Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern 1, W. Germany

SEKI-PROJEKT

SEKI MEMO

COMBINATION OF UNIFICATION ALGORITHM

Alexander Herold

MEMO SEKI-85-VIII-KL

COMBINATION OF UNIFICATION ALGORITHM

Alexander Herold

MEMO SEKI-85-VIII-KL

COMBINATION OF UNIFICATION ALGORITHMS

Alexander Herold

Universität Kaiserslautern
Fachbereich Informatik
Postfach 3049
6750 Kaiserslautern
F.R. Germany

MEMO SEKI-85-VIII-KL

ABSTRACT:

Unification in equational theories, i.e. solving equations in varieties, is a basic operation in many applications of computer science, particularly in automated deduction [Si 84]. A combination of unification algorithms for regular finitary collapse free equational theories with disjoint function symbols is presented. The idea is first to replace certain subterms by constants and to unify this constant abstraction and then in a recursive step to handle the replaced subterms. Total correctness is shown, i.e. the algorithm terminates and yields a correct and complete set of unifiers provided the special algorithms do so.

CONTENTS

1. Introduction
2. Definitions and Notations
3. The Algorithm
4. The Merge of Substitutions
5. An Example
6. Termination
7. Correctness and Completeness
8. Conclusion

1. INTRODUCTION

Unifcation theory is concerned with problems of the following kind: given two terms built from function symbols, constants and variables, do there exist terms that can be substituted for the variables such that the two terms thus obtained become equal? Robinson [Ro 65] was the first to give an algorithm to find such a substitution with the additional property that the returned 'unifier' is most general (or is an mgu for short), i.e. all other substitutions 'unifying' the two terms can be computed from that substitution. From an algebraic point of view unification is solving equations and an mgu is a 'basis' of the whole set of unifiers.

Equational unification extends the classical unification problem to solving equations in equationally defined theories. But then there may not exist one single mgu. Depending on the equational theory there are finite or infinite sets of mgu's and in some cases the set of mgu's does not even exist. The equational theories can therefore be classified into unitary, finitary and infinitary theories and the class of nullary theories. In the literature there are many unification algorithms for special equational theories,but they only solve problems with input terms built from the function symbols defining the equational theories, arbitrary constants and variables. For a detailed bibliography we refer to the-state-of-the-art survey of J. Siekmann [Si 84]. In that article the problem of extending these algorithms to handle terms with additional 'free' function symbols (i.e. there is no equational theory defined for that function symbol) is mentioned. F. Fages was the first who solved that problem for the equational theory defined by an associative and commutative (AC) function symbol [Fa 84]. Building upon this work K. Yelick [Ye 85] and E. Tidén [Ti 85] independently gave algorithms for combining finitary theories by abstracting those subterms to variables that do not belong to the theory of the top function symbol. Yelick restricts the problem to regular finitary collapse free theories (she calls them confined) whereas Tidén gives a proof of the completeness of his algorithm for the whole class of finitary theories. But in that general

framework termination fails. Another approach is given by C. Kirchner [Ki 85] who tackles the problem by a decomposition of the terms to be unified. This algorithm only admits a more restrictive class of equational theories than the regular finitary collapse free theories.
Working on an extension of the AC-unification algorithm of Livesey and Siekmann [LS 76][HS 85] we independently found a unification algorithm for regular finitary collapse free equational theories. The total correctness of the algorithm is shown, i.e. the algorithm terminates with a complete and correct set of unifiers. The essential idea of the algorithm is as follows: for the given terms the subterms not starting with a function symbol from the same equational theory as the original terms are temporarily replaced by special constants not occurring in the whole problem, thus reducing the case at hand to a problem that can be solved by special unification algorithms. The replaced subterms are then taken care of in a recursive call of the same process.
After some definitions and notation we shall present our algorithm, demonstrate its working by an example and prove its total correctness. Finally we shall compare our algorithm with those of Yelick and Tidén.

2. DEFINITIONS AND NOTATIONS

2.1 Terms and Substitutions

Unification theory rests upon the usual algebraic notions (see e.g. [Gr 79] [BS 81]) with the familiar concept of an algebra $\mathcal{A} = (\mathbf{A}, \mathbf{F})$ where $\mathbf{A}$ is the **carrier** and $\mathbf{F}$ is a family of **operators** given with their arities. For a given **congruence relation** ϱ the quotient algebra modulo ϱ is written as $\mathcal{A}_{/\varrho} = (\mathbf{A}_{/\varrho}, \mathbf{F})$.
Assuming that there is at least one constant (operator of arity 0) in $\mathbf{F}$ and a denumerable set of variables $\mathbf{V}$, we define $\mathbf{T}$, the set of **first order terms**, over $\mathbf{F}$ and $\mathbf{V}$, as the least set with (i) $\mathbf{V} \subseteq \mathbf{T}$, and if arity(f) = 0 for $f \in \mathbf{F}$ then $f \in \mathbf{T}$ and (ii) if $t_1, \ldots, t_n \in \mathbf{T}$ and arity(f) = n then $f(t_1 \ldots t_n) \in \mathbf{T}$. For a given term $t = g(t_1 \ldots t_n)$ the term $t_k, 1 \le k \le n$, is called an **immediate subterm** of t and t is the **immediate superterm** of t_k; the **leading function symbol** of t is g denoted as hd(t) = g. If t is a constant or a variable then hd(t) = t.
Let $\mathbf{V}(s)$ be the set of variables occuring in term s; a term s is **ground** if $\mathbf{V}(s) = \emptyset$.
As usual $\mathcal{T}$ denotes the algebra with carrier $\mathbf{T}$ and with operators, namely the term constructors corresponding to each operator of $\mathbf{F}$. $\mathcal{T}$ is called the absolutely free (term) algebra i.e. it just gives an algebraic structure to $\mathbf{T}$. If the carrier is the set of ground terms it is called the initial algebra [GT 78] or Herbrand Universe [Lo 78].

