Semi-Unification

Fritz Henglein
Courant Institute of Mathematical Sciences
New York University
715 Broadway, 7th floor
New York, N.Y. 10003
Internet: henglein@nyu.edu

April 6th, 1988

Abstract

Semi-unification is the problem of solving inequalities of the form $\tau_1 \leq \tau_2$ in the subsumption lattice of (free) terms. Since this problem does not seem to have attracted much attention despite its fundamental character (we know of no reference that addresses this problem — possibly because of lack of obvious applications) we give a comprehensive introductory treatment and contrast it with unification, which is the corresponding problem of solving term equations. We prove a structure theorem showing the existence of most general semi-unifiers analogous to the structure theorem for most general unifiers in unification theory and present several nonoptimal algorithms for computing most general semi-unifiers. The main theorem shows that the *uniform* semi-unfication problem is decidable, and we conjecture that the algorithm terminates also for *nonuniform* semi-unification problems. Finally we present a partial arithmetization of the uniform semi-unification problem that leads to a speed-up over the original algorithm. Semi-unification is of utility for type inference in parametric polymorphic type systems. In particular, a proof of termination of our algorithm for nonuniform semi-unication problems implies decidability of the typability problem in the Milner-Mycroft Calculus, an open problem simultaneously raised by Mycroft [Myc84] and Meertens [Mee83].

1 Introduction

Unification and semi-unification deal with related problems. Unification addresses solving equations between terms with variables while semi-unification tackles the question of solving inequations of the form $\tau_1 \leq \tau_2$ between terms τ_1 and τ_2.[1] Here $\leq$ refers to the subsumption preordering on terms. Whereas unification has innumerous applications semi-unification seems to have ducked investigative scrutiny, possibly because there have been no apparent applications for it.[2]

[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. Since the term "inequation" has popped up in several published articles, we will use it in this paper, too, in order to be able to distinguish between inequations and inequalities in analogy to equations and equalities.

[2] We are not aware of any treatises on this problem and would appreciate hearing of relevant references.

In this paper we present some structural and computational results on semi-unification and point out an application of semi-unification in type theory, which actually led to this work. First, we give a brief background on unification and its many-fold applications. In section 5, we describe terms and substitutions and their algebraic structure. Section **??** contains some basic results and examples of semi-unification. We contrast the algebraic structure of unifiers and semi-unifiers in section **??**. The following section, section **??**, presents a structure theorem for most general semi-unifiers that is analogous to the structure theorem for most general unifiers in unification. Section **??** contains several algorithms for computing most general semi-unifiers for a quite standard graph-theoretic representation of terms and substitutions (cf. [PW78]). A partial arithmetization of the problem in section **??** remedies some computational bottlenecks and leads to a speed-up for uniform semi-unification problems. Semi-unification is at the heart of the type inference problem for the Milner-Mycroft Calculus [Myc84]. This is briefly treated in section **??**, although we refer to [Hen88] for a comprehensive treatment of this problem.[3] Finally, section **??** gives a brief summary and outlook to related problems that we think deserve further study.

[3] Since [Hen88] refers back to this paper in its treatment of semi-unification, both [Hen88] and this paper should be consulted for a complete treatment of the type inference problem for the Milner-Mycroft Calculus. We have made sure that there are no problems with circular references between these papers.

2 Work on Unification and Semi-Unification

3 Work on Unification

Unification is the problem (and informally also the process) of finding solutions to term equations of the form $\tau_1 = \tau_2$ where $\tau_1, \tau_2 \in T$. A solution of $\tau_1 = \tau_2$ is a substitution σ such that $\sigma(\tau_1) \equiv \sigma(\tau_2)$.

Although Herbrand [Her68] and Prawitz [Pra60] had already used unification algorithms, the utility of and interest in unification was essentially initiated by Robinson's novel resolution principle in theorem proving [Rob65] at the heart of which was a unification algorithm.

Since then papers on unification as well as applications of unification have abounded. While Robinson's original algorithm took exponential time to compute the solutions, new representations and algorithms have been found (see, e. g., [PW78] and [MM82]) that achieve linear bounds on the computation time, and the unification problem has been found to be P-complete [Sta88]. Universal unification theory addresses the problem of unification in term algebras that are subject to equational [Sie84] or conditional-equational [Hus85] laws such as associativity, commutativity, and idempotence. Several unification algorithms (e. g., citeLSSU79, [Bue86], [Sti81], or see [Sie84]) for such term algebras have been presented. Kapur and Narendran [KN86] showed that most of these unification problems are inherently hard, though. Huet [Hue75, Hue76] investigated higher-order unification and proved that it is recursively undecidable.

Unification has permeated the field of resolution-based and even non-resolution-based theorem proving [Ble77]. With the identification of a subset of First Order Logic that is especially amenable to resolution theorem proving (Horn Clause Logic, c. f. [Kow79]) unification plays an eminent role in logic programming languages such as Planner [Hew71] and PROLOG [WPP77, SS86].

A concise and clean treatment of the algebraic aspects of unification can be found in [LMM86] or in [Ede85].

4 Work on Semi-Unification

Semi-unification addresses the problem of solving inequalities of the form $\tau_1 \preceq \tau_2$ where $\tau_1, \tau_2 \in T$. A substitution σ is a solution to $\tau_1 \preceq \tau_2$ if there exists $\rho \in S$ such that $\rho(\sigma(\tau_1)) = \sigma(\tau_2)$.

Whereas reasoning with equalities and $\neq$-inequalities is drawing more and more attention (c. f. [Col84, LMM86, MSK87]), $\leq$-inequalities in the subsumption lattice of terms don't seem to have attracted much attention, probably because of a lack of apparent applications in other theoretical or practical areas so far. We have found that semi-unification is at the heart of the type inference problem in the Milner-Mycroft Calculus [Myc84, KTU88], but we are unaware of other treatments or applications of semi-unification.

5 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 [LMM86]; much of the material dates back to [Plo70a], [Plo70b], [Rey70], and [Hue76]. Some definitions and results are new. They are not deep, but extremely useful for later sections.

Definition 1 *Let V be a countably infinite set, called* variables*; let F be another countable (finite or infinite) set disjoint from V, called* function symbols *or* functors*; and let $a : F \to \mathcal{N}$ be an* arity *function from F into the natural numbers (including 0).*

The set $T(V, F, a)$ of terms *is defined inductively. It is the smallest set closed with respect to the following rules.*[4]

- *Every variable $v \in V$ is a term.*
- *Every functor $c \in F$ with arity $a(f) = 0$ is a term.*
- *If functor $f \in F$ has arity $a(f) = n$, with $n \geq 1$, and $t_1, \ldots, t_n \in T$ are terms, then the string $f(t_1, \ldots, t_n)$ is a term.*

Functors with arity 0 are also called constants*. For $\tau \in T, \Theta \subset T$, $V(\tau)$ denotes the set of variables $v \in V$ that occur in τ and $V(\Theta)$ is $\bigcup\{V(\tau) \mid \tau \in \Theta\}$.*

Terms are simply uninterpreted expressions: they denote themselves; that is, two terms τ_1 and τ_2 are equal if and only if they are syntactically identical. Term equality is denoted by $\equiv$; i. e., $f(x, y) \equiv f(x, y)$, but $f(x, y) \not\equiv f(u, v)$.[5]

