Unpublished HC

Hen 88a

Algebraic Properties of Semi-Unification

Fritz Henglein*
Courant Institute of Mathematical Sciences
New York University
715 Broadway, 7th floor
New York, N.Y. 10012, USA
Internet: henglein@nyu.edu or henglein@paul.rutgers.edu

October 12, 1988

Abstract

Semi-unification is the problem of solving inequalities of the form $M_1 \leq M_2$ in the subsumption lattice of free first-order terms. Since this problem does not seem to have attracted general attention we give a comprehensive treatment of its basic algebraic properties and contrast it with unification, the problem of solving term equations. We show that, contrary to some statements in the literature and in contrast to term equations, term inequalities do not have unique most general solutions with respect to strong equivalence, the preferred notion of "renaming of variables". A natural, weaker notion of equivalence, however, admits unique most general solutions and, more generally, induces a complete lattice onto the set of all solutions of term inequalities. We provide two rewriting specifications of most general solutions, the second of which is uniformly terminating due to a novel extended occurs check.

1 Introduction

Unification and semi-unification deal with related problems. Unification addresses solving equations between free first-order terms while semi-unification tackles the question of solving inequalities of the form $M_1 \leq M_2$ between terms M_1 and M_2.[1] Here $\leq$ refers to the subsumption preordering on terms.

Whereas unification has innumerous well-known uses and applications, semi-unification and related problems have apparently only recently received attention. Inequality constraints in general [GRDR88], and semi-unification in particular [Cho86,Hen88], have been shown to be at the heart of type checking in implicitly typed polymorphic programming languages. Term inequalities have also been explored as a partial order theory for constraint logic programming [PM88] and, in general, as a form of "partial order programming" [Par89]. Recently term inequalities

*This research has been supported by the ONR under contract number N00014-85-K-0413.

[1] We find the prevalent terminology somewhat unfortunate. While there is a distinction between "equation" (something that is to be *solved*) and "equality" (something that *holds*), there is no corresponding distinction with "inequality" since the term "inequation" is not commonly used in the English language. Even worse, "inequality" gives no indication as to whether $\leq$ (*less-than-or-equal-to*) or $\neq$ (*not-equal-to*) is meant, and there is no standard linguistic mechanism for distinguishing between these two. The term "inequation" has popped up in the literature, but, since it is still uncommon, we will use "inequality" in this paper throughout. This also makes it possible, admittedly somewhat artificially, to distinguish our systems of equations and inequalities from the related, but different, systems of equations and inequations in [Col84] and [LMM87].

of the form $M_1 \neq M_2$ have been studied in a general setting [Col84,LMM87,MSK87]. We believe that, similarly, semi-unification is worthy of investigation on the basis of its fundamental character.

A correct treatment of the algebraic structure of semi-unifiers — solutions of term inequalities — is trickier than is apparent at first sight. This is evidenced by technically incorrect treatments and statements in the literature [PM88,Cho86]. In this paper we present some results on the algebraic structure of semi-unifiers and briefly report some complexity-theoretic results. Our main goal is to convince the reader that, in the same fashion in which *strong* equivalence classes of idempotent substitutions (see below for definitions) characterize the solutions of term equations and vice versa (see theorem 4), the *weak* equivalence classes of all substitutions characterize the solutions of term inequalities and vice versa (see theorem 5). In particular, we cannot replace "strong" by "weak" in this statement. Two substitutions σ_1 and σ_2 are strongly equivalent if there are substitutions α and α' such that $\alpha \circ \alpha' = \iota$, where ι denotes the identity substitution, and $\alpha \circ \sigma = \sigma'$. Strong equivalence is the preferred formalization the common phrase "equivalent up to renaming of variables". We will show that, unlike term equations, term inequalities do *not* have most general solutions that are unique modulo strong equivalence. The weaker notion of weak equivalence admits unique most general solutions.

After this introduction, in section 2, we describe terms and substitutions and their basic algebraic structure. Section 3 contains definitions of systems of equations and inequalities and their solutions, semi-unifiers, as well as some basic results. We compare the algebraic structure of unifiers, modulo strong equivalence, and the structure of semi-unifiers, modulo weak equivalence, in section 4. The following section, section 5, contains two rewriting specifications for computing most general semi-unifiers. Section 6 briefly reports some recent results on the relation of semi-unification with problems in type theory and computational complexity of nonuniform and uniform semi-unification. Finally, section 7 gives a brief summary.

2 The Algebraic Structure of Terms and Substitutions

In this section we define the objects of our universe of discourse, terms and substitutions, and investigate aspects of their algebraic structure. The material is mostly extracted from [Hue80], [Ede85], and [LMM87]; much of the material dates back to [Plo70a], [Plo70b], [Rey70], and [Hue76]. Some definitions and results are cast in new way. They are, though simple refinements of standard results, useful in later sections.

2.1 Basic Definitions

Definition 1 *(Variables, functors, constants, terms)*

Let V be an infinite denumerable set, F a nonempty denumberable set, and C a denumerable set disjoint from V. The set of (first-order) terms *$T(F,C,V)$ (or simply T whenever F, C, and V are understood) consists of all strings derivable from M in*

$$M ::= x|c|f(M,\ldots,M)$$

where f, c, and x range over F, C, and V, respectively. F, C, and V are called functors, constants, *and* variables, *respectively.*

The set of extended terms *$T(F,C,V)^\Omega$ (or simply T^Ω) is $T(F,C,V)$ with an additional distinguished element Ω called the* undefined term.

Two terms $M_1, M_2 \in T$ are *equal*, denoted $M_1 = M_2$, if and only if M_1 and M_2 are identical as strings; e. g., $f(x,y) = f(x,y)$, but $f(x,y) \neq f(u,v)$.[2] Ω is equal to itself and no other term.

Definition 2 *(Substitution)*
The set of (first-order) substitutions $\mathcal{S}(F,C,V)$ *(or simply* $\mathcal{S}$ *whenever* F, C, *and* V *are understood from the context) is a mapping from* V *to* T *that is the identity almost everywhere. Every substitution* σ *can be applied to extended terms by defining*

$$\begin{aligned} \sigma(\Omega) &= \Omega \\ \sigma(c) &= c, \text{ if } c \in C \\ \sigma(f(M_1,\ldots,M_n)) &= f(\sigma(M_1),\ldots,\sigma(M_n)). \end{aligned}$$

The domain $D(\sigma)$ *of* σ *is* $\{x \in V \mid \sigma(x) \neq x\}$. *The* canonical representation *of* σ *with* $D(\sigma) = \{x_1,\ldots,x_n\}$ *is* $\{x_1 \mapsto \sigma(x_1),\ldots,x_n \mapsto \sigma(x_n)\}$. $V(\sigma) = D(\sigma) \cup V(\sigma(D(\sigma)))$ *denotes the set of all variables that occur in the canonical representation of* σ.

The mapping ω, *which maps all extended terms to* Ω, *is called the* undefined substitution. *For any set of substitutions* Υ *we will write* Υ^ω *for* Υ *with the additional element* ω; $\mathcal{S}(F,C,V)^\omega$ *is called the set of* extended substitutions.[3]

A substitution specifies the simultaneous replacement of some set of variables by specified terms. For example, for $\sigma_0 = \{x \mapsto u, y \mapsto v, u \mapsto y, v \mapsto x\}$ we have $\sigma_0(f(x,y)) = f(u,v)$. The undefined substitution maps everything to the undefined term; e. g., $\omega(f(x,y)) = \Omega$ and $\omega(\Omega) = \Omega$.

The undefined term Ω and the undefined substitution ω are useful in providing a meaning for the dynamic notion of "failure" in unification and other applications. They also lead to a very satisfying algebraic structure of terms and substitutions (see theorems 1 and 3).