A **substitution** $\sigma: \mathbf{T} \to \mathbf{T}$ is an endomorphism on the term algebra $\mathcal{T}$ which is identical almost everywhere on $\mathbf{V}$ and can be represented as a finite set of pairs: $\sigma = \{ x_1 \leftarrow t_1, \ldots, x_n \leftarrow t_n \}$. The restriction $\sigma|_V$ of a substitution σ to a set of variables V is defined as $\sigma|_V x = \sigma x$ if $x \in V$ and $\sigma|_V x = x$ else.

$\mathbf{\Sigma}$ is the set of substitutions on $\mathcal{T}$ and ε the identity. The application of a substitution σ to a term $t \in \mathbf{T}$ is written as σt. The composition of substitutions is defined as the usual composition of mappings : $(\sigma \cdot \tau)t = \sigma(\tau t)$ for $t \in \mathbf{T}$.

Let	$DOM\sigma = \{ x \in \mathbf{V} \mid \sigma x \neq x \}$	(domain of σ)
	$COD\sigma = \{ \sigma x \mid x \in DOM\sigma \}$	(codomain of σ)
	$VCOD\sigma = \mathbf{V}(COD\sigma)$	(variables in codomain of σ)

If $VCOD\sigma = \emptyset$ then σ is a **ground substitution**.
A set of substitutions $\Sigma \subseteq \mathbf{\Sigma}$ is said to be **based** on a set of variables W away from $Z \supseteq W$ iff the following two conditions are satisfied

(i)	$DOM\sigma = W$	for all $\sigma \in \Sigma$
(ii)	$VCOD\sigma \cap Z = \emptyset$	for all $\sigma \in \Sigma$

In particular for based substitutions baed on some W we have $DOM\sigma \cap VCOD\sigma = \emptyset$ which is equivalent to the idempotence of σ, i.e. $\sigma \cdot \sigma = \sigma$. We shall use this property in the proofs later on.

2.2 Equational Logic and Unification

An **equation** $s = t$ is a pair of terms. A set of equations T is called an **equational theory** iff an equation e is in T whenever e is true in every model of T i.e. e is a consequence of T (or for short: $e \in T$ whenever $T \models e$). A set of axioms T of an equational theory T is a set of equations such that T is the least equational theory containing this set T. We sometimes say that the equational theory T is presented by T. For simplicity we do not distinguish between the equational theory and its presentation.
The equality $=_T$ generated by a set of equations T is the finest congruence over $\mathbf{T}$ containing all pairs $\sigma s = \sigma t$ for $s = t \in T$ and $\sigma \in \mathbf{\Sigma}$. (i.e. the $\mathbf{\Sigma}$-invariant congruence relation generated by T). The following is Birkhoffs well-known completeness theorem of equational logic [Bi 35]

Theorem 2.1: $T \models s = t$ iff $s =_T t$.

We shall sometimes use another derivation system for equational logic which has been useful in induction proofs (see e.g. McNulty [Mc 76]).

We define

$$s \longrightarrow_{\sigma,e} t$$

iff there exists an equation e of the form l = r in T and a substitution $\sigma \in \Sigma$ such that t results from s by replacing a subterm of s equal to σl by σr. By a derivation of $s =_T t$ we mean a finite sequence of steps $s_{i-1} \longrightarrow_{\sigma i,ei} s_i$

$$s = s_0 \longrightarrow_{\sigma 1,e1} s_1 \longrightarrow_{\sigma 2,e2} s_2 \longrightarrow_{\sigma 3,e3} \cdots \longrightarrow_{\sigma n,en} s_n = t$$

where $\sigma_i \in \Sigma$ and $e_i \in T$. If they are clear from the context we omit the indices σ and e. If we consider T as a directed rewrite system we have $=_T = \overset{*}{\longleftrightarrow}$ where $\overset{*}{\longleftrightarrow}$ is the reflexive, symmetric and transitive closure of $\longrightarrow$. Our definitions and notations are consistent with [Gr 79][HO 80][Mc 76] and [Ta 79].

We extend T-equality in $\mathbf{T}$ to the set of substitutions Σ by:

$$\sigma =_T \tau \qquad \text{iff} \quad \forall x \in V \quad \sigma x =_T \tau x .$$

If T-equality of substitutions is restricted to a set of variables W we write

$$\sigma =_T \tau \ [W] \quad \text{iff} \quad \forall x \in W \quad \sigma x =_T \tau x$$

and say σ and τ are **T-equal on W**.