Definition 2 *A* substitution *is a mapping $\sigma : V \to T$ from V to T, with $\sigma(x) \neq x$ for only a finite number of elements $x \in V$. We define the* domain *$D(\sigma) = \{x \in V \mid \sigma(x) \neq x\}$. A substitution σ is extended to T by defining recursively*

$$\sigma(f(t_1, \ldots, t_n)) = f(\sigma(t_1), \ldots, \sigma(t_n))$$

and

$$\sigma(c) = c.$$

For $\Theta \subset T$ we define $\sigma(\Theta) = \{\sigma(\tau) \mid \tau \in \Theta\}$. The canonical representation *of σ with $D(\sigma) = \{x_1, \ldots, x_n\}$ is $\{x_1 \leftarrow \sigma(x_1), \ldots, x_n \leftarrow \sigma(x_n)\}$. $V(\sigma) = D(\sigma) \cup V(\sigma(D(\sigma)))$ is the set of all variables that occur in the canonical representation of σ.*

A substitution σ is idempotent *if $\sigma \circ \sigma = \sigma$. The set of all substitutions is denoted by S; the set of idempotent substitutions by I.*

A substitution specifies the simultaneous replacement of some set of variables by specified terms. For example, for $\sigma_0 = \{x \leftarrow u, y \leftarrow v, u \leftarrow y, v \leftarrow x\}$ we have $\sigma_0(f(x, y)) \equiv f(u, v)$.

We can phrase the notions of term and substitution in the terminology of universal algebra [Gra77] by saying T is the free a-graded F-algebra generated by V, and S is the set of F-morphisms with finite basis.

[4] Whenever V, F, and a are understood from the context will simply write T instead of $T(V, F, a)$.

[5] 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.

Definition 3 *The preordering $\leq$ of* subsumption[6] *in T is defined by*

$$\tau_1 \leq \tau_2 \Leftrightarrow (\exists \sigma \in S)\sigma(\tau_1) = \tau_2$$

for any $\tau_1, \tau_2 \in T$.

The equivalence relation $\cong$ of α-conversion *in T is defined by*

$$\tau_1 \cong \tau_2 \Leftrightarrow \tau_1 \leq \tau_2 \wedge \tau_2 \leq \tau_1.$$

For any $\tau \in T$, $[\tau]$ denotes the equivalence class of τ in T.

If $\tau_1 \leq \tau_2$ we say τ_1 *subsumes* τ_2; e. g., $f(x,y)$ subsumes $f(g(y),z)$ since for $\sigma_1 = \{x \leftarrow g(y), y \leftarrow z\}$ the equality $\sigma_1(f(x,y)) \equiv f(g(y),z)$ holds. If $\tau_1 \cong \tau_2$ we say τ_2 is an *α-variant* of τ_1 and vice versa; e. g., $f(x,y)$ is an α-variant of $f(u,v)$.

Definition 4 *A substitution $\pi \in S$ is a* permutation *if it is a bijection of V into V. We say $\alpha \in S$ is* injective *on $W \subset V$ if $\sigma(W) \subset V$ and $\sigma(x_1) = \sigma(x_2) \Rightarrow x_1 = x_2$ for $x_1, x_2 \in W$.*

Note that the permutations are exactly those substitutions that are injective on any $W \subset V$. It is also easy to see that a substitution σ is a permutation exactly when its domain $D(\sigma)$ and its range $V(\sigma(D(\sigma)))$ are identical and it is injective on $D(\sigma)$.

The following proposition is easy to prove.

Proposition 1 *Let $\tau_1, \tau_2 \in T$. The following statements are equivalent.*

- *$\tau_1 \cong \tau_2$.*
- *There exists a permutation π such that $\pi(\tau_1) = \tau_2$.*
- *There exists a substitution α injective on $V(\tau_1)$ such that $\alpha(\tau_1) = \tau_2$.*

We can view permutations as substitutions that are "uniformly" injective for all $W \subset V$ whereas there are other substitutions that are injective on some, but not all subsets of V. For example, $\{x \leftarrow y, y \leftarrow x\}$ is a permutation, but $\{x \leftarrow y\}$ is not; it is injective on W if and only if $W \subset V - \{x\}$ or $W \subset V - \{y\}$. The above proposition shows that we can characterize α-conversion in T both in terms of permutations and the weaker notion of injective substitutions. We will see later that permutations and injective substitutions lead to different algebraic structures in S.

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 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 $\tau_1 > \tau_2 > \ldots$ [Hue80]. The structure of terms with respect to subsumption is captured in the following theorem.

Theorem 1 *Let $\hat{T}$ be the quotient set $T/_{\cong} = \{[\tau] \mid \tau \in T\}$ with an additional maximum element $\top$; let $\leq$ denote also the partial ordering in $\hat{T}$ canonically induced by the preordering $\leq$ in T. Then*

- *$(\hat{T}, \leq)$ is Noetherian.*
- *$(\hat{T}, \leq)$ is a complete lattice.*

[6] Note that this definition follows [Hue80] and [Ede85], but is dual to the definition in [LMM86].

Proof
See [Hue80].

The least upper bound of a set Θ of 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 a unification problem (disjoint variable case). Finding the most specific common anti-instance of a pair is the anti-unification problem [Hue76, LMM86]. 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 $f(s, t)$.

Definition 5 *Let Φ be a subset of V. The preordering $\leq_\Phi$ in S over Φ is defined by*

$$\sigma_1 \leq_\Phi \sigma_2 \Leftrightarrow (\exists \rho \in S)(\forall \tau \in T(\Phi, F, a))\rho(\sigma_1(\tau)) = \sigma_2(\tau).$$

The equivalence relation $\cong_\Phi$ in S over Φ is defined by

$$\sigma_1 \cong_\Phi \sigma_2 \Leftrightarrow \sigma_1 \leq_\Phi \sigma_2 \wedge \sigma_2 \leq_\Phi \sigma_1.$$

For any $\sigma \in S$, $[\sigma]_\Phi$ denotes the $\cong_\Phi$-equivalence class of σ in S.

If $\Phi = V$ then $\sigma_1 \leq_V \sigma_2 \Leftrightarrow (\exists \rho)\rho \circ \sigma_1 = \sigma_2$. $\leq_V$ and its associated equivalence relation $\cong_V$ are the standard structures on substitutions found in the literature. For this reason we drop the subscript V in the following and refer to them simply by $\leq$ and $\cong$ whenever there is no danger of confusing them with the identically named preorder and equivalence relation in T.

The definition of $\leq_\Phi$ expresses the fact that only the variables in Φ are relevant for comparing two substitutions. In fact it is obvious that $\sigma_1 \leq_\Phi \sigma_2$ if and only if $(\exists \rho \in S)(\forall \tau \in \Phi)\rho(\sigma_1(\tau)) = \sigma_2(\tau)$. This restriction to a certain subset Φ of V comes in handy in a later section when we need to generate "new" variables that play no role in a $\leq$-comparison. These new variables can be chosen from amongst the set $V - W$. W is always chosen such that $V - W$ is nonempty, in fact infinite.

For a given subset Φ of V we can ask ourselves whether it is possible to construct a sequence of ever more and more general subsitutions from a given starting substiution $\sigma \in S$. The answer to this question is no and is proved below.