2.2 Term Subsumption

Definition 3 *(Subsumption, α-conversion)*
The preordering $\leq$ *of* subsumption[4] *on* T^Ω *is defined by*

$$M_1 \leq M_2 \Leftrightarrow (\exists \sigma \in S^\omega)\, \sigma(M_1) = M_2$$

for any $M_1, M_2 \in T^\Omega$.

The equivalence relation $\cong$ *of* α-conversion *on* T^Ω *is defined by*

$$M_1 \cong M_2 \Leftrightarrow M_1 \leq M_2 \wedge M_2 \leq M_1$$

for all $M_1, M_2 \in T^\Omega$. *We write* $M_1 < M_2$ *if* $M_1 \leq M_2$, *but* $M_1 \not\cong M_2$. *For any* $M \in T^\Omega$, $[M]$ *denotes the equivalence class of* M *in* T^Ω.

If $M_1 \leq M_2$ we say M_1 *subsumes* M_2; e. g., $f(x,y)$ subsumes $f(g(y),z)$ since for $\sigma_1 = \{x \mapsto g(y), y \mapsto z\}$ the equality $\sigma_1(f(x,y)) = f(g(y),z)$ holds. If $M_1 \cong M_2$ we say M_2 is an *α-variant* of M_1 and vice versa; e. g., $f(x,y)$ is an α-variant of $f(u,v)$.

Recall that a partial ordering on the set L is a *lattice* if it has a greatest lower bound and a least upper bound for every finite subset of L. It is a *complete* lattice if it has greatest lower

[2] We use the convention that identifiers starting with letters from the lower half of the alphabet denote functors and identifiers starting with letters from the upper half of the alphabet stand for variables.

[3] Note that ω is *not* a substitution from V to T^Ω since it is *not* the identity almost everywhere.

[4] Note that this definition follows [Hue80] and [Ede85], but is dual to the definition in [LMM87].

bounds and least upper bounds for all subsets of L, not just finite ones [MB79]. Recall also that a partial ordering is *Noetherian* if it has no infinite descending chains $M_1 > M_2 > \dots$ [Hue80].

The preordering $\leq$ on T^{Ω} induces a partial order on the quotient set $T^{\Omega}/_{\cong} = \{[M] \mid M \in T^{\Omega}\}$, which we will also denote by $\leq$. The structure of terms with respect to subsumption is captured in the following theorem.

Theorem 1 *1. $(T^{\Omega}/_{\cong}, \leq)$ is Noetherian.*

2. $(T^{\Omega}/_{\cong}, \leq)$ is a complete lattice.

Proof: See [Hue80].

The least upper bound of a set Θ of extended terms is called its *most general common instance*; its greatest lower bound is called its *most specific common anti-instance*. The theorem expresses that both most general common instance and most specific common anti-instance are unique modulo α-conversion. Finding the most general common instance of a pair of terms is a special case of the unification problem (disjoint variable case). Finding the most specific common anti-instance of a pair is the anti-unification problem [Hue76,LMM87]. A most general common instance of $\{f(x, g(y)), f(g(y), z)\}$ is $f(g(y), g(z))$, but also $f(g(u), g(v))$; a most specific common anti-instance is $f(s, t)$. Clearly, V is the least element and $\{\Omega\}$ is the greatest element in $T^{\Omega}/_{\cong}$.

2.3 Generality of Substitutions

For any subset W of V we will write $\sigma\mid_W$ for the substitution defined by

$$\sigma\mid_W(x) = \begin{cases} \sigma(x), & x \in W \\ x, & x \notin W \end{cases}$$

Definition 4 *(Generality preorder, strong equivalence)*

Let W be a subset of V. The preordering $\leq_W$ on S^{ω} over W is defined by

$$\sigma_1 \leq_W \sigma_2 \Leftrightarrow (\exists \rho \in S^{\omega})\, (\rho \circ \sigma)\mid_W = \sigma_2\mid_W .$$

The equivalence relation $\cong_W$ on S^{ω} over W is defined by

$$\sigma_1 \cong_W \sigma_2 \Leftrightarrow \sigma_1 \leq_W \sigma_2 \wedge \sigma_2 \leq_W \sigma_1 .$$

for all $\sigma_1, \sigma_2 \in S^{\omega}$. We write $\sigma_1 <_W \sigma_2$ if $\sigma_1 \leq_W \sigma_2$, but $\sigma_1 \not\cong_W \sigma_2$. For any $\sigma \in S^{\omega}$, $[\sigma]_W$ denotes the $\cong_W$-equivalence class of σ in S^{ω}.

If $\sigma_1 \leq_W \sigma_2$ we say that σ_1 is at least as general *as σ_2. The equivalence relation $\cong_V$ is called* strong equivalence.

If $\sigma_1, \sigma_2 \in S$, then $\sigma_1 \leq_V \sigma_2 \Leftrightarrow (\exists \rho \in S) \rho \circ \sigma_1 = \sigma_2$. Strong equivalence is the standard notion of "renaming" found in the literature [CL73,LMM87].

For a given subset W of V we can ask whether it is possible to construct a sequence of ever more and more general subsitutions from a given starting substitution $\sigma \in S$. The answer to this question is negative and is proved below.

Definition 5 *(degree)*

Let W be a subset of V; let σ be a substitution in S. Let the length *$l(M)$ denote the number of occurrences of elements from $F \cup V$ in M, for $M \in T$. We define the* degree *$d(\sigma, W)$ of σ over W as follows.*[5]

$$d(\sigma, W) = \max\{(\sum_{x \in W'} l(\sigma(x))) - |V(\sigma(W'))| : W' \subset W \wedge |W'| < \infty\}$$

[5]Here $\sigma(W')$ denotes the set $\{\sigma(x) : x \in W'\}$, and $V(\sigma(W'))$ stands for the set of all variables occurring in it.

Of course, we would have liked to define $d(\sigma, W)$ simply by $(\sum_{x \in W} l(\sigma(x))) - |V(\sigma(W))|$, as in [Ede85], but this definition would be ill-defined for infinite W's. It is easy to see that due to the finiteness of the domain of any substitution $d(\sigma, W)$ is well-defined, that is, $0 \leq d(\sigma, W) < \infty$, for any $W \subset V$ and $\sigma \in \mathcal{S}$.

As in the case of terms, the preordering $\leq_W$ induces a partial order on $\mathcal{S}^\omega/_{\cong_W} = \{[\sigma]_W \mid \sigma \in \mathcal{S}^\omega\}$, denoted also by $\leq_W$.

Theorem 2 *$(T/_{\cong_W}, \leq_W)$ is Noetherian for any $W \subset V$.*

This theorem establishes an analog to theorem 1, part 1, for *any* $W \subset V$. To prove it we establish a lemma first, which is a simple generalization of a similar lemma in [Hue80].

Lemma 1 *Let W be a subset of V; let σ_1, σ_2 be substitutions in $\mathcal{S}$. Then*

1. $\sigma_1 \cong_W \sigma_2 \Rightarrow d(\sigma_1, W) = d(\sigma_2, W)$
2. $\sigma_1 <_W \sigma_2 \Rightarrow d(\sigma_1, W) < d(\sigma_2, W)$

Proof: (Proof of lemma) [omitted for space reasons]

The proof of the theorem is now straightforward.