A substitution τ is **more general than** σ **on W** (or σ is **a T-instance of** τ **on W**):

$$\sigma \leq_T \tau \ [W] \text{ iff } \exists \lambda \in \Sigma \quad \sigma =_T \lambda\tau \ [W].$$

Two substitutions σ,τ are called **T-equivalent on W**

$$\sigma \equiv_T \tau \ [W] \text{ iff } \sigma \leq_T \tau \ [W] \text{ and } \tau \leq_T \sigma \ [W].$$

Given two terms s, t and an equational theory T, a unification problem for T is denoted as

$$\langle s = t \rangle_T$$

We say $\sigma \in \Sigma$ is a solution of $\langle s = t \rangle_T$ (or σ is a T-unifier of s and t) iff $\sigma s =_T \sigma t$. For the set of all T-unifiers of s and t we write $U\Sigma_T(s, t)$. Without loss of generality we can assume that the unifiers of s and t are idempotent (if not, one can find an equivalent set of unifiers that is idempotent). For a given

unification problem $\langle s = t \rangle_T$, it is not necessary to compute the whole set of unifiers $U\Sigma_T(s, t)$, which is always recursively enumerable for a decidable theory T, but instead a smaller set useful in representing $U\Sigma_T$. Therefore we define $cU\Sigma_T(s, t)$, a **complete set of unifiers of s and t** on $W = \mathbf{V}(s, t)$ as:

(i) $cU\Sigma_T \subseteq U\Sigma_T$ (correctness)

(ii) $\forall \delta \in U\Sigma_T \quad \exists \sigma \in cU\Sigma_T: \quad \delta \leq_T \sigma\ [W]$ (completeness)

A **set of most general unifiers** $\mu U\Sigma_T(s, t)$ is a complete set with

(iii) $\forall \sigma, \tau \in \mu U\Sigma_T: \quad \sigma \leq_T \tau\ [W]$ implies $\sigma = \tau$ (minimality).

For technical reasons it turned out to be useful to have the following requirement: For a set of variables Z with $W \subseteq Z$

(iv) $\mu U\Sigma_T(s, t)$ (resp. $cU\Sigma_T(s, t)$) is based on W away from Z (protection of Z)

If conditions (i) - (iv) are fulfilled we say $\mu U\Sigma_T$ is a **set of most general unifiers away from Z** (resp. $cU\Sigma_T(s, t)$ is a **complete set of unifiers away from Z**) [PL72].

The set $\mu U\Sigma_T$ does not always exist [FH 83][Sc 86]; if it does then it is unique up to the equivalence $\equiv_T [W]$ (see [Hu 76][FH 83]). For that reason it is sufficient to generate just one $\mu U\Sigma_T$ as some representative of the equivalence class $[\mu U\Sigma_T]_{\equiv T}$.

Depending on the cardinality of the set of most general unifiers we can classify the equational theories into the following subclasses:

- a theory is **unitary** iff $\mu U\Sigma_T$ exists and $|\mu U\Sigma_T(s, t)| = 1$ for all s and t
- a theory is **finitary** iff $\mu U\Sigma_T$ exists and $|\mu U\Sigma_T(s, t)| < \infty$ for all s and t
- a theory is **infinitary** iff $\mu U\Sigma_T$ exists and $|\mu U\Sigma_T(s, t)| = \infty$ for some s and t
- a theory is **nullary** iff $\mu U\Sigma_T$ does not exist for some s and t

Sometimes it turned out to be useful to change the relation $\leq_T [W]$ used in the definition of completeness and minimality to $\leq_T [X]$ with $W \subseteq X \subseteq Z$. This procedure is justified by the so called "Fortsetzungslemma" as follows

Lemma 2.1: For two idempotent substitutions θ_1, θ_2 and the sets of variables $U \subseteq V$ with $DOM\theta_2 \subseteq U$ and $VCOD\theta_2 \cap V = \emptyset$:

$$\theta_1 \leq_T \theta_2 [U] \text{ iff } \theta_1 \leq_T \theta_2 [V].$$

Proof: Let $W = V \setminus U$ be the extension of the validity domain. By assumption there exists λ_U with $\theta_1 =_T \lambda_U\theta_2 [U]$. Since $VCOD\theta_2 \cap V = \emptyset$ we can find λ_U such that for all $x \in W$ $\lambda_U x = x$. Define $\lambda_W = \{ x \leftarrow \theta_1 x \mid x \in W \} = \theta_1|_W$ and let $\lambda = \lambda_W\lambda_U$. Then for $x \in U$ it is $\lambda\theta_2 x =_T \lambda_W\lambda_U\theta_2 x = \lambda_W\theta_1 x = \theta_1 x = \theta_1 x$ by definition of λ_W and λ_U and the idempotence of θ_1. For $x \in W$ it is $\lambda\theta_2 x = \lambda_W\lambda_U\theta_2 x =_T \lambda_W x = \theta_1 x$ since $\lambda_U x = x$ for $x \in W$ and $DOM\theta_2 \cap W = \emptyset$. Hence $\theta_1 =_T \lambda\theta_2 [U \cup W = V]$, i.e. $\theta_1 \leq_T \theta_2 [V]$. The other direction is trivial. ■

Another technical lemma which is useful for later proofs is the following:

Lemma 2.2: For idempotent substitutions δ, σ, τ and a set of variables V with $DOM\tau = \mathbf{V}(\sigma(V))$ and $VCOD\tau \cap (VCOD\sigma \cup V) = \emptyset$
(i) $DOM\tau\sigma = V \cup DOM\sigma$ and
(ii) if $\delta \leq_T \tau [\mathbf{V}(\sigma(V))]$ then $\delta\sigma \leq_T \tau\sigma [DOM\tau\sigma]$.

