Problem. 4th IEEE Symposium on Logic in Computer Science, Asilomar, California, June 5-8, 1989

[9] A.J.Kfoury, J.Tiuryn, and P.Urcyczyn: The Undecidablity of the Semi-Unification Problem. Boston University, Dept. of Computer Science, Tech. Report, 1988. To appear in: Information and Computation.

[10] A.J.Kfoury, J.Tiuryn, and P.Urcyczyn: An Analysis of ML-Typability. 15th Colloquium on Trees in Algebra and Programming, CAAP'90, Arnold (ed.). Springer LNCS 431, pp. 206 - 220, 1990.

[11] H.Leiß, On type inference for object-oriented programming languages. E. Börger e.a. (eds.) CSL'87. Proc. of the 1st Workshop on Computer Science Logic, Universität Karlsruhe, 1987. Springer LNCS 329, pp. 151 - 172

[12] H.Leiß, Semi-unification and type inference for polymorphic recursion. Siemens AG, Technical Report INF2-ASE-5-89, München, May 1989

[13] H.Leiß: Polymorphic Recursion and Semi-Unification. E. Börger e.a. (eds.) CSL '89. Proc. of the 3rd Workshop on Computer Science Logic, Universität Kaiserslautern, Oct 2 - 6, 1989. Springer LNCS 440, pp. 211 - 224

[14] A. Mycroft: Polymorphic Type Schemes and Recursive Definitions. In: International Symposium on Programming, 6th Colloquium. Toulouse, April 17 - 19, 1984. Springer LNCS 187, pp. 217 - 228.

[15] P. Pudlák: On a Unification Problem related to Kreisel's Conjecture. Commentationes Mathematicae Universitatis Carolinae, no. 29 (3), pp. 551-556, 1988

[16] P. Purdom: Detecting Looping simplifications. In: P. Lescanne (ed.) Proc. 2nd Conference on Rewriting Techniques and Applications, Bordeaux, 1987. Springer LNCS 250, pp. 54 - 62

[17] W. Snyder and J. Gallier: Higher Order Unification Revisited: Complete Sets of Transformations. The Journal of Symbolic Computation 8, pp. 101 - 140, 1989

[18] G. Takeuti: Proof Theory. Second Edition. North-Holland, Amsterdam 1975.