Definition 6 *Let Φ be a subset of V; let σ be a substitution in S. Let the* length $l(\tau)$ *denote the number of occurrences of any $m \in F \cup V$ in τ for any $\tau \in T(V, F, a)$. We define the* degree $d(\sigma, \Phi)$ *of σ in Φ as follows.*

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

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

Theorem 2 *Let $\hat{S}_\Phi$ be the quotient set $T/_{\cong_\Phi} = \{[\sigma]_\Phi \mid \sigma \in S\}$; let $\leq_\Phi$ denote also the partial ordering in $\hat{S}_\Phi$ canonically induced by the preordering $\leq_\Phi$ in S. Then*

- *$(\hat{S}_\Phi, \leq_\Phi)$ is Noetherian for any $\Phi \subset V$.*

To prove this theorem we establish a lemma first, which is a simple generalization of a similar lemma in [Hue80].

Lemma 1 *Let Φ be a subset of V; let σ_1, σ_2 be substitutions in S. We will write $\sigma_1 <_\Phi \sigma_2$ if and only if $\sigma_1 \leq_\Phi \sigma_2$, but $\sigma_1 \not\cong_\Phi \sigma_2$. We have*

- $\sigma_1 \cong_\Phi \sigma_2 \Rightarrow d(\sigma_1, \Phi) = d(\sigma_2, \Phi)$
- $\sigma_1 <_\Phi \sigma_2 \Rightarrow d(\sigma_1, \Phi) < d(\sigma_2, \Phi)$

Proof *(Proof of lemma)*

Assume $\sigma_1 \cong_\Phi \sigma_2$. By definition we know that there exist substitutions α, β such that $\alpha(\sigma_1(x)) = \sigma_2(x)$ and $\beta(\sigma_2(x)) = \sigma_1(x)$ for all $x \in \Phi$. Consequently $\alpha(\beta(y)) = y$ for all $y \in V(\sigma_2(\Phi))$ and $\beta(\alpha(z)) = z$ for all $z \in V(\sigma_1(\Phi))$. From this we can conclude immediately that α is injective on $V(\sigma_1(\Phi))$ and β on $V(\sigma_2(\Phi))$. Now let F be any finite subset of Φ and let x be an arbitrary element of F. Since α is injective on $V(\sigma_1(\Phi))$ it is of course injective on $V(\sigma_1(x))$. Since α just replaces variables by variables, the length of an argument to α is invariant. Thus $l(\sigma_2(x)) = l(\alpha(\sigma_1(x))) = l(\sigma_1)$; furthermore, $|V(\sigma_2(F))| = |V(\alpha(\sigma_1(F)))| = |V(\sigma_1(F))|$. This establishes $(\sum_{x\in F} l(\sigma_1(x))) - |V(\sigma_1(F))| = (\sum_{x \in F} l(\sigma_2(x))) - |V(\sigma_2(F))|$ and consequently $d(\sigma_1, \Phi) = d(\sigma_2, \Phi)$.

Now let us assume $\sigma_1 <_\Phi \sigma_2$. By definition we have $(\forall x \in \Phi)\alpha(\sigma_1(x)) = \sigma_2(x)$ for some substitution α. α cannot be injective on $V(\sigma_1(\Phi))$ because otherwise we could construct a β as above, which would show $\sigma_1 \cong_\Phi \sigma_2$. If α is not injective on $V(\sigma_1(\Phi)$ then there are two cases. Either there is a $y_0 \in V(\sigma_1(\Phi)$ such that $\alpha(y_0) = f(\tau_1, \ldots, \tau_n)$ or $\alpha(y_0) = f$ for some functor $f \in F$ (first case), or $\alpha(\sigma(\Phi)) \subset V$ and there exist variables $y_1, y_2 \in V$ such that $\alpha(y_1) = \alpha(y_2) \wedge y_1 \neq y_2$ (second case). Considering the first case let x_0 be an element of Φ such that $y_0 \in V(\sigma_1(x_0))$. x_0 is guaranteed to exist since y_0 is in the range of σ under Φ. Let F be a finite subset of Φ containing x_0. Now imagine the $\sigma_1(x)$ where $x \in F$ written out as a string in which there is exactly one occurrence of every variable in it that is tagged with some mark. The number of untagged functor and variable occurrences is $d(\sigma_1, F)$. We can do the same for σ_2, of course, where the number of untagged functor and variable occurrences is $d(\sigma_2, F)$. Since we can derive this string representation for σ_2 by replacing all occurrences of a variable y in the string representation for σ_1 with $\alpha(y)$, if an occurrence y in σ_1 is untagged, then we can assume without loss of generality that all *the variable occurrences in $\alpha(y)$ in the corresponding position in sigma$_2$ are also untagged. If we consider the tagged occurrence of y_0 (in σ_1) in the first case then the corresponding string $\alpha(y_0)$ has at least one more untagged occurrence than y_0, namely the functor f. If we consider the untagged occurrences of y_0 in σ_1, then the corresponding string $\alpha(y_0)$ in σ_1 contains at least as many untagged element occurrences as the untagged y_0, namely one. Since by the way we did the tagging there cannot be any untagged variable occurrence in σ_1 whose corresponding substring in σ_2 has only tagged element occurrences, we can see that $d(\sigma_2, F) \geq d(\sigma_1, F) + 1$. Similarly, in the second case, consider the tagged occurrences of y_1 and y_2 in the string representation (with respect to F) of σ_1 and the corresponding occurrences of the variable $\alpha(y_1)(= \alpha(y_2))$ in the string representation (with respect to F) of σ_2. Since not both occurrences of $\alpha(y_1)$ can be tagged, this shows that there is at least one more untagged symbol in σ_2 than in σ_1. Consequently, we have again $d(\sigma_2, F) \geq d(\sigma_1, F) + 1$. This finishes the proof.*

The proof of the theorem is now straightforward.

Proof *(Proof of theorem)*

Assume there is a set of equivalence classes $\{E_i \mid i \in \mathcal{N}\}$ such that $E_i > E_{i+1}$ for all $\imath \in \mathcal{N}$. Let $\sigma_i \in E_i$ be arbitrary representatives of the E_i's for any $i \in \mathcal{N}$. By assumption we have $\sigma_i > \sigma_{i+1}$. We know that $d(\sigma_0, \Phi)$ is finite by definition of d. Lemma 1 asserts that for any i it must be that $d(\sigma_i, \Phi) > d(\sigma_{i+1}, \Phi)$. 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 can view this theorem as establishing an analogy between the structure of $\hat{T}$ and $\hat{S}_\Phi$ for any $\Phi \subset V$. A natural question to ask is whether or not $\hat{S}_\Phi$ forms a (complete) lattice under $\leq_\Phi$ just as $(\hat{T}, \leq)$ is a complete lattice. It is well-known, though, that $(\hat{S}_V, \leq_V)$ fails to be a lattice. Eder [Ede85] shows that the pair of substitutions $\{x \leftarrow f(x, f(y, z)), y \leftarrow f(x, f(y, z)), z \leftarrow f(x, f(y, z))\}$ and $\{x \leftarrow f(f(x, y), z), y \leftarrow f(f(x, y), z), z \leftarrow f(f(x, y), z)\}$ have an infinite set of upper bounds, but no least upper bound with respect to $\leq_V$. The reason for this "failure" is due to the fact that we cannot "hide" any variables from $\leq_V$. For subsets Φ of V that leave "enough" variables hidden in $V - \Phi$ the partial orders $(\hat{S}_\Phi, \leq_\Phi)$ have indeed a lattice structure. We make the previous precise in the following theorem.