Proof: Using the previous lemma we have $\delta \leq_T \tau [\mathbf{V}(\sigma(V \cup DOM\sigma))]$ and hence $\delta\sigma \leq_T \tau\sigma [V \cup DOM\sigma]$ and $DOM\tau\sigma = V \cup DOM\sigma$. ■

A unification algorithm is called complete (and minimal) if it returns a correct and complete (and minimal) set of unifiers for every pair of terms.

2.3 Combination of Equational Theories

In this section we shall describe the equational theories for which we shall give a unification algorithm.
An equation $l = r$ is called **regular** iff $\mathbf{V}(l) = \mathbf{V}(r)$. It is called a **collapse axiom** iff it is of the form $x = t$ where t is a non-variable term. A set of equations is called **regular** iff all equations are regular, and **collapse free** iff it does not contain any collapse axioms. In [Ye 85] collapse free theories are called confined. A theory T is **consistent** iff the equation $x =_T y$ is not deducible in T.

Lemma 2.3: (i) An equational theory T is regular iff some presentation of T is regular.
(ii) A theory T is collapse free iff some presentation of T is collapse free.

Proof: (i) Suppose $s \longrightarrow_{\sigma,e} t$ and let $l = r$ be the equation e then $\mathbf{V}(\sigma l) = \mathbf{V}(\sigma r)$ since T is regular. As t differs form s only by a subterm with the same variables as the replaced subterm of s we have $\mathbf{V}(s) = \mathbf{V}(t)$.
(ii) Suppose $s \longrightarrow_{\sigma,e} x$ then $l = r$ must be a collapse axiom which is a contradiction. The lemma follows by induction on the length of a derivation of $s \xrightarrow{*} t$. ■

Let T be a presentation of an equational theory then $\mathbf{F}(T)$ is the set of function symbols and constants occurring in T. We sometimes call them interpreted function symbols or interpreted constants to distinguish them from the set $\mathbf{F}_\emptyset$ of function symbols and $\mathbf{C}_\emptyset$ of constants for which no equational theory is defined. We say these function symbols belong to the empty theory $\emptyset$ or are uninterpreted. A term t is **constrained** by a theory T iff $hd(t) \in \mathbf{F}(T)$ and we write $TH(t) = T$.
In the sequel we shall assume that T is the union of a set of presentations T_i whose set of function symbols are mutually disjoint, i.e. $T = \bigcup\{ T_i \mid 1 \le i \le n \}$ and $\mathbf{F}(T_i) \cap \mathbf{F}(T_j) = \emptyset$ for $1 \le i \ne j \le n$.
We say a subterm r is **alien** in s if it is not an uninterpreted constant or a variable, i.e. $r \notin \mathbf{C}_\emptyset \cup \mathbf{V}$, and if it is constrained by another theory than its immediate superterm, i.e. r is an immediate subterm of some subterm r' of s and $TH(r) \ne TH(r')$. By abuse of notation t is an alien subterm of t if $t \notin \mathbf{C}_\emptyset \cup \mathbf{V}$.
For a set S of terms we denote as **ALIEN(S)** a set of representatives of the T-equivalence classes of the alien subterms of S. Hence $s =_T t$ iff $s = t$ for all $s, t \in$ ALIEN(S).
We have to impose some restrictions on the subtheoies T_i of T in order for the algorithm to work:

(i) each equational theory T_i must be regular;
(ii) each equational theory T_i must not contain any collapse axioms;
(iii) each equational theory T_i must be consistent;
(iv) the wordproblem in the equational theory T must be decidable, i.e. it must be decidable whether $s =_T t$ for every $s, t \in \mathbf{T}$;
(v) each equational theory T_i must be unitary or finitary;
(vi) for each equational theory there must exist a complete unification algorithm for uninterpreted constants and variables, i.e. we can solve unification problems $\langle s = t \rangle_{T_i}$ where $s, t \in \mathbf{T}(\mathbf{F}(T_i) \cup \mathbf{C}_\emptyset, \mathbf{V})$ and $\mathbf{C}_\emptyset$ is a denumerable set of uninterpreted constants.

The first restriction is needed for the completeness and termination of the main algorithm. The second restriction is necessary in order to know which special unification algorithm is to be called. Regularity and the absence of collapse

axioms is inherited to the whole equational theory T by Lemma 2.3. The third restriction is obvious since otherwise the unification problem would be trivial. The next condition is heavily relied upon in our proofs and without this restriction any unification problem is senseless. The last restrictions are necessary since the main idea of our algorithm is to abstract subterms constrained by a different theory than the original terms to uninterpreted constants. We shall now formalize this abstraction process and give some useful lemmata about T-equal terms.