Proof: (Proof of theorem)
Assume there is a set of equivalence classes $\{E_i \mid i \geq 0\}$ such that $E_i >_W E_{i+1}$ for every $i \geq 0$. Let $\sigma_i \in E_i$ be arbitrary representatives of the E_i's for $i \geq 0$. By assumption we have $\sigma_i > \sigma_{i+1}$. If $\sigma_i = \omega$ for some i then $i = 0$ since there is no $\sigma \in \mathcal{S}^\omega$ such that $\sigma >_W \omega$; consequently $\sigma_i \in \mathcal{S}$ for $i \geq 1$. We know that $d(\sigma_1, W)$ is finite by definition of d. Lemma 1 asserts that for any $i \geq 1$ it must be that $d(\sigma_i, W) > d(\sigma_{i+1}, W)$. Consequently, there must be a σ_{i_0} with a negative degree, but this is impossible. Thus the assumption cannot hold, which proves that there are no infinite descending chains.

We will call a subset W of V *co-infinite* if $|V - W| = \infty$ and *co-finite* otherwise. A natural question is whether there is an analog to theorem 1, part 2; i.e., whether $\hat{S}_W$ forms a (complete) lattice under $\leq_W$ just as $(T^\Omega/_{\cong}, \leq)$ is a complete lattice. Interestingly, the analogue holds for all co-infinite W, but fails for all co-finite W in a major way: $\mathcal{S}^\omega/_{\cong_W}$ is neither an upper nor a lower semi-lattice under the partial order $\leq_W$. This shall be proved in the following two propositions.

Proposition 2 *For every co-finite subset W of V there is a pair of substitutions $\sigma_1, \sigma_2 \in \mathcal{S}$ with two minimal upper bounds $v_1, v_2 \in \mathcal{S}$ with respect to $\leq_W$ such that $v_1 \not\cong_W v_2$.*

Proof: Eder [Ede85] shows that the pair of substitutions

$$\{x \mapsto f(x, f(y, z)), y \mapsto f(x, f(y, z)), z \mapsto f(x, f(y, z))\}$$

and

$$\{x \mapsto f(f(x, y), z), y \mapsto f(f(x, y), z), z \mapsto f(f(x, y), z)\}$$

has an infinite set of minimal upper bounds, but no least upper bound with respect to $\leq_V$.

A simple generalization of Eder's pair will do the trick. Let W be a co-finite set. Without loss of generalization we can assume that $V - W = \{w_1, \ldots, w_n\}$ for some n and that $\{x_1, \ldots, x_{n+1}, y_1, \ldots, y_{n+1}, z_1, \ldots, z_{n+1}\}$ is a subset of W. Now with $\rho_i = \{x_i \mapsto f(x_i, f(y_i, z_i)), y_i \mapsto f(x_i, f(y_i, z_i)), z_i \mapsto f(x_i, f(y_i, z_i))\}$ and $\sigma_i = \{x_i \mapsto f(f(x_i, y_i), z_i), y_i \mapsto f(f(x_i, y_i), z_i), z_i \mapsto f(f(x_i, y_i), z_i)\}$ consider the substitutions $\rho = \bigcup_{i \in \{1,\ldots,n+1\}} \rho_i$ and $\sigma = \bigcup_{i \in \{1,\ldots,n+1\}} \sigma_i$.[6] The minimal upper bounds of ρ and σ are the substitutions

$$\begin{aligned} \bigcup_{i \in \{1,\ldots,n+1\}} \quad & \{x_i \mapsto f(f(s_i, t_i), f(u_i, v_i)), \\ & y_i \mapsto f(f(s_i, t_i), f(u_i, v_i)), \\ & z_i \mapsto f(f(s_i, t_i), f(u_i, v_i))\} \end{aligned}$$

for pairwise distinct variables $W = \{s_1, t_1, u_1, v_1, \ldots, s_{n+1}, t_{n+1}, u_{n+1}, v_{n+1}\}$. Consider one such minimal upper bound, say σ_1. Simple counting shows that there must be some variable $w \in W$ such that

$$w \in \{w_1, \ldots, w_n, x_1, \ldots, x_{n+1}, y_1, \ldots, y_{n+1}, z_1, \ldots, z_{n+1}\}.$$

Thus w is in W. If we consider another minimal upper bound, σ_2, with range variables

$$V(\sigma_2(\{x_1, \ldots, x_{n+1}, y_1, \ldots, y_{n+1}, z_1, \ldots, z_{n+1}\}))$$

disjoint from

$$V(\sigma_1(\{x_1, \ldots, x_{n+1}, y_1, \ldots, y_{n+1}, z_1, \ldots, z_{n+1}\})),$$

then it is clear that $\sigma_1 \not\leq_W \sigma_2$ because $w \notin V(\sigma_2(D(\sigma_2)))$.

This shows that $(\mathcal{S}^\omega/\cong_W, \leq_W)$ is not an upper semi-lattice. We can also show that it fails to be a lower semi-lattice.

Proposition 3 *For every co-finite subset W of V there is a pair of substitutions $\sigma_1, \sigma_2 \in \mathcal{S}$ with two maximal lower bounds $v_1, v_2 \in \mathcal{S}$ with respect to $\leq_W$ such that $v_1 \not\cong_W v_2$.*

Proof:

We shall only treat the case $W = V$. The general case is a generalization analogous to the previous proof.

Let $c_1, \ldots, c_4, d_1, \ldots, d_4$ be eight pairwise distinct constants and let f be an arbitrary functor. (The proof is along the same lines as here, only a little bit more cumbersome, if there are fewer than eight constants; in particular, it also works if there are no constants at all. Note that there must be at least one functor by definition.) Consider

$$\begin{aligned} \sigma_1 \quad = \quad & \{x_1 \mapsto f(f(c_1, c_2), f(c_3, c_4)), \\ & x_2 \mapsto f(f(c_1, c_2), f(c_3, c_4)), \\ & x_3 \mapsto f(f(c_1, c_2), f(c_3, c_4))\} \end{aligned}$$

[6]More formally, $\rho = \rho_1 \circ \ldots \circ \rho_{n+1}$ and $\sigma = \sigma_1 \circ \ldots \circ \sigma_{n+1}$. Since the order of composition is insignificant the informal set union operation on the canonical representations of the ρ_i's and σ_i's is well-defined.

and

$$\begin{aligned}\sigma_2 &= \{x_1 \mapsto f(f(d_1,d_2), f(d_3,d_4)),\\ &\quad\; x_2 \mapsto f(f(d_1,d_2), f(d_3,d_4)),\\ &\quad\; x_3 \mapsto f(f(d_1,d_2), f(d_3,d_4))\}.\end{aligned}$$

Both

$$\begin{aligned}\upsilon_1 &= \{x_1 \mapsto f(x_1, f(x_2,x_3)),\\ &\quad\; x_2 \mapsto f(x_1, f(x_2,x_3)),\\ &\quad\; x_3 \mapsto f(x_1, f(x_2,x_3))\}\end{aligned}$$

and

$$\begin{aligned}\upsilon_2 &= \{x_1 \mapsto f(f(x_1,x_2),x_3),\\ &\quad\; x_2 \mapsto f(f(x_1,x_2),x_3),\\ &\quad\; x_3 \mapsto f(f(x_1,x_2),x_3)\}\end{aligned}$$

are maximal lower bounds since, somewhat unexpectedly,

$$\{x_1, x_2, x_3 \mapsto f(f(y_1,y_2), f(y_3,y_4))\}$$

does not form a lower bound of σ_1 or σ_2 for any variables $x_1, \ldots, x_4$. Clearly, υ_1 and υ_2 are not equivalent under $\cong_V$.

The reason for this "misbehavior" of $(\mathcal{S}^\omega/_{\cong_W}, \leq_W)$ for co-finite W is due to the fact that we cannot "hide" enough variables from "consideration" under $\leq_W$. For subsets W of V that leave "enough" variables hidden in $V - W$ — for co-infinite W's — the partial orders $(\mathcal{S}^\omega/_{\cong_W}, \leq_W)$ have indeed a lattice structure. The proof of this is a consequence of the more general corollary 5 proved in section 4.