Theorem 3 *Let Φ be a subset of V. Let $\hat{S}_\Phi$ be as before, but with an additional maximal element $\top$. The following statements are equivalent.*

- *$(\hat{S}_\Phi, \leq_\Phi)$ is a complete lattice.*
- *Φ is co-infinite; that is, $|V - \Phi| = \infty$.*

Furthermore, if Φ is co-finite, then $(\hat{S}_\Phi, \leq_\Phi)$ is not a lattice.

Lemma 2 *Let Φ be a co-infinite subset of V.*

- *Let σ_1, σ_2 be two substitutions in S. Then $\{\sigma_1, \sigma_2\}$ has a greatest lower bound with respect to $\leq_\Phi$; that is, there is a σ_0 such that $\sigma_0 \leq_\Phi \wedge \sigma_0 \leq_\Phi$ and for all $\sigma \in S$ for which $\sigma \leq_\Phi \sigma_0$ holds we have $\sigma \leq_\Phi \sigma_0$.*
- *Let R be a subset of S. Then R has a greatest lower bound with respect to $\leq_\Phi$.*

Of course, the second statement implies the first, but we can show that the first statement in connection with the Noetherian-ordering property of $\leq_\Phi$ (theorem 2) implies the second statement.

Proof *(Proof of lemma)*

Consider any variable $x \in D(\sigma_1) \cup D(\sigma_2)$. Clearly for any lower bound σ of $sigma_1$ and $sigma_2$ it must hold that

Proof

First we prove that for any co-finite $\Phi \subset V$ there is a pair of substitutions with no least upper bound. A simple generalization of Eder's pair will do the trick. Let Φ be a co-finite set. Without loss of generalization we can assume that $V - \Phi = \{w_1, \ldots, w_n\}$ 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 Φ. Now with $\rho_i = \{x_i \leftarrow f(x_i, f(y_i, z_i)), y_i \leftarrow f(x_i, f(y_i, z_i)), z_i \leftarrow f(x_i, f(y_i, z_i))\}$ and $\sigma_i = \{x_i \leftarrow f(f(x_i, y_i), z_i), y_i \leftarrow f(f(x_i, y_i), z_i), z_i \leftarrow f(f(x_i, y_i), z_i)\}$ consider the substitutions $\rho = \cup_{i \in \{1,\ldots,n+1\}} \rho_i$ and $\sigma = \cup_{i \in \{1,\ldots,n+1\}} \sigma_i$.[7] The minimal upper bounds of ρ and σ are the the substitutions

$$\cup_{i \in \{1,\ldots,n+1\}} \{x_i \leftarrow f(f(s_i, t_i), f(u_i, v_i)), y_i \leftarrow f(f(s_i, t_i), f(u_i, v_i)), z_i \leftarrow f(f(s_i, t_i), f(u_i, v_i))\}$$

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 $w \notin \{w_1, \ldots, w_n, x_1, \ldots, x_{n+1}, y_1, \ldots, y_{n+1}, z_1, \ldots, z_{n+1}\}$. Thus w is in Φ. 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 $\sigma_1 \leq_\Phi \sigma_2$ because $w \notin V(\sigma_2)$.

[7] 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.

6 Equations and Inequalities

6.1 Equations and Sets of Equations

Definition 7 - *A* (term) equation *is a pair of terms written as*

$$\tau = \upsilon$$

where $\tau, \upsilon \in T$. A set of (term) equations *is written as*

$$\{\tau_1 = \upsilon_1, \ldots, \tau_n = \upsilon_n\}.$$

- *Any substitution $\sigma \in S$ for which*

$$\sigma(\tau) \equiv \sigma(\upsilon)$$

is called a unifier *of the equation $\tau = \upsilon$. $\sigma \in S$ is a unifier of a set of equations $\{\tau_1 = \upsilon_1, \ldots, \tau_n = \upsilon_n\}$ if it is a unifier of every equation in it; that is,*

$$\sigma(\tau_1) \equiv \sigma(\upsilon_1), \ldots, \sigma(\tau_n) \equiv \sigma(\upsilon_n)$$

.

- *For any equation G, $U(G)$ is its set of unifiers; similarly, for any set of equations G, $U(\mathcal{G})$ denotes G's unifiers.*

Let us consider the sets of sets of unifiers $\{U(G) \mid \text{G is a term equation}\}$ and $\{U(\mathcal{G}) \mid \text{G is a set of equations}\}$. It is conceivable — and possible — that those two sets are not the same. However, under quite weak conditions they are always identical.

Definition 8 *We call (F, a) — the functors with their arity function —* nonlinear *if there is an $f \in F$ such that $a(f) \geq 2$.*

Proposition 2 *Let, as usual, $T = T(V, F, a)$. The following two statements are equivalent.*

- *$\{U(G) \mid G \text{ is a term equation over } T\}$ and $\{U(\mathcal{G}) \mid G \text{ is a set of equations over } T\}$ are identical.*

- *(F, a) is nonlinear.*

Proof

Let the two sets $\{U(G) \mid G \text{ is a term equation over } T\}$ and $\{U(\mathcal{G}) \mid G \text{ is a set of equations over } T\}$ be identical. Consider the set of equations $\mathcal{G}_{,} = \{x_1 = y_1, x_2 = y_2\}$. Clearly $\sigma_0 = \{x_1 \leftarrow y_1, x_2 \leftarrow y_2\}$ is a unifier of $\mathcal{G}_{,}$, but no proper subset of σ_0 is. Let G_0 be a term equation $\tau = \upsilon$ such that $U(G_0) = U(\mathcal{G}_{,})$, which is guaranteed to exist by assumption. Let us further assume that there is no functor with arity greater than 1. In this case all terms, in particular τ and υ have at most one variable occurrence. Since σ_0 is a unifier of G_0, τ and υ are either identical or contain either x_1 and x_2 or x_3 and x_4 in corresponding positions in τ and υ. But in any of those cases there would be a proper subset of σ_1 that is also a unifier of G_0. Consequently our assumption that there is no functor with arity greater than 1 is false, and thus (F, a) is nonlinear.

To prove the converse, assume that functor $f_0 \in F$ has arity $a(f) = n_0 \geq 2$. Note that for any equation $\tau = \upsilon$ the set of equations $\{\tau = \upsilon\}$ has the same unifiers. To show that for any set of equations there is a simple term equation with the same unifiers we can construct a sequence of terms $\psi_1, \psi_2, \ldots$ by $\psi_1 = f_0(x_1, \ldots, x_{n_0}), \psi_2 = f_0(f_0(x_1, \ldots, x_{n_0}), f_0(x_{n_0+1}, \ldots, x_{2n_0}), \ldots, f_0(x_{(n_0-1)n_0+1}, \ldots, x_{n_0^2}))$

and so on. For all $i \in \mathcal{N}$ the term ψ_i has in_0 variables. Let $\mathcal{G} = \{\tau_1 = \upsilon_1, \ldots, \tau_n = \upsilon_n\}$ be any set of equations over $T(V, F, a)$. Let i_0 be such that $i_0 n_0 \geq n$. We define $\sigma_1 = \{x_1 \leftarrow \tau_1, \ldots, x_n \leftarrow \tau_n\}$ and $\sigma_2 = \{x_1 \leftarrow \upsilon_1, \ldots, x_n \leftarrow \upsilon_n\}$. Now it is easy to see that any unifier of G is a unifier of $\sigma_1(\psi_{i_0}) = \sigma_2(\psi_{i_0})$ and the other way around.