Given a set of constants $\mathbf{C}_{/\sim T} = \{ c_{[t]} \mid [t] \in \mathbf{T}(\mathbf{F}(T), \mathbf{V})_{/\sim T} \} \subseteq \mathbf{C}_{\emptyset}$ indexed by the equivalence classes modulo $=_T$, the replacement of the subterms of a term t not constrained by the theory of t can be described by the following recursive function

$$\text{C-abstract}_i : \mathbf{T}(\mathbf{F}(T), \mathbf{V}) \longrightarrow \mathbf{T}(\mathbf{F}(T_i) \cup \mathbf{C}_{\emptyset}, \mathbf{V})$$

with $\text{C-abstract}_i(t) = t$ if $t = x$ with $x \in \mathbf{V}$ or $t = c$ with $c \in \mathbf{C}_{\emptyset}$ and $\text{C-abstract}_i(t) = f(\text{C-abstract}_i(t_1) \ldots \text{C-abstract}_i(t_n))$ if $t = f(t_1 \ldots t_n)$ and $f \in \mathbf{F}(T_i)$ and $\text{C-abstract}_i(t) = c_{[t]}$ if $t = f(t_1 \ldots t_n)$ and $f \notin \mathbf{F}(T_i)$. We omit the indices of C-abstract_i if it is clear to which subalgebra we abstract. Note that we replace T-equal subterms by the same constant.

Let $TH(t) = T_i$ then we denote the abstracted term by $\underline{t} = \text{C-abstract}_i(t)$. The set $\text{I-ALIEN}(t) = \{t_1, \ldots, t_m\} \subset \text{ALIEN}(t)$ of immediate subterms constrained by another theory than t and replaced by some constants $c_{[t1]}, \ldots, c_{[tm]}$ denotes the set of **immediate alien** subterms and $\alpha = [t_1 \Leftarrow c_{[t1]}, \ldots, t_m \Leftarrow c_{[tm]}]$ the **subterm replacement** with $\underline{t} = \alpha t$. Now consider the inverse subterm replacement $\alpha^{-1} = [c_{[t1]} \Leftarrow t_1, \ldots, c_{[tm]} \Leftarrow t_m]$. If we treat the constants $c_{[t1]}, \ldots, c_{[tm]}$ in α^{-1} as '**special variables**' there is no need to formally distinguish between the subterm replacement α^{-1} and the substitution $\alpha = \{c_{[t1]} \leftarrow t_1, \ldots, c_{[tm]} \leftarrow t_m\}$. We then have $\alpha\underline{s} = \alpha\alpha s =_T s$. Note that the set I-ALIEN is again a set of representatives of the T-equivalence classes.

We now define the **theory height** of a term t as the maximal number of theory changes in that term:

$$h_T(t) = \begin{cases} 1 + \max\{ h_T(s) \mid s \in \text{I-ALIEN}(t) \} & \text{if } \text{I-ALIEN}(t) \neq \emptyset \text{ and} \\ 1 & \text{else} \end{cases}$$

and call terms whose theory height equals 1 **pure terms**. Note that for pure terms $t = \text{C-abstract}(t)$ and $\text{I-ALIEN}(t) = \emptyset$.

Next we collect some lemmata which are needed later on. All lemmas apply to

equational theories T which satisfy the above restrictions. The proofs are always induction proofs on the length of a derivation of $s \xrightarrow{*} t$. We only show the induction base.

Lemma 2.4: If $s =_T t$ then $TH(s) = TH(t)$.

Proof: Suppose $s \longrightarrow_{\sigma,e} t$ and let $l = r$ be the equation e in T. If $\sigma l = s$ then $t = \sigma r$ and $hd(s), hd(t) \in \mathbb{F}(T)$ since T is collapse free. If σl is equal to a subterm of s then $hd(s) = hd(t)$, i.e. s and t are constrained by the same theory. ■

As an immediate consequence we have the following corollary which is useful as an extended 'clash criterium' for the equational theories under consideration:

Corollary 2.1: If $TH(s) \neq TH(t)$ then s and t are not T-unifiable.

Lemma 2.5: If $s =_T t$ then $h_T(s) = h_T(t)$.

Proof: Again we consider one derivation step $s \longrightarrow_{\sigma,e} t$. Let $l = r$ be the equation e then $h_T(l) = h_T(r) = 1$ and since T is regular we have $h_T(\sigma l) = h_T(\sigma r)$. But now t only differs from s in a subterm with the same theory height and constrained by the same theory and therefore we have $h_T(s) = h_T(t)$. ■

We finally want to show that a complete set of T_i-unifiers for T_i-pure terms is a complete set of T-unifiers. The idea of the proof is to abstract the non-pure subterms in the codomain of an arbitrary unifier by constants to get a pure unifier which is more general than the original unifier. E. Tidén [Ti 85] showed the lemma for the more general case of nonregular theories with collapse axioms.

Lemma 2.6: Let s and t be pure terms with $TH(s) = TH(t) = T_i$. Then every complete set $cU\Sigma_{Ti}(s, t)$ of T_i-unifiers is a complete set of T-unifiers.