Theorem 3 *Let W be any subset of V. The following statements are equivalent.*

- $(\mathcal{S}^\omega/_{\cong_W}, \leq_W)$ *is a complete lattice.*
- W *is co-infinite; that is,* $|V - W| = \infty$.

3 Term Inequalities and Semi-Unifiers

In this section we present basic definitions and properties of inequalities over the subsumption preordering of terms.

Definition 6 *(System of equations and inequalities, nonuniform/uniform semi-unifier, unifier)*
A system S *of* equations and inequalities (SEI) *is a pair* $(\mathcal{E}, \mathcal{I})$ *where each of* $\mathcal{E}$ *and* $\mathcal{I}$ *consists of a set of pairs of terms from* T *written in the form*[7]

$$\left\{\begin{array}{ccc} M_{11} & = & M_{12} \\ M_{21} & = & M_{22} \\ & \cdots & \\ M_{m1} & = & M_{m2} \end{array}\right\} \mathcal{E}$$

$$\left\{\begin{array}{ccc} N_{11} & \leq & N_{12} \\ N_{21} & \leq & N_{22} \\ & \cdots & \\ N_{n1} & \leq & N_{n2} \end{array}\right\} \mathcal{I}$$

An extended substitution σ *for which there exist extended substitutions*[8] $\rho_1, \ldots, \rho_n$ *such that*[9]

$$\begin{array}{ccc} \sigma(M_{11}) & = & \sigma(M_{12}) \\ \sigma(M_{21}) & = & \sigma(M_{22}) \\ & \cdots & \\ \sigma(M_{m1}) & = & \sigma(M_{m2}) \end{array} \quad (\mathcal{E})$$

$$\begin{array}{ccc} \rho_1(\sigma(N_{11})) & = & \sigma(N_{12}) \\ \rho_2(\sigma(N_{21})) & = & \sigma(N_{22}) \\ & \cdots & \\ \rho_n(\sigma(N_{n1})) & = & \sigma(N_{n2}) \end{array} \quad (\mathcal{I})$$

holds simultaneously is called a (nonuniform) semi-unifier *of* S. *If* $\rho_1 = \rho_2 = \ldots = \rho_n = \rho$ *for some* ρ, *then* σ *is called a* uniform semi-unifier, *and if furthermore* $\rho = \iota$, *the identity substitution, then* σ *is called a* unifier.

S *is* solvable *if it has a semi-unifier other than* ω.

A semi-unifier, in other words, is a solution to a given set of equations and inequalities. A uniform semi-unifier additionally solves the inequalities in a "uniform" fashion[10], and a unifier solves the inequalities by making both sides equal. By definition, if an SEI has a unifier it has a uniform semi-unifier, and if it has a uniform semi-unifier it has a semi-unifier.

For any SEI S, $SU(S)$ is its set of semi-unifiers, $USU(S)$ its uniform semi-unifiers, and $U(S)$ its unifiers. Clearly, for unifiers there is no need to distinguish between equations and inequalities, and we can view, in this case, an SEI $S = (\mathcal{E}, \mathcal{I})$ as a system of equations alone made up of $\mathcal{E} \cup \mathcal{I}$.

It is well known that a set of equations can be expressed by a single equation in the sense that the set of its solutions (unifiers) is identical to the set of solutions of the original set of equations. An analogous result, with the same simple proof, holds for *uniform* semi-unifiers, but apparently not for *nonuniform* semi-unifiers.

Proposition 4 *For every SEI* S *there are SEI's* S' *and* S'' *such that*

1. S' *consists of at most one equation (and no inequality) and* $U(S) = U(S')(= USU(S') = SU(S'))$.

[7] Note that the symbols $=$ and $\leq$ here are only *formal*, not logical symbols as in the definition of term equality and subsumption.

[8] It is actually irrelevant whether ω is permitted amongst the ρ_i or not.

[9] Here the symbols $=$ and $\leq$ denote their logical meanings.

[10] Note that $\{x \leq c_1, x \leq c_2\}$ has a semi-unifier — the identity substitution ι — but no uniform semi-unifier.

2. *S'' consists of at most one equation and one inequality and* $USU(S) = USU(S'')(= SU(S''))$.

Proof: For (1) form term M_1 by tupling all the left-hand sides of S, and M_2 by tupling all the right-hand sides. Define $S' = \{M_1 = M_2\}$. For (2) proceed by tupling both sides of equations and inequalities separately.

A more precise characterization of this property can be achieved if we restrict ourselves to *ranked functors*, that is, functors with a fixed arity. In this case it can be shown that the proposition above holds true if and only if there exists at least one functor with arity greater than or equal to 2 in F. The case where no such functor exists is algebraically and computationally much simpler. It is treated in [Cho86] under the name *prefix inequalities.* As a consequence of the above proposition we could restrict ourselves to single equation/inequality combinations. However, multiple equations and inequalities come in handy in rewriting specifications (section 5) for computing semi-unifiers.

The following proposition is easily proved.

Proposition 5 *Let S be any SEI. For all W such that $V(S) \subset W \subset V$ and substitutions σ_1 and σ_2 such that $\sigma_1 \cong_W \sigma_2$ we have*

1. $\sigma_1 \in U(S) \Leftrightarrow \sigma_2 \in U(S)$
2. $\sigma_1 \in USU(S) \Leftrightarrow \sigma_2 \in USU(S)$
3. $\sigma_1 \in SU(S) \Leftrightarrow \sigma_2 \in SU(S)$

Thus the solutions of any SEI S are closed with respect to equivalence relation $\cong_W$ as long as W contains at least all variables occurring in S, and every unifier/uniform semi-unifier/semi-unifier can viewed as (a representative) of a whole equivalence class of solutions.

4 The Structure of Semi-Unifiers

It is often quoted that most general unifiers are unique "up to renaming of variables". As pointed out in [LMM87] there are several *distinct* notions of what this innocuous-looking little phrase can be taken to mean. The most commonly used notion is strong equivalence (i.e., equivalence modulo $\cong_V$). While different notions lead to a slightly different structure of unifiers for a given system of equations, they all admit the existence of most general unifiers (though most general unfiers with respect to one notion (e.g., [SS86]) are not necessarily most general with respect to another equivalence).

The fact that there are most general unifiers under any of the different notions of renaming may have prompted Chou to write that, similarly, "it is evident" that the most general semi-unifier of an SEI is unique modulo strong equivalence, if it exists at all [Cho86, page 11]. The breakdown in the analogy of the structure of $T/_{\simeq}$ and $S/_{\simeq_V}$ (see theorem 3 and the discussion before it), however, already suggests that this claim may not be true in general, and, indeed, it is incorrect.[11] A weaker notion of equivalence (see, e.g. [SS86, chapter 4]), however, admits the existence of most general semi-unifiers and an equivalent to the main structure theorem for unifiers.

[11] We feel tempted to say that, in view of theorem 3, uniqueness of most general unifiers with respect to strong equivalence is a "lucky coincidence"; or, less dramatically, a very specific property of unification that cannot simply be "transferred" to other problems; or, in more neutral terms, an outgrowth of the fact that the theory of unifiers can be viewed as a representation theory for idempotent substitutions, which indeed form a lattice with respect to $\leq_V$ [Ede85, theorem 4.9].

4.1 Strong Equivalence

Strong equivalence, $\cong_V$, corresponds to renaming of substitutions by composition of permutation substitutions; i.e., by substitutions α for which there is α^{-1} such that $\alpha \circ \alpha^{-1} = \alpha^{-1} \circ \alpha = \iota$. Two substitutions σ_1 and σ_2 are strongly equivalent if and only if there is such a permutation substitution α such that $\alpha \circ \sigma_1 = \sigma_2$. Strong equivalence has attracted a lot of attention because of its close connection to idempotent substitutions, which in turn are strongly related to systems of equations.