In the rest of the paper we will always assume that F is nonlinear. Consequently we could work only with term equations instead of sets of such equations. However, sets of equations come in handy when describing algorithms for computing unifiers and will follow us around for a while.

6.2 Inequalities and Sets of Inequalities

Definition 9

- *A* (term) inequality *is a pair of terms written as*

$$\tau \preceq \upsilon$$

where $\tau, \upsilon \in T$. A set of (term) inequalities *is written as*

$$\{\tau_1 \preceq \upsilon_1, \ldots, \tau_n \preceq \upsilon_n\}.$$

- *A substitution $\sigma \in S$ for which*

$$\sigma(\tau) \leq \sigma(\upsilon)$$

holds, i. e. $(\exists\rho \in S)\rho(\sigma(\tau)) \equiv \sigma(\upsilon)$, is called a semi-unifier *of $\tau \preceq \upsilon$. $\sigma \in S$ is a* nonuniform *semi-unifier of $\{\tau_1 \preceq \upsilon_1, \ldots, \tau_n \preceq \upsilon_n\}$ if $\sigma(\tau_1) \leq \sigma(\upsilon_1), \ldots, \sigma(\tau_n) \leq \sigma(\upsilon_n)$ or, equivalently, $(\exists\rho_1, \ldots, \rho_n)\rho_1(\sigma(\tau_1)) \equiv \sigma(\upsilon_1), \ldots, \rho_n(\sigma(\tau_n)) \equiv \sigma(\upsilon_n)$. σ is a* uniform *semi-unifier of $\{\tau_1 \preceq \upsilon_1, \ldots, \tau_n \preceq \upsilon_n\}$ if $(\exists\rho)\rho(\sigma(\tau_1)) \equiv \sigma(\upsilon_1), \ldots, \rho(\sigma(\tau_n)) \equiv \sigma(\upsilon_n)$.*

- *For any inequality G, $V(G)$ is its set of semi-unifiers; similarly, for any set of inequalities G, $V(\mathcal{G})$ denotes G's uniform semi-unifiers, and $\vec{V}(\mathcal{G})$ stands for G's nonuniform semi-unifiers.*

In complete analogy to equations semi-unifiers of inequalities and *uniform* semi-unifiers of sets of inequalities have the same structure.

Proposition 3 ??

Let, as usual, $T = T(V, F, a)$. The following two statements are equivalent.

- *$\{V(G) \mid G$ is a term inequality over $T\}$ and $\{V(\mathcal{G}) \mid G$ is a set of inequalities over $T\}$ are identical.*
- *(F, a) is nonlinear.*

Proof

Let the two sets $\{V(G) \mid G$ is a term inequality over $T\}$ and $\{V(\mathcal{G}) \mid G$ is a set of inequalities over $T\}$ be identical. Consider the set of inequalities $\mathcal{G}_\infty = \{y_0 \preceq x_1, y_0 \preceq x_2\}$. Clearly $\sigma_1 = \{x_1 \leftarrow x_2\}$ is a semi-unifier of $\mathcal{G}_\infty$, but $\{\}$ is not. If we assume that no functor in F has arity greater than 1, we already know that all terms in $T(V, F, a)$ have at most one variable occurrence. Thus if an inequality $\tau \preceq \upsilon$ has a solution at all then there must be subterms τ' and υ' of τ and υ, respectively, such that $\tau' \preceq \upsilon'$ has the same set of unifiers as $\tau \preceq \upsilon$ and either τ' is a variable or υ' is a variable. If τ' is a variable then the identity substitution $\{\}$ is a semi-unifier, and if it is not, then σ_1 is not a semi-unifier of $\tau' \preceq \upsilon'$. Consequently there is no term inequality with the same set of semi-unifiers as $\mathcal{G}_\infty$ under the assumption that F has no functor with arity greater than 1, and we can conclude that (F, a) must be nonlinear.

To prove the converse we can make use of the same construction as in the corresponding part of the proof of proposition 2.

While this proposition gives a straightforward connection between semi-unifiers of inequalities and uniform semi-unifers of sets of inequalities, it is not clear off-hand how nonuniform semi-unifiers are related. We will come back to this question later.

6.3 Simple Connections

We may ask ourselves if and how the unifiers of an equation $\tau = \upsilon$ are related to the semi-unifiers of the corresponding inequality $\tau \preceq \upsilon$. In this subsection we give a couple of straightforward answers to some simple questions of this nature.

Proposition 4 *Let $\tau, \upsilon \in T$ be terms. Let $G_=$ be the equation $\tau = \upsilon$, and let $G_\preceq$ be the inequality $\tau \preceq \upsilon$.*

- *If there is a unifier for $G_=$ then there is a semi-unifier for $G_\preceq$; more specifically, every unifier of $G_=$ is a semi-unifier of $G_\preceq$.*
- *If τ is a variable, i. e. $\tau = x \in V$, then $U(G_=) = \{\sigma \circ \{x \leftarrow \upsilon\} \mid \alpha \in S\}$ whenever υ does not contain x; otherwise $U(G_=)$ is empty. For $\tau = x \in V$, $V(G_\preceq) = S$ no matter whether υ contains x or not.*

  If, on the other hand, υ is a variable, i. e. $\upsilon = y \in V$, and τ is a term of the form $f(\tau_1, \ldots, \tau_n)$ for some functor $f \in F$ and terms $\tau_1, \ldots, \tau_n \in T$, then $U(G_=) = \{\sigma \circ \{y \leftarrow \tau\} \mid \alpha \in S\}$ whenever τ does not contain y; otherwise $U(G_=)$ is empty, which is completely symmetric to the previous case. However, for $\upsilon = y \in V$ and $\tau = f(\tau_1, \ldots, \tau_n)$, $V(G_\preceq)$ is properly contained in S, but contains $U(G_=)$. Unless τ contains y, in which case $V(G_\preceq)$ is empty, or τ contains no variables at all, the containment $V(G_\preceq) \supset U(G_=)$ is proper.

The first part of this proposition follows immediately from the definition of semi-unifier. The second part is also very easy to prove.

7 Most General Unifiers and Most General Semi-Unifiers

7.1 Most General Unifiers

Definition 10 *Let G be a term equation (or set of term equations). Let Φ be a subset of V. A substitution $\sigma \in S$ is called a* most general unifier (mgu) *of G over Φ if it is a unifier of G and there is no other unifier ρ of G such that $\rho <_{\leq} \sigma$.*

Note that every equation that has a unifier has a most general unifier. This follows immediately from theorem 2. Furthermore, we have the following theorems, which are well-known (c. f. [Ede85], [LMM86]).