Proof: For a substitution σ we define in an analogue way the constant abstraction $\underline{\sigma}$ of σ by $\underline{\sigma}x = C\text{-abstract}_i(\sigma x) = \underline{\sigma x}$. For a T_i-pure term s it is obvious that $\underline{\sigma}s = \underline{\sigma s}$. Taking the subterm replacement $\beta = [\, c_{|t|} \leftarrow t \mid t \in ALIEN(COD\sigma)\,]$ we have $\sigma x =_T \beta\underline{\sigma}x$ for all $x \in DOM\sigma$.

We now show that for arbitrary terms s,t with $s =_T t$ we have $\underline{s} =_{Ti} \underline{t}$. As above the proof is by induction on the length of a derivation of $s =_T t$. Given a single

derivation step $s \longrightarrow_{\sigma,e} t$. If s is a variable then t must be a variable since T is consistent. Now let $s = s(s_1, ..., s_m)$ with I-ALIEN(s) = $\{s_1, ..., s_m\}$ and $t = t(t_1, ..., t_n)$ with I-ALIEN(t) = $\{t_1, ..., t_n\}$ and hence $\underline{s} = s(q_{[s1]}, ..., q_{[sm]})$ and $\underline{t} = t(q_{[t1]}, ..., q_{[tn]})$. We have to distinguish two cases.

CASE 1: The subterm that is replaced by the derivation step is a subterm of sj for $1 \le j \le m$. Then by definition we have $t = s(s_1, ..., s_j', ..., s_m)$ and $\underline{t} = s(q_{[s1]}, ..., q_{[sj']}, ..., q_{[sm]})$ and $\underline{s} = \underline{t}$ since $c_{[sj]} = q_{[sj']}$.

CASE 2: The subterm that is replaced starts in $\underline{s}$. Then we can deduce a derivation $\underline{s} \longrightarrow_{\underline{\sigma},e} \underline{t}$ and e is an equation in T_i, i.e. $\underline{s} =_{Ti} \underline{t}$.

To finish the proof let θ be an arbitrary unifier of the T_i-pure terms s and t with $DOM\theta = \mathbf{V}(s, t)$ then $\underline{\theta}$ is a T_i-unifier of s and t since $\underline{\theta}s = \underline{\theta s} =_{Ti} \underline{\theta t} = \underline{\theta}t$. Since $cU\Sigma_{Ti}(s, t)$ is a complete set of T_i-unifiers there exists $\sigma \in cU\Sigma_{Ti}(s, t)$ and λ with $\underline{\theta} =_{Ti} \lambda\sigma\,[\mathbf{V}(s, t)]$. We have to show that $\theta \le_T \sigma\,[\mathbf{V}(s, t)]$: for $x \in \mathbf{V}(s, t)$ $\theta x =_T \mathfrak{b}(\underline{\theta}x) =_T \mathfrak{b}(\lambda\sigma x) = \mathfrak{b}(\lambda)(\sigma x)$ where $\mathfrak{b}(\lambda)$ is defined by $\mathfrak{b}(\lambda)x = \mathfrak{b}(\lambda x)$. The equations are easy to see since $\mathfrak{b}$ acts like a substitution. Hence $cU\Sigma_{Ti}(s, t)$ is a complete set of T-unifiers. ■

Finally we want to introduce a new class of theories called **simple theories**: a theory T is simple iff $\mu U\Sigma_T(x, t) = \emptyset$ for all terms t and all variables x occurring in t. This is equivalent to the fact that a term is never T-equal to one of its subterms. It is easy to see that theories with finite congruence classes are always simple. The converse does not hold. For simple theories some of the proofs are easier and especially in the algorithms a lot of recursive calls can be dropped.

3. THE ALGORTIHM

Before we state the algorithm we need some notation: given two arbitrary terms s and t we defined the set I-ALIEN(s, t) of toplevel subterms that are constrained by another theory than s resp. t. We then need the set of subproblems of s and t

$$SP(s, t) := \{ (s', t') \mid s', t' \in \text{I-ALIEN}(s, t) \text{ and } TH(s') = TH(t') \}$$

i.e. those pairs which are potentially unifiable.

We do not explicitly consider the details of basing the unifier on $\mathbf{V}(s, t)$ away from some set of variables containing $\mathbf{V}(s, t)$ since it would only complicate the notation. The proofs demonstrating that the unifiers are based on $\mathbf{V}(s, t)$ away from Z are not difficult. The special algorithms for the particular theories T_i are

denoted by T_i-UNIFY.

In our main algorithm we shall use an operation called the merge $\sigma * \alpha$ of two substitutions σ and α. Essentially the merge is the set of most general instances of the two substitutions and is defined in chapter 4 along with some properties. For a set Σ of substitutions we abbreviate $\{\sigma \cdot \delta \mid \sigma \in \Sigma\}$ by $\Sigma \cdot \delta$ and $\bigcup\{\sigma * \delta \mid \sigma \in \Sigma\}$ by $\Sigma * \delta$.