In this subsection the terms "minimal" and "most general" always refer to $\leq_V$.

4.1.1 Strong Equivalence and Idempotent Substitutions

We recapitulate the most important result on the structure of unifiers modulo strong equivalence from [Ede85] (see also [LMM87]). Note that every SEI has a minimal unifier .[12] This follows immediately from theorem 2. We call a minimal unifier σ of S a *most general* unifier of S if for all unifiers υ of S there is a substitution ρ such that $\rho \circ \sigma = \upsilon$.

A substitution σ is *idempotent* if it satisfies $\sigma \circ \sigma = \sigma$. The significance of idempotent substitutions and their relation to unification is summarized in the main structure theorem of idempotent substitutions.

Theorem 4 *Let $\mathcal{I} \subset \mathcal{S}$ denote the set of all idempotent substitutions.*

1. *Every system of equations S has a most general unifier that is idempotent, and for every idempotent substitution σ there is a system of equations S' such that σ is a most general unifier of S' (with respect to $\leq_V$).*
2. *$((\mathcal{I}^\omega \cap U(S))/_{\cong_V}, \leq_V)$ is a complete lattice for every system of equations S.*

Proof: By refinement of the proof of theorem 4.9 in [Ede85].

Since there are substitutions that are not strongly equivalent to any idempotent substitution, we have as a consequence of part 4 of this theorem that there are substitutions in $\mathcal{S}$ that are not most general unifiers. For example, $\{z_1 \mapsto f(z_1), \ldots, z_n \mapsto f(z_n)\}$ is not strongly equivalent to any idempotent substitution.

Part 4 expresses not only that every system of equations has a most general unifier, but that there is always an idempotent most general substitution. An instance of the theorem is Eder's structure theorem for idempotent substitutions.

Corollary 6 *Let S be any system of equations. Let $\mathcal{I}$ denote the set of all idempotent substitutions (without ω). Then*
$(\mathcal{I}^\omega/_{\cong_V}, \leq_V)$ is a complete lattice.

Proof: Consider $S = \{\}$ in theorem 4

4.1.2 Strong Equivalence and Semi-Unifiers

The set of idempotent unifiers of any system of equations forms a lattice. The fact that every system of equations has an idempotent most general unifier justifies in some sense the restriction of consideration to idempotent substitutions and unifiers, as is done from the outset in [Rob79].

In this subsection we show that idempotent substitutions and strong equivalence fail to capture the structure of semi-unifiers in a major way; namely,

[12] A unifier σ of an SEI S is minimal if for every other unifier σ' of S it holds that $\sigma' \leq \sigma \Rightarrow \sigma \leq \sigma'$.

1. for any SEI S neither $U(S)$ nor $USU(S)$ nor $SU(S)$ induce a lower or upper semi-lattice (under $\leq_V$).
2. there are systems of equations and inequalities that have a most general semi-unifier, but no idempotent one;
3. there are systems of equations and inequalities with no most general semi-unifier;

Proposition 7 *Neither one of* $(U(S)/_{\simeq_V}, \leq_V)$, $(USU(S)/_{\simeq_V}, \leq_V)$, *and* $(SU(S)/_{\simeq_V}, \leq_V)$ *forms a lower or upper semi-lattice for any SEI* S.

Proof: Almost directly from the proofs of propositions 2 and 3.

Proposition 8 *There is an infinite family of SEI's* S *such that* S *has uniform and nonuniform minimal semi-unifiers* σ_{i1} *and* σ_{i2}, *but* $\sigma_{i1} \not\cong_V \sigma_{i2}$.

Proof:
Consider $S_i = \{f(x_1, \ldots, x_i) \leq y\}$. The substitutions $\sigma_{i1} = \{y \mapsto f(u_1, \ldots, u_i)\}$ and $\sigma_{i2} = \{y \mapsto f(v_1, \ldots, v_i)\}$ are minimal semi-unifiers of S_i since the only for $\rho = \{\}$ we have $\rho <_V \sigma_{i1}$ or $\rho <_V \sigma_{i2}$ and $\{\}$ is not a semi-unifier of G_i. But there is no substitution $\alpha \in S$ such that $\alpha \circ \sigma_{i1} = \sigma_{i2}$ or $\alpha \circ \sigma_{i2} = \sigma_{i1}$.

Proposition 9 *There is an infinite family of SEI's* S *such that* S *has a most general uniform and nonuniform semi-unifier, but no idempotent one.*

Proof:
Consider $S_i = \{f(y_1) \leq z_1, \ldots, f(y_i) \leq z_i\}$. The substitution

$$\sigma_i = \{z_1 \mapsto f(z_1), \ldots, z_i \mapsto f(z_i)\}$$

and its $\cong_V$-equivalent substitutions are the only most general uniform and nonuniform unifiers of S_i. As we remarked earlier there is no idempotent substitution amongst them.

The reason why $\mathcal{S}^\omega, U(S), USU(S), SU(S)$ fail to be lattices under $\leq_V$ are intuitively rather pathological and should cast some doubt on the appropriateness of choosing strong equivalence as the "proper" notion of renaming on substitutions for semi-unification.

4.2 Weak Equivalence

In this section we define an equivalence relation on substitutions relative to systems of equations and inequalities that is properly weaker than strong equivalence. We will show that this relation, weak equivalence, ties general substitutions and systems of equations and inequalities together just as strong equivalence ties idempotent substitutions and systems of equations together (theorem 4).

Definition 7 *(Weak equivalence)*
Substitutions σ_1 *and* σ_2 *are called* weakly equivalent *with respect to SEI* S *(or simply* S-equivalent *if* $\sigma_1 \cong_{V(S)} \sigma_2$ *where* $V(S)$ *denotes the set of variables occurring in* S.

A k-ary *context* is a term $C \in T(F, C, V \cup MV)$ where MV is a k-element set $\{y_1, \ldots, y_k\}$ of *meta-variables* disjoint from V and C. For subsitution $\sigma : V \cup MV \mapsto T(F, C, V), \sigma = \{y_1 \mapsto M_1, \ldots, y_k \mapsto M_k\}$ the result of applying σ to C is denoted by $C[M_1, \ldots, M_k]$.

Lemma 10 *There is an operation* $\wedge : T \times T \mapsto T$ *such that*

1. $[M \wedge N] = [M] \wedge [N]$ *for all* $M, N \in T$.
2. $C[M_1, \ldots, M_k] \wedge C[M'_1, \ldots, M'_k] = C[M_1 \wedge M'_1, \ldots, M_k \wedge M'_k]$ *for all* k*,* k*-ary contexts* C*, and terms* $M_1, \ldots, M_k$ *and* $M'_1, \ldots, M'_k$.

Proof:

ad (1): See [Hue76]; see also [LMM87].

ad (2): Huet's definition of $\wedge$ has the property that $f(M) \wedge f(N) = f(M \wedge N)$ for every functor f. The result follows by structural induction on C.

For every operation that satisfies lemma 10, part 1, the following proposition holds.

Proposition 11 *For all terms* $M_1, M_2, N_1, N_2 \in T$ *such that* $M_1 \leq M_2$ *and* $N_1 \leq N_2$ *it holds that* $M_1 \wedge N_1 \leq M_2 \wedge N_2$.

For any SEI S, we call a (uniform) semi-unifier σ of S a *most general* (uniform) semi-unifier of S if for all (uniform) semi-unifiers υ of S there is a substitution ρ such that $(\rho \circ \sigma) \mid_{V(S)} = \upsilon \mid_{V(S)}$. Similarly, from now on a unifier of S will be called most general if it is minimum with respect to $\leq_{V(S)}$ instead of $\leq_V$ as in the previous section.