Theorem 4 *Let σ_1 and σ_2 be most general unifiers of term equation G. Then $\sigma_1 \cong_V \sigma_2$.*

Because $\sigma_1 \cong_V \sigma_2$ implies $\sigma_1 \cong_\Phi \sigma_2$ for any subset $\Phi \subset V$ we see immediately that the above theorem holds for any such Φ.

Theorem 5 *Let $\rho \in S$ be a most general unifier of term equation G. Then there is an idempotent most general unifier σ of G such that $\rho \cong_V \sigma$.*

In view of theorem 4 we could have formulated this theorem somewhat stronger by asserting that for every unifiable equation G there is an idempotent most general unifier. However, it is conceivable that for a different set of substitutions (instead of all most general unifiers of an equation) theorem 4 fails, but theorem 5 still holds. The particular formulation of these theorems reflects this separation of concerns.

Since there are substitutions that are not $\cong_V$-equivalent to any idempotent substitution, as a consequence of the last theorem not every substitution in S is a most general unifier (of some equation). For example, $\{z_i \leftarrow f(z_i), \ldots, z_i \leftarrow f(z_i)\}$ has no $\cong_V$-equivalent substitution [Ede85].

7.2 Most General Semi-Unifiers

Definition 11 *Let Φ be a subset of V.*

- *Let G be a term inequality. A substitution $\sigma \in S$ is called a* most general semi-unifier (mgsu) *of G over Φ if it is a semi-unifier of G and there is no other semi-unifier ρ of G such that $\rho <_\Phi \sigma$.*
- *Let $\mathcal{G}$ be a set of term inequalities. Let σ be a substitution in S. σ is called a* most general uniform semi-unifier (mgusu) *of $\mathcal{G}$ if it is a uniform semi-unifier of $\mathcal{G}$ and there is no other uniform semi-unifier ρ of $\mathcal{G}$ such that $\rho <_\Phi \sigma$. σ is called a* most general nonuniform semi-unifier (mgnsu) *of $\mathcal{G}$ if it is a nonuniform semi-unifier of $\mathcal{G}$ and there is no other nonuniform semi-unifier ρ of $\mathcal{G}$ such that $\rho <_\Phi \sigma$.*

Again, we are assured of the existence of most general (uniform/nonuniform) semi-unifiers for inequality (set of inequalities) due to theorem 2. Unfortunately, the analog of theorem 4 does not hold. In fact we have the following proposition:

Proposition 5 *There is a famlily $\mathcal{F} = \{G_i\}$ of term inequalities such that for every $i \in \mathcal{N}$ the inequality G_i has most general semi-unifiers σ_{i1} and σ_{i2} such that $\sigma_{i1} \not\cong_V \sigma_{i2}$.*

Proof

Consider $G_i = f(x_1, \ldots, x_i) \preceq y$.[8] *The substitutions* $\sigma_{i1} = \{y \leftarrow f(u_1, \ldots, u_i)\}$ *and* $\sigma_{i2} = \{y \leftarrow f(v_1, \ldots, v_i)\}$ *are most general semi-unifiers of* G_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 substitutions* $\alpha \in S$ *such that* $\alpha \circ \sigma_{i1} = \sigma_{i2}$ *or* $\alpha \circ \sigma_{i2} = \sigma_{i1}$.

Unfortunately, the analog of theorem 5 is not true either.

Proposition 6 *There is a family* $\mathcal{F} = \{\mathcal{G}_\rangle\}$ *of sets of term inequalities such that for every* $i \in \mathcal{N}$ *the set of inequalities* $\mathcal{G}_\rangle$ *has no idempotent most general uniform semi-unifier.*

Proof

Consider $\mathcal{G}_\rangle = \{f(y_1) \preceq z_1, \ldots, f(y_i) \preceq z_i\}$. *The substitution* $\sigma = \{z_1 \leftarrow f(z_1), \ldots, z_i \leftarrow f(z_i)\}$ *and its* $cong_V$*-equivalent substitutions are the only most general unifiers of* $\mathcal{G}_\rangle$. *As we remarked earlier there is no idempotent substitution amongst them.*

Some of the questions that arise from this "failure" of most general semi-unifiers to exhibit the same structure as most general unifiers are:

- Is there any other notion that will let us say that most general semi-unifiers are unique in some sense?
- Are all substiutions most general semi-unifiers?
- Is there any other ordering but $\leq_V$ such that the substitutions form a (complete) lattice?

We will address these questions in the following sections. In particular, we will discover a lattice structure on substitutions with respect to $\leq_\Phi$ for all co-infinite subsets Φ of V.

7.3 The Structure of Unifiers

A term equation $\tau_1 = \tau_2$ can have zero, one, or many solutions. But the set of all solutions is well-structured. In particular there always exists a least solution called a *most general unifier* (of τ_1 and τ_2). This, and the importance of idempotent substitutions, is expressed in the following theorem.

Definition 12 *The set of unifiers* $\mathcal{U}(\tau_1, \tau_2) is defined by \{\sigma \mid \sigma(\tau_1) \equiv \sigma(\tau_2)\}$

Theorem 6 *Let* $\tau_1, \tau_2 \in T$. *Then* $(\mathcal{U}(\tau_1, \tau_2), \preceq_\Phi)$ *is a complete lattice for any* $\Phi \subset V$. *Furthermore, if* $\mathcal{U}(\tau_1, \tau_2)$ *is nonempty, then its greatest element is* $\top$ *and its least element an equivalence class of substitutions containing an idempotent substitution.*

[8] We can always assume that any functor can have a variable number of arguments as long as the underlying (F, a) is nonlinear. See propositions 2 and **??** and their proofs.

8 Most General Semi-Unifiers are Unique

In the previous sections we saw that most general semi-unifiers (of any fixed inequality G) are not unique modulo $\cong_V$-congruence whereas most general unifiers (of any fixed equation G') are. In this section we will show that most general (uniform/nonuniform) semi-unifiers are unique modulo $\cong_\Phi$ *for any co-infinite subset* Φ *of* V.

8.1 Most General Uniform Semi-Unifiers

Recall that simple term inequalities give rise to the same (most general) semi-unifiers as the *uniform* (most general) semi-unifiers of sets of equations under our proviso of a nonlinear functor system (F, a). In this section we will show that simple all most general semi-unifier (with respect to $\leq$) of term inequalities are $\cong_\Phi$-congruent for any $\Phi \subset V$. This implies that sets of inequalities also have most general uniform semi-unifiers modulo $\cong_\Phi$.