The main idea of the unification algorithm is first to unify the constant abstractions of the original terms. In order to obtain finally the unifiers for the original terms we have to merge the unifiers of the abstracted terms with the substitution reversing the abstraction, where the newly introduced constants are now regarded as variables (STEP 5). Once we have solved the unification problem for the constant abstraction we have to apply it recursively to all subterms that have been 'abstracted away' (STEP 6).

FUNCTION UNIFY

INPUT: Two arbitrary terms s and t

STEP 1: if $s \in \mathbf{V}$ or $t \in \mathbf{V}$ then $\Sigma_T(s, t) :=$ VARIABLE-UNIFY(s, t)

STEP 2: elseif $s \in \mathbf{C}$ or $t \in \mathbf{C}$ then $\Sigma_T(s, t) :=$ CONSTANT-UNIFY(s, t)

STEP 3: elseif TH(s) ≠ TH(t) then $\Sigma_T(s, t) := \emptyset$

STEP 4: elseif I-ALIEN(s, t) = $\emptyset$ then $\Sigma_T(s, t) :=$ TH(s)-UNIFY(s, t)
else let s, t be the constant abstraction and α be the corresponding substitution reversing the abstraction
in

STEP 5: $\Sigma_T(s, t) :=$ TH(s)-UNIFY(s, t) $* \alpha$

STEP 6: forall (s', t') ∈ SP(s, t) do
forall σ' ∈ UNIFY (s', t') do
$\Sigma_T(s, t) := \Sigma_T(s, t) \cup$ UNIFY(σ's, σ't)$\cdot\sigma'$
od
od

OUTPUT: The set of unifiers $\Sigma_T(s, t)$ away from $Z \supseteq \mathbf{V}(s, t)$

ENDOF UNIFY

There remain the cases where at least one term is either a variable or a constant.

FUNCTION CONSTANT-UNIFY

INPUT: Two non-variable terms s and t where at least one is a constant

STEP 1: if $s \in \mathbf{C}_{\emptyset}$ and $t \in \mathbf{C}_{\emptyset}$ then
if $s = t$ then $\Sigma_T(s, t) := \{\varepsilon\}$ else $\Sigma_T(s, t) := \emptyset$
STEP 2: elseif $s \in \mathbf{C}_{\emptyset}$ then $\Sigma_T(s, t) := \emptyset$
STEP 3: elseif $t \in \mathbf{C}_{\emptyset}$ then $\Sigma_T(s, t) := \emptyset$
STEP 4: elseif $TH(s) \neq TH(t)$ or $h_T(s) \neq h_T(t)$
then $\Sigma_T(s, t) := \emptyset$ else $\Sigma_T(s, t) :=$ TH(s)-UNIFY(s, t)

OUTPUT: The set of unifiers $\Sigma_T(s, t)$ away from $Z \supseteq \mathbf{V}(s, t)$

ENDOF CONSTANT-UNIFY

Provided the T_i-unification algorithms terminate, the termination of CONSTANT-UNIFY is no problem.

Lemma 3.1: For a constant and a non-variable term CONSTANT-UNIFY returns a correct and complete set of unifiers.

Proof: For STEP 1 - 3 the lemma is obvious.
For STEP 4 let s be a constant in $\mathbf{F}(T_i)$ and hence $h_T(s) = 1$. If t is constrained by another theory than T_i then s and t are not unifiable by Corollary 2.1. so we have $TH(s) = TH(t) = T_i$. Now suppose $h_T(t) > 1$ and $\sigma \in U\Sigma_T(s, t)$ then $1 < h_T(t) \leq h_T(\sigma t) = h_T(\sigma s) = h_T(s) = 1$ which is a contradiction. Hence s and t are T_i-pure terms and the assertion follows from Lemma 2.6. ■

The variable case is more complicated if the considered theories are not simple.
FUNCTION VARIABLE-UNIFY

INPUT: Two terms s and t where at least one is a variable

STEP 1: if $s \in \mathbf{V}$ and $t \in \mathbf{V}$ then
if $s = t$ then $\Sigma_T(s, t) := \{\varepsilon\}$ else $\Sigma_T(s, t) := \{\{s \leftarrow t\}\}$
STEP 2: elseif $s \in \mathbf{V}$ then $\Sigma_T(s, t) :=$ VARIABLE-TERM-UNIFY(s, t)
STEP 3: elseif $t \in \mathbf{V}$ then $\Sigma_T(s, t) :=$ VARIABLE-TERM-UNIFY(t, s)

OUTPUT: The set of unifiers $\Sigma_T(s, t)$ away from $Z \supseteq \mathbf{V}(s, t)$

ENDOF VARIABLE-UNIFY

The following example shows the difficulty in the variable-term-case: let $T_1 = \{ f(f(x, y), y) = f(x, y) \}$, $T_2 = \emptyset$ with $g \in \mathbf{F}(T_2)$ and $T = T_1 \cup T_2$ be the equational theories and $\langle u = f(u, g(v)) \rangle_T$ the unification problem. Then there exists a unifier $\sigma = \{ u \leftarrow f(u', g(v)) \}$ which will be computed by the following algorithm in STEP 5.