Now we are ready to prove the main theorem of this section.

Theorem 5 1. *Every system of equations and inequalities* S *has a most general (uniform) semi-unifier, and for every substitution* σ *there is a system of equations and inequalities* S' *such that* σ *is a most general (uniform) semi-unifier of* S.

2. $(SU(S)/_{\cong_{V(S)}}, \leq_{V(S)})$ *(as well as* $(USU(S)/_{\cong_{V(S)}}, \leq_{V(S)})$*) is a complete lattice for every system of equations and inequalities* S.

As an immediate consequence we have

Corollary 12 *Every solvable SEI* S *has a most general idempotent semi-unifier.*

Proof: (Proof of corollary)

Take a most general semi-unifier σ of S. If $V(S) = \{x_1, \ldots, x_n\}$ define $\sigma' = \{x_1 \mapsto x'_1, \ldots, x_n \mapsto x'_n\}$ where $x'_1, \ldots, x'_n$ are pairwise distinct variables not occurring in S. Then σ' is idempotent and a most general semi-unifier of S.

The theorem can be strengthened and still holds if we replace $\cong_{V(S)}$ (weak equivalence) and $\leq_{V(S)}$ by $\cong_W$ and $\leq_W$, respectively, where W is any co-infinite subset of V containing $V(S)$. With $S = \{\}$ and part 2 of this strengthened version we obtain the missing part of the proof of theorem 3.

Proof: (Proof of theorem)

For part 2, since every complete semi-lattice is automatically a complete lattice and since every Noetherian lower semi-lattice is a complete lower semi-lattice, it is sufficient to show that $(SU(S)/_{\cong_{V(S)}}, \leq_{V(S)})$ is a lower semi-lattice.

Let σ_1 and σ_2 be semi-unifiers of S. Let $x_1, \ldots, x_k$ be the set $V(S)$ of variables occurring in S. Denote $\sigma_1(x_i)$ by M_i and $\sigma_2(x_i)$ by N_i for $1 \leq i \leq k$. Now define $\sigma = \{x_1 \mapsto M_1 \wedge N_1, \ldots, x_k \mapsto M_k \wedge N_k\}$ with $\wedge$ defined as in lemma 10.

First we show that σ is a semi-unifier of S. Without loss of generality (see proof of proposition 4) we can assume that S consists of one equation and n inequalities. There are contexts $C_0, C_1, \ldots, C_n$ and $C'_0, C'_1, \ldots, C'_n$ such that S is equal to

$$\{\ C_0[x_1, \ldots, x_k] \ = \ C'_0[x_1, \ldots, x_k]\ \} \qquad \text{(equation)}$$

$$\left\{ \begin{array}{rcl} C_1[x_1, \ldots, x_k] & \leq & C'_1[x_1, \ldots, x_k] \\ & \cdots & \\ C_n[x_1, \ldots, x_k] & \leq & C'_n[x_1, \ldots, x_k] \end{array} \right\} \qquad \text{(inequalities)}$$

By assumption both σ_1 and σ_2 are semi-unifiers of S, and so

$$\begin{array}{rcl} C_0[M_1, \ldots, M_k] & = & C'_0[M_1, \ldots, M_k] \\ C_1[M_1, \ldots, M_k] & \leq & C'_1[M_1, \ldots, M_k] \\ & \cdots & \\ C_n[M_1, \ldots, M_k] & \leq & C'_n[M_1, \ldots, M_k] \end{array}$$

holds as well as

$$\begin{array}{rcl} C_0[N_1, \ldots, N_k] & = & C'_0[N_1, \ldots, N_k] \\ C_1[N_1, \ldots, N_k] & \leq & C'_1[N_1, \ldots, N_k] \\ & \cdots & \\ C_n[N_1, \ldots, N_k] & \leq & C'_n[N_1, \ldots, N_k] \end{array}$$

By proposition 11 this implies that

$$\begin{array}{rcl} C_0[M_1, \ldots, M_k] \wedge C_0[N_1, \ldots, N_k] & = & C'_0[M_1, \ldots, M_k] \wedge C'_0[N_1, \ldots, N_k] \\ C_1[M_1, \ldots, M_k] \wedge C_1[N_1, \ldots, N_k] & \leq & C'_1[M_1, \ldots, M_k] \wedge C'_1[N_1, \ldots, N_k] \\ & \cdots & \\ C_n[M_1, \ldots, M_k] \wedge C_n[N_1, \ldots, N_k] & \leq & C'_n[M_1, \ldots, M_k] \wedge C'_n[N_1, \ldots, N_k] \end{array}$$

holds, and by lemma 10, part 2, we conclude that

$$\begin{array}{rcl} C_0[M_1 \wedge N_1, \ldots, M_k \wedge N_k] & = & C'_0[M_1 \wedge N_1, \ldots, M_k \wedge N_k] \\ C_1[M_1 \wedge N_1, \ldots, M_k \wedge N_k] & \leq & C'_1[M_1 \wedge N_1, \ldots, M_k \wedge N_k] \\ & \cdots & \\ C_n[M_1 \wedge N_1, \ldots, M_k \wedge N_k] & \leq & C'_n[M_1 \wedge N_1, \ldots, M_k \wedge N_k] \end{array}$$

holds true. This, in turn, shows that σ is a semi-unifier of S.

We now show that any other semi-unifier σ' that is a lower bound of both σ_1 and σ_2 is also a lower bound of σ. Define $\sigma'(x_i) = L_i$ for $1 \leq i \leq k$. Since σ' is a lower bound of σ_1 (with respect to $\leq_{V(S)}$) it holds that $[L_1, \ldots, L_k] \leq [M_1, \ldots, M_k]$ for some arbitrary functor $[\ldots]$ written in infix-notation; similarly, $[L_1, \ldots, L_k] \leq [N_1, \ldots, N_k]$. Consequently, $[L_1, \ldots, L_k] \leq [M_1, \ldots, M_k] \wedge [N_1, \ldots, N_k]$ and, by lemma 10, part 2, $[L_1, \ldots, L_k] \leq [M_1 \wedge N_1, \ldots, M_k \wedge N_k]$; i.e., there is a substitution ρ such that

$\rho([L_1, \ldots, L_k] = [M_1 \wedge N_1, \ldots, M_k \wedge N_k]$. But this immediately implies $\rho(\sigma'(x_i)) = \sigma(x_i)$ for $1 \leq i \leq k$, and thus $\sigma' \leq_{V(S)} \sigma$.

For part 1, part 2 shows that every system of equations and inequalities has a most general semi-unifier. Conversely, let σ be an arbitrary substitution. If $\sigma = \omega$ then clearly σ is a most general semi-unifier of $\{f(x) = x\}$. If $\sigma = \{x_1 \mapsto M_1, \ldots, x_k \mapsto M_k\}$ let $\alpha = \{x_1 \mapsto x'_1, \ldots, x_k \mapsto x'_k\}$ where $x'_1, \ldots, x'_k$ are pairwise distinct variables disjoint from $x_1, \ldots, x_k$. Now define $S = \{\alpha(M_1) \leq x_1, \ldots, \alpha(M_k) \leq x_k\}$. Clearly, σ is a most general semi-unifier of S.

There are more constructive proofs of the uniqueness of most general semi-unifiers modulo weak equivalence, but they neither yield the powerful structure theorem 5 nor do they lead to a uniformly terminating algorithm for computing most general semi-unifiers. A pair of rewriting specifications for computing most general semi-unifiers are given in the following section.