Proposition 7 *Let $\tau_1, \tau_2 \in T$ be arbitrary terms; let Φ be a subset of V. Then $\{\sigma \in S \mid \tau_1 \leq \sigma(\tau_2)\}$ has a greatest lower bound with respect to Φ; i. e., there exists a $\sigma_0 \in S$ such that $\tau_1 \leq \sigma_0(\tau_2)$ and $(\forall \sigma \in S)\tau_1 \leq \sigma(\tau_2) \Rightarrow \sigma_0 \leq_\Phi \sigma$.*

Proof
By theorem 1 we know that the lowest upper bound $[\tau] = [\tau_1] \vee [\tau_2]$ exists. Without loss of generality we can assume that $V(\tau) \cap \Phi = \emptyset$. Consequently there is a substitution σ_0' such that $\sigma_0'(\tau_2) = \tau$. Now define σ_0 as follows.

$$\sigma_0(x) = \begin{cases} \sigma_0'(x), & \textit{if } x \textit{ occurs in } \tau_2 \\ x, & \textit{otherwise} \end{cases}$$

It is easy to see that $\sigma_0(\tau_2) = \tau$ and, consequently, $\tau_1 \leq_\Phi \sigma_0(\tau_2)$ by construction of σ_0. Now let σ be any other substitution such that $\tau_1 \leq \sigma(\tau_2)$. $\sigma(\tau_2)$ must be an upper bound of τ_1 and τ_2, and thus $\tau \leq \sigma(\tau_2)$ must hold. By definition this means that there is a substitution ρ' such that $\rho'(\tau) = \sigma(\tau_2)$. If we define ρ by

$$\rho(x) = \begin{cases} \rho'(x), & \textit{if } x \textit{ occurs in } \tau \\ \sigma(x), & \textit{otherwise} \end{cases}$$

then it is straightforward to check that $(\forall x \in \Phi)\rho(\sigma_0(x)) = \sigma(x)$, which means $\sigma_0 \leq_\Phi \sigma$.

This proposition tells us that the set $\{\sigma \in S \mid \tau_1 \leq \sigma(\tau_2)\}$ has a greatest lower bound with respect to any co-infinite $\Phi \subset V$ (and which is unique modulo $\cong_\Phi$) for any choice of $\tau_1, \tau_2 \in T$. We will denote this greatest lower bound by $\bigwedge\{\sigma \in S \mid \tau_1 \leq \sigma(\tau_2)\}$. We assume that we have a "top"-element $\top$ in S that is greater than any other substitution in S.

From now on let us fix a co-infinite subset $\Phi \subset V$.

Lemma 3 *Let $M_1, M_2 \in T$ be arbitrary terms; let Φ be a subset of V. Then $\{\sigma \in S \mid M_1 \leq \sigma(M_2)\}$ has a greatest lower bound with respect to Φ; i. e., there exists a $\sigma_0 \in S$ such that $M_1 \leq \sigma_0(M_2)$ and $(\forall \sigma \in S)M_1 \leq \sigma(M_2) \Rightarrow \sigma_0 \leq_\Phi \sigma$.*

Proof *By theorem 1 we know that the lowest upper bound $[M] = [M_1] \vee [M_2]$ exists. Without loss of generality we can assume that $V(M) \cap \Phi = \emptyset$. Consequently there is a substitution σ_0' such that $\sigma_0'(M_2) = M$. Now define σ_0 as follows.*

$$\sigma_0(x) = \begin{cases} \sigma_0'(x), & \textit{if } x \textit{ occurs in } M_2 \\ x, & \textit{otherwise} \end{cases}$$

It is easy to see that $\sigma_0(M_2) = M$ and, consequently, $M_1 \leq_\Phi \sigma_0(M_2)$ by construction of σ_0. Now let σ be any other substitution such that $M_1 \leq \sigma(M_2)$. $\sigma(M_2)$ must be an upper bound of M_1 and M_2, and thus $M \leq \sigma(M_2)$ must hold. By definition this means that there is a substitution ρ' such that $\rho'(M) = \sigma(M_2)$. If we define ρ by

$$\rho(x) = \begin{cases} \rho'(x), & \textit{if } x \textit{ occurs in } M \\ \sigma(x), & \textit{otherwise} \end{cases}$$

then it is straightforward to check that $(\forall x \in \Phi)\rho(\sigma_0(x)) = \sigma(x)$, which means $\sigma_0 \leq_\Phi \sigma$.

This proposition tells us that the set $\{\sigma \in S \mid M_1 \leq \sigma(M_2)\}$ has a greatest lower bound with respect to any co-infinite $\Phi \subset V$ (and which is unique modulo $\cong_\Phi$) for any choice of $M_1, M_2 \in T$. We will denote this greatest lower bound by $\bigwedge\{\sigma \in S \mid M_1 \leq \sigma(M_2)\}$. We define $\hat{\mathcal{S}}$ to be $\mathcal{S}$ with an additional maximum element $\top$.

From now on let us fix a co-infinite subset $\Phi \subset V$.

Definition 13 *Define $F_{M_1,M_2}(\sigma) = \bigwedge\{\sigma' \in S \mid \sigma(M_1) \leq \sigma'(M_2)\}$.*

In view of the previous proposition we are assured that F is well-defined. The following lemma will tell us that F is monotonic.

Lemma 4 *F is monotonic with respect to $\leq_\Phi$; more precisely, $(\forall \sigma \in S)\sigma_1 \leq_\Phi \sigma_2 \wedge D(\sigma_1) \cup D(\sigma_2) \subset \Phi \Rightarrow F(\sigma_1) \leq_\Phi F(\sigma_2)$.*

Proof *It is sufficient to show $\{\sigma' \in S \mid \sigma_2(M_1) \leq \sigma'(M_2)\} \subset \{\sigma' \in S \mid \sigma_1(M_1) \leq \sigma'(M_2)\}$ or, even simpler, $\sigma_2(M_1) \leq \sigma'(M_2) \Rightarrow \sigma_1(M_1) \leq \sigma'(M_2)$ for any $\sigma' \in S$. But this is trivial because, by assumption (the domains of σ_1 and σ_2 are contained in Φ), for some ρ we have $\sigma_1(M_1) \leq \rho(\sigma_1(M_1) = \sigma_2(M_2))$.*

We can define a sequence of substitution operators

$$F^k_{M_1,M_2}(\sigma) = \begin{cases} F_{M_1,M_2}(F^{(k-1)}_{M_1,M_2}(\sigma), & k > 0 \\ \sigma, & k = 0 \end{cases}$$

and their limit

$$F^\infty_{M_1,M_2}(\sigma) = \begin{cases} F^k_{M_1,M_2}(\sigma), & F^k_{M_1,M_2}(\sigma) = F^{(k+1)}_{M_1,M_2}(\sigma) \\ \top, & \text{otherwise} \end{cases}$$

Theorem 7 *Every SEI S has a most general nonuniform, respectively uniform, semi-unifier.*[9]

Proof *We can simply apply the above lemmas.*

The results above can be considerably strenghtened, yielding an analog of the main structure theorem of unification. Let us say that two substitutions σ_1 and σ_2 are *weakly equivalent* with respect to an SEI S if and only if $\sigma_1 \cong_{V(S)} \sigma_2$ where $V(S)$ denotes the set of variables in S.

For any SEI S let $\hat{(}SU(S)_{V(S)})$ and $\hat{(}USU(S)_{V(S)})$ denote the sets of equivalence classes $\{[\sigma]_{V(S)} \mid \sigma \in SU(S)\}$ and $\{[\sigma]_{V(S)} \mid \sigma \in USU(S)\}$, respectively, with an added maximum element $\top$.

Theorem 8 *Let S be any SEI. Then*

1. *$(\hat{(}SU(S)_{V(S)}), \leq_{V(S)})$ is Noetherian.*

[9] Recall that Φ is an arbitrary, but fixed co-infinite subset of V.