5 Specification of Most General Semi-Unifiers

In this section we present basic, implementable rewriting specifications for most general semi-unifiers. The first is a natural and straightforward extension of the rewriting specification for most general unifiers from [Her30], which was expounded by Martelli and Montanary and used as the starting point for the development of efficient unification algorithms [MM82]. This system is in general, though, nonterminating. The second rewriting specification refines the first one by adding an "extended" occurs check. It can be shown that there is an effective rewriting strategy for the second specification that leads to uniform termination of rewritings.

5.1 The Naive Rewriting Specification

For technical reasons we consider the special string "Unsolvable SEI" an SEI from now on that has only semi-unifier, namely ω. The first specification, given in figure 1 is straightforward, and similar versions can be found in the literature (see [Cho86]). This rewriting system preserves semi-unifiers in a sense that we shall make precise below.

Definition 8 *Let $\Rightarrow$ be a reduction relation on systems of equations and inequalities.*

1. *The relation $\Rightarrow$ is* sound *if for every S, S' such that $S \Rightarrow S'$ and for every semi-unifier σ' of S' there is a semi-unifier σ of S such that $\sigma\,|_{V(S)} = \sigma'\,|_{V(S)}$ (and thus $\sigma \cong_{V(S)} \sigma'$).*
2. *The relation $\Rightarrow$ is* complete *if for every S, S' such that $S \Rightarrow S'$ and for every semi-unifier σ of S there is a semi-unifier σ' of S' such that $\sigma\,|_{V(S)} = \sigma'\,|_{V(S)}$ (and thus $\sigma \cong_{V(S)} \sigma'$).*

Informally and imprecisely speaking, soundness expresses that a reduction step does not add semi-unifiers, and completeness means that no semi-unifiers are lost in a reduction step.

Proposition 13 *The reduction relation defined by the naive rewriting system (in figure 1) is sound and complete.*

Proof: Induction on the number of rewriting steps.

Any SEI S is in *normal form* with respect to a reduction relation $\Rightarrow$ if there is no S' such that $S \Rightarrow S'$. If an SEI is in normal form with respect to the naive rewriting system or the canonical rewriting system below it is easy to extract a most general semi-unifier from it.

Given an SEI S with k inequalities we initially tag all the inequality symbols with distinct "colors" $1, \ldots, k$ indicated by superscripts of the inequality symbol; e.g., $\leq^{(1)}$. Then non-deterministically choose an equation or inequality and take a rewriting action depending on its form.[a] (The string "Unsolvable SEI" is a special symbol not occurring in any term.)

1. $f(M_1, \ldots, M_k) = f(N_1, \ldots, N_k)$:

   Replace by the equations $M_1 = N_1, \ldots, M_k = M_k$.

2. $f(M_1, \ldots, M_k) = g(N_1, \ldots, N_l)$ where f and g are distinct functors:

   Replace current SEI by "Unsolvable SEI" (functor clash).

3. $f(M_1, \ldots, M_k) = x$:

   Replace by $x = f(M_1, \ldots, M_m)$.

4. $x = f(M_1, \ldots, M_k)$ where x occurs in at least one of $M_1, \ldots, M_k$:

   Replace current SEI by "Unsolvable SEI" (occurs check).

5. $x = f(M_1, \ldots, M_k)$ where x does not occur in $M_1, \ldots, M_k$, but occurs in another equation or inequality:

   Replace x by $f(M_1, \ldots, M_k)$ in all other equations or inequalities.

6. $x = x$:

   Delete it.

7. $f(M_1, \ldots, M_k) \leq^{(i)} f(N_1, \ldots, N_k)$:

   Replace by inequalities $M_1 \leq^{(i)} N_1, \ldots, M_k \leq^{(i)} M_k$.

8. $x \leq^{(i)} M$ and $x \leq^{(i)} N$:

   Delete one of the two inequalities and add the equation $M = N$.

9. $f(M_1, \ldots, M_k) \leq^{(i)} x$:

   Add the equation $x = f(x'_1, \ldots, x'_k)$ where $x'_1, \ldots, x'_k$ are new variables not occurring anywhere else.

[a] Without loss of generality we restrict ourselves to the case where $C = \{\}$.

Figure 1: Naive rewriting specification

Proposition 14 *Let S be a system of equations and inequalities in normal form with respect to the reduction relation defined by the naive (canonical) rewriting system in figure 1.*

If $S = \{x_1 = M_1, \ldots, x_k = M_k, y_1 \leq N_1, \ldots, y_l \leq N_l\}$ then the substitution $\sigma = \{x_1 \mapsto M_1, \ldots, x_k \mapsto M_k\}$ is a most general idempotent semi-unifier of S.

Proof: By inspection

To determine a most general semi-unifier of an SEI S we can apply the naive rewriting system to it and, if it terminates in a normal form S' we can extract a most general semi-unifier of S'. If $S' =$ "Unsolvable SEI" then S is unsolvable; otherwise there is a most general semi-unifier σ' of S' according to proposition 14. As a result of proposition 13 the restriction $\sigma' |_{V(S)}$ (or σ' itself) is a most general semi-unifier of S.

5.2 The Canonical Rewriting Specification

There are systems of equations and inequalities for which there is no finite rewriting derivation in the naive rewriting system; that is, no sequence of rewriting steps such that after a finite number of steps no more rewritings are possible. Consider, for example, the system $S_0 = \{f(x, g(y)) \leq f(y, x)\}$. It is easy to see that there is always at least one rule applicable.

The main reason for nontermination is that the last inequality rule, rule (9), introduces new variables every time it is executed. Replacing it with the deceivingly pleasing rule

$f(M_1, \ldots, M_k) \leq x$:
Add the equation $x = f(M_1, \ldots, M_k)$.

would indeed eliminate the nontermination problem of rewriting derivations, but also its completeness. To see this, consider, for example, the system $S_1 = \{f(g(y), g(y)) \leq f(x, f(f(y)))\}$. There is a derivation that would lead us to claim, incorrectly, that S_1 has no semi-unifiers.

If we reconsider system S_0 it is easy to see that it is unsolvable. This is due to the fact that the inequalities

$$\begin{aligned} g(y) &\leq x \\ x &\leq y \end{aligned}$$

are not uniformly solvable. If we denote the length of a term M by $|M|$, then any solution M_1 for x and M_2 for y would have to satisfy the numeric inequalities $|M_1| \leq |M_2|$ and $|M_1| \geq |g(M_2)| \geq |M_2| + 1$, which is clearly impossible. We can catch this case by refining rule (9) with an "extended" occurs check. More precisely, let us call the rewriting system with rule (9) replaced by the rules in figure 2 the *canonical* rewriting system.

Proposition 15 *The reduction relation defined by the rewriting system in figure 1 with rule (9) replaced by the rules (9.1) and (9.2) from figure 2 is sound and complete.*

Proof: See discussion of system S_0.

For any rewriting system we will call any uniformly terminating algorithm that, given an input, picks a rewrite rule to be executed, an *effective rewriting strategy.*

Even though there are still infinite rewriting derivations possible in the canonical rewriting system we have the following theorem.

(9.1) $f(M_1, \ldots, M_k) \leq^{(i_0)} x$ and there are variables $x_0, \ldots, x_n$ such that $x = x_0$, $x_i \leq^{(j_i)} x_{i+1}$ are inequalities in the current SEI for $0 \leq i \leq n-1$ and some colors $i_1, \ldots, i_{n-1}$, and there exists an i such that x_n occurs in M_i:

Replace current SEI by "Unsolvable SEI" (extended occurs check).

(9.2) $f(M_1, \ldots, M_k) \leq^{(i_0)} x$ and there is no sequence of variables $x_0, \ldots, x_n$ such that $x = x_0$, $x_i \leq^{(j_i)} x_{i+1}$ are inequalities in the current SEI for $0 \leq i \leq n-1$ and some colors $i_1, \ldots, i_{n-1}$, and x_n occurs in some M_i:

Add the equation $x = f(x'_1, \ldots, x'_k)$ where $x'_1, \ldots, x'_k$ are new variables not occurring anywhere else.