2. $(\hat{(}SU(S)_{V(S)}), \leq_{V(S)})$ *is a complete lattice.*

Similarly,

1. $(\hat{(}USU(S)_{V(S)}), \leq_{V(S)})$ *is Noetherian.*

2. $(\hat{(}USU(S)_{V(S)}), \leq_{V(S)})$ *is a complete lattice.*

9 Computing Most General Unifiers and Most General Semi-Unifiers

9.1 An Algorithm for Computing Most General Unifiers

Although it is not more or less difficult to solve a set of simultaneous equations instead of just a single equation, we describe, for convenience, an algorithm originally used by Herbrand and the basis for Martelli and Montanari's algorithms that solves sets of equations.

Definition 14 *Let $E = \{\tau_{11} = \tau_{12}, \ldots, \tau_{n1} = \tau_{n2}\}$ be a set of term equations. The substitution $\sigma \in S$ is a* unifier *of E if $\tau_{i1} \equiv \tau_{i2}, 1 \leq i \leq n$.*

Given a set of equations the algorithm chooses randomly, but fairly, an equation $e \in E$, for which there is a rule in the following rule system H, and takes an action depending on the form of e until no rule is applicable any more.

$f(\tau_1, \ldots, \tau_m) = f(\upsilon_1, \ldots, \upsilon_m)$**:** Replace e by the equations $\tau_1 = \upsilon_1, \ldots, \tau_m = \tau_m$.

$f(\tau_1, \ldots, \tau_k) = g(\upsilon_1, \ldots, \upsilon_l$**:** Halt with failure (functor clash).

$f(\tau_1, \ldots, \tau_m) = x$**:** Replace by $x = f(\tau_1, \ldots, \tau_m)$.

$x = f(\tau_1, \ldots, \tau_m)$ **where x occurs in at least one of** $tau_1, \ldots, \tau_m$**:** Halt with failure (occurs check).

$x = f(\tau_1, \ldots, \tau_m)$ **where x does not occur in** $tau_1, \ldots, \tau_m$**, but occurs in another equation** $e' \in E$**:** Replace x by $f(\tau_1, \ldots, \tau_m)$ in e'.

$x = x$**:** Delete e.

Theorem 9 *Given a set of equations E, the algorithm H either terminates with failure or returns a set of equations $E' = \{x_1 = \rho_1, \ldots, x_p = \rho_p\}$ such that $x_1, \ldots, x_p \in V$ and x_i does not occur in ρ_j for $1 \leq i, j \leq p$. If H terminates with failure the set E has no unifier. If H succeeds, then the substitution $\sigma = \{x_1 \leftarrow \rho_1, \ldots, x_p \leftarrow \rho_p\}$ is an idempotent most general unifier of E.*

The algorithm H is conceptually clear, but very inefficient on the standard representation of terms as strings. This is due to the space explosion that can arise in the step that replaces all occurrences of a variable x in E by the right-hand side of an equation $x = \tau$.

Definition 15 *The* size *of a term equation set E is the number of nonblank character used to write it down. The size of E is denoted by $|E|$.*

Proposition 8 *There is a class of equation sets $\mathcal{E} = \{E_1, \ldots, E_m$ such that for any $E \in \mathcal{E}$ of size n the equation set E' returned by H has size $\Omega(2^n)$.*

The copying cost and space blow-up can be avoided in a graph-theoretic representation of terms in which occurrences of a variable are designated by pointers to a single variable. Amazingly, in such a representation the unification problem becomes an instance of the equivalence problem for finite automata. This problem has an elegant $O(nG(n))$ time algorithm [AHU74, pp. 143-145]. By exploiting the specific structure of unification problems linear time algorithms are also possible [PW78, MM82].

9.2 Basic Properties of Semi-Unification

References

[AHU74] A. Aho, J. Hopcroft, and J. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.

[Ble77] W. Bledsoe. Non-resolution theorem proving. *Artificial Intelligence*, 9(1):1–35, 1977.

[Bue86] W. Buettner. Unification in the data structure sets. In *Proc. 8th Int'l Conf. on Automated Deduction*, pages 470–488. Springer-Verlag, 1986. Lecture Notes in Computer Science, Vol. 230.

[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.

[Gra77] G. Graetzer. *Universal Algebra.* Addison-Wesley, 1977.

[Hen88] Fritz Henglein. Semi-unification. Technical Report (SETL Newsletter) 222, New York University, April 1988.

[Her68] J. Herbrand. Recherches sur la theorie de la demonstration. In *Ecrits logiques de Jacques Herbrand.* PUF, Paris, 1968. thèse de Doctorat d'Etat, Université de Paris (1930).

[Hew71] C. Hewitt. *Description and Theoretical Analysis (Using Schemata) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot.* PhD thesis, MIT, 1971.

[Hue75] G. Huet. A unification algorithm for typed *lambda*-calculus. *Theoretical Computer Science*, 1(1):27–57, 1975.

[Hue76] G. Huet. *Résolution d'equations dans des langages d'ordre 1, 2, . . . , omega (thèse de Doctorat d'Etat).* PhD thesis, Univ. Paris VII, September 1976.

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

[Hus85] H. Hussmann. Unification in conditional equational theories. In *European Conf. on Computer Algebra (EUROCAL)*, pages 543–553. Springer-Verlag, April 1985. Lecture Notes in Computer Science, Vol. 204; also Universitaet Passau technical report MIP-8502, January 1985.

[KN86] D. Kapur and P. Narendran. NP-completeness of the set unification and matching problems. In *Proc. 8th Int'l Conf. on Automated Deduction*, pages 489–495. Springer-Verlag, 1986. Lecture Notes in Computer Science, Vol. 230.

[Kow79] R. Kowalski. *Logic for Problem Solving.* Artificial Intelligence Series. North-Holland, 1979.

[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, January 1988.

[LMM86] J. Lassez, M. Maher, and K. Marriott. Unification revisited. Technical report, IBM Yorktown Heights, 1986.

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

[Mee83] L. Meertens. Incremental polymorphic type checking in B. In *Proc. 10th ACM Symp. on Principles of Programming Languages (POPL)*, pages 265–275, 1983.

[MM82] A. Martelli and U. Montanari. An efficient unification algorithm. *ACM Transactions on Programming Languages and Systems*, 4(2):258–282, April 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, December 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.

[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.

[Pra60] D. Prawitz. An improved proof procedure. *Theoria*, 26:102–139, 1960.

[PW78] M. Paterson and M. Wegman. Linear unification. *J. Computer and System Sciences*, 16:158–167, 1978.

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

[Rob65] J. Robinson. A machine-oriented logic based on the resolution principle. *J. Assoc. Comput. Mach.*, 12(1):23–41, 1965.

[Sie84] J. Siekmann. Universal unification. In *Proc. 7th Int'l Conf. on Automated Deduction*, pages 1–42. Springer-Verlag, 1984. Lecture Notes in Computer Science, Vol. 170, Springer-Verlag.

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

[Sta88] R. Statman. Personal communication, May 1988.

[Sti81] M. Stickel. A unification algorithm for associative-commutative functions. *J. Assoc. Comput. Mach.*, 28(3):423–434, July 1981.

[WPP77] D. Warren, L. Pereira, and F. Pereira. Prolog — the language and its implementation compared with LISP. *SIGPLAN Notices*, 12(8):109–115, 1977.