Figure 2: Extended occurs check

Lemma 16 *There exists an effective rewriting strategy for the canonical rewriting system such that the strategy admits only finite rewriting derivations.*

In fact any strategy that executes rule (9.2) only if there are no other rules applicable satisfies this lemma.

The proof of this lemma is nontrivial and is omitted here. Precise complexity-theoretic characterizations of nonuniform and uniform semi-unification and related polymorphic type inference problems as well as "structurally" optimal concrete algorithms can be found in a separate article (see also section 6). An immediate consequence of this lemma is the decidability of semi-unification.

Theorem 6 *The set of all solvable systems of equations and inequalities is decidable.*

We have implemented a concise functional program for computing most general semi-unifiers in SETL [SDDS86]. This specification has already appeared in [Hen88].

6 Other Results

Our interest in semi-unification stems from its close connection to parametric polymorphic type inference. The programming language ML [Har86] is an implicitly typed polymorphic programming language with a polymorphic typing rule for **let**-bindings (thus sometimes also called **let**-polymorphism). While the type inference problem for its functional core had been thought to be theoretically and practically feasible, recently Kanellakis and Mitchell showed that it is PSPACE-hard [KM89]. An extension of ML's typing system with a polymorphic typing rule for possibly nested recursive definitions, which we call the Milner-Mycroft Calculus, was studied by Mycroft [Myc84] and, in a restricted form, by Meertens [Mee83]. Kfoury, Tiuryn, and Urzycyn devised a complicated, essentially nonconstructive method to show decidability of type inference in this system [KTU88]. Their proof, however, was later retracted.

We have recently been able to prove the following results in collaboration with Ken Perry, which will appear in a forthcoming paper [HP88].

1. The following problems are log-space equivalent:

    - Type inference in the Milner-Mycroft Calculus
    - Type inference in the Milner-Mycroft Calculus restricted to instances with only one outermost recursive definition and no **let**-bindings and no nested recursive definitions

- Nonuniform semi-unification

2. Nonuniform semi-unification can be computed in doubly exponential time.

3. Uniform semi-unification can be computed in polynomial space.

The reduction of the Milner-Mycroft Calculus to semi-unification was already published in [Hen88]. Somewhat surprisingly, the log-space equivalence of the Milner-Mycroft Calculus with a highly restricted version of it expresses that, in terms of type inference, nesting of **let**-bindings and recursive definitions is no harder than a single recursive definition.

7 Conclusion

In this paper we have presented semi-unification, which is the problem of solving equations and subsumption inequalities of the form $M_1 \leq M_2$ between first-order terms. We have pointed out that systems of equations and inequalities do not permit unique most general solutions modulo strong equivalence. However, the solutions of any system of equations and inequalities form a complete lattice with respect to a weaker form of equivalence (and corresponding partial order), called weak equivalence here; this is in analogy to the main structure theorem for unifiers. We have given a straightforward rewriting specification for computing most general solutions; and we have presented a refined version that guarantees uniform termination. The exact connection of semi-unification with type inference and recent complexity-theoretic results are only briefly mentioned.

References

[Cho86] C. Chou. *Relaxation Processes: Theory, Case Studies and Applications.* Master's thesis, UCLA, February 1986. technical report CSD-860057.

[CL73] C. Chang and R. Lee. *Symbolic Logic and Mechanical Theorem Proving.* Academic Press, New York and London, 1973.

[Col84] A. Colmerauer. Equations and inequations on finite and infinite trees. In *Proc. Int'l Conf. on Fifth Generation Computer Systems*, 1984.

[Ede85] E. Eder. Properties of substitutions and unifications. *J. Symbolic Computation*, 1:31–46, 1985.

[GRDR88] P. Giannini and S. Ronchi Della Rocca. Characterization of typings in polymorphic type discipline. In *Proc. Symp. on Logic in Computer Sciene*, pages 61–70, IEEE, Computer Society, Computer Society Press, June 1988.

[Har86] R. Harper. *Introduction to Standard ML (Preliminary Draft).* Technical Report, University of Edinburgh, February 1986.

[Hen88] F. Henglein. Type inference and semi-unification. In *Proc. ACM Conf. on LISP and Functional Programming*, ACM, ACM Press, July 1988.

[Her30] J. Herbrand. *Recherches sur la theorie de la demonstration.* PhD thesis, Universite de Paris, 1930. In: Ecrits logiques de Jacques Herbrand, Paris, PUF, 1968.

[HP88] F. Henglein and K. Perry. The computational complexity of semi-unification. 1988. forthcoming.

[Hue76] G. Huet. *R'esolution d''equations dans des langages d'ordre 1, 2, ldots, omega.* PhD thesis, Univ. Paris VII, Sept. 1976.

[Hue80] G. Huet. Confluent reductions: abstract properties and applications to term rewriting systems. *J. Assoc. Comput. Mach.*, 27(4):797–821, Oct. 1980.

[KM89] P. Kanellakis and J. Mitchell. Polymorphic unification and ML typing (extended abstract). In *Proc. 16th Annual ACM Symp. on Principles of Programming Languages*, ACM, January 1989.

[KTU88] A. Kfoury, J. Tiuryn, and P. Urzyczyn. A proper extension of ML with an effective type-assignment. In *Proc. 15th Annual ACM Symp. on Principles of Programming Languages*, pages 58–69, ACM, ACM Press, Jan. 1988.

[LMM87] J. Lassez, M. Maher, and K. Marriott. Unification revisited. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, Morgan Kauffman, 1987.

[MB79] S. MacLane and G. Birkhoff. *Algebra*. Macmillan, 1979. 2nd edition.

[Mee83] L. Meertens. Incremental polymorphic type checking in B. In *Proc. 10th ACM POPL*, pages 265–275, 1983.

[MM82] A. Martelli and U. Montanari. An efficient unification algorithm. *ACM TOPLAS*, 4(2):258–282, Apr. 1982.

[MSK87] C. Mohan, M. Srivas, and D. Kapur. Reasoning in systems of equations and inequations. In *Proc. 7th Conf. on Foundations of Software Technology and Theoretical Computer Science*, pages 305–325, Springer-Verlag, Dec. 1987. Lecture Notes in Computer Science, Vol. 287.

[Myc84] A. Mycroft. Polymorphic type schemes and recursive definitions. In *Proc. 6th Int. Conf. on Programming, LNCS 167*, 1984.

[Par89] D. Parker. Partial order programming. In *Proc. 16th Annual ACM Symp. on Principles of Programming Languages*, ACM, ACM Press, January 1989.

[Plo70a] G. Plotkin. *Lattice-Theoretic Properties of Subsumption.* Technical Report MIP-R77, Univ. of Edinburgh, 1970.

[Plo70b] G. Plotkin. A note on inductive generalization. *Machine Intelligence*, 5:153–163, 1970.

[PM88] D. Parker and R. Muntz. A theory of directed logic programs and streams. In *Proc. Int'l Conf. on Logic Programming*, pages 620–650, August 1988.

[Rey70] J. Reynolds. Transformational systems and the algebraic structure of atomic formulas. *Machine Intelligence*, 5:135–152, 1970.

[Rob79] J. Robinson. *Logic: Form and Function – The Mechanization of Deductive Reasoning.* North-Holland, New York, 1979.

[SDDS86] J. Schwartz, R. Dewar, E. Dubinsky, and E. Schonberg. *Programming with Sets: An Introduction to SETL.* Springer-Verlag, 1986.

[SS86] L. Sterling and E. Shapiro. *The Art of PROLOG.* MIT Press, 1986.