FUNCTION VARIABLE-TERM-UNIFY

INPUT: A variable x and a non-variable term t

STEP 1: if $x \notin \mathbf{V}(t)$ then $\Sigma_T(x, t) := \{\{ x \leftarrow t \}\}$
else let $TH(t) = T_i$
in
STEP 2: if T_i is simple then $\Sigma_T(x, t) := \emptyset$
STEP 3: elseif 1-ALIEN(t) = $\emptyset$ then $\Sigma_T(x, t) := T_i$-UNIFY(x, t)
else let t be the constant abstraction and α be the corresponding substitution reversing the abstraction
in
STEP 4: if $x \in$ VCODα then $\Sigma_T(x, t) := \emptyset$
else
STEP 5: $\Sigma_T(x, t) := T_i$-UNIFY(x, t) $* \alpha$
STEP 6: forall $(s', t') \in$ SP(t) do
forall $\sigma' \in$ UNIFY (s', t') do
$\Sigma_T(x, t) := \Sigma_T(x, t) \cup$ UNIFY$(\sigma' x, \sigma' t) \cdot \sigma'$
od
od

OUTPUT The set of unifiers $\Sigma_T(x, t)$ away from $Z \supseteq \mathbf{V}(s, t)$

ENDOF VARIABLE-TERM-UNIFY

In STEP 1 and STEP 2 termination, correctness and completeness are trivial. By Lemma 2.6 we know that in STEP 3 a correct and complete set of unifiers is returned and termination is trivial. In STEP 4 correctness and completeness follows from the next lemma, whereas termination is inherited from termination of T_i-UNIFY. For the other steps termination, correctness and completeness is shown as for the main algorithm.

Lemma 3.2: If $x \in \mathbf{V}(t)$ and x does not occur in some immediate alien subterm of t, i.e. $x \in VCOD\alpha$ where α reverses the constant abstraction of t, i.e. $\alpha\underline{t} =_T t$, then $U\Sigma_T(x, t) = \emptyset$.

Proof: Suppose there exists a σ such that $\sigma x =_T \sigma t$. Since $x \in VCOD\alpha$ there exists a subterm s in t that is constrained by another theory as t and $x \in \mathbf{V}(s)$. Hence σx is a subterm of σs and therefore $h_T(\sigma x) \leq h_T(\sigma s) < h_T(\sigma t)$ which is a contradiction to $\sigma x =_T \sigma t$ and Lemma 2.5. ■

The lemma is a generalized 'occur-check' for non-simple theories. With the theories T_1 and T_2 of the above example the unification problem $\langle u = f(v\ g(u)) \rangle_T$ has no solutions since u occurs in the immediate alien subterm g(u).

4. THE MERGE OF SUBSTITUTIONS

The algorithm of the previous section used an operation called the merge of substitutions or unification of substitutions. Given two substitutions σ and τ we say σ and τ are T-unifiable iff there exists λ such that $\lambda\sigma =_T \lambda\tau$. Then λ is called a T-unifier of σ and τ. The sets $U\Sigma_T(\sigma, \tau)$, $cU\Sigma_T(\sigma, \tau)$, $\mu U\Sigma_T(\sigma, \tau)$ are defined accordingly. If $\mu U\Sigma_T(\sigma, \tau)$ exists then $\sigma * \tau := \{ \lambda\sigma \mid \lambda \in \mu U\Sigma_T(\sigma, \tau) \}$ is called a merge of σ and τ.
In the special situation of the previous algorithms there are certain constraints on the two substitutions which we want to exploit in the computation of the merge: first $\tau = \{x_1 \leftarrow t_1, \ldots, x_m \leftarrow t_m\}$ is a unifier of the constant abstractions which are pure terms and $\alpha = \{c_1 \leftarrow r_1, \ldots, c_n \leftarrow r_n\}$ reverses a constant abstraction. Note that the newly introduced constants are now considered as special variables. Hence we have

(i) $DOM\tau \cap DOM\alpha = \emptyset$
(ii) $VCOD\tau \cap VCOD\alpha = \emptyset$.

The last equation holds since $COD\tau$ only contains the special variables and new variables not occurring in s and t, whereas $VCOD\alpha \subseteq \mathbf{V}(s, t)$.
In the following lemmata we shall always assume that τ and α satisfy these conditions. We show that unifying two substitutions is the same as unifying two termlists (σ unifies $(s_1, \ldots, s_n)$ and $(t_1, \ldots, t_n)$ iff $\sigma s_i =_T \sigma t_i$ for $1 \leq i \leq n$, or equivalently σ unifies the set of termpairs $\{ (s_i, t_i) \mid 1 \leq i \leq n \}$).

Lemma 4.1: For τ and α as above:

$$U\Sigma_T(\tau, \alpha) = U\Sigma_T((x_1, \ldots, x_m, c_1, \ldots, c_n), (t_1, \ldots, t_m, r_1, \ldots, r_n))$$

Proof: Let $\lambda\alpha =_T \lambda\tau$ and $x \in DOM\tau$ then $x = x_i$ and $\lambda x_i = \lambda\alpha x_i =_T \lambda\tau x_i = \lambda t_i$ for $1 \le i \le m$ and for $x \in DOM\alpha$ then $x = c_j$ and $\lambda c_j = \lambda\tau c_j =_T \lambda\alpha c_j = \lambda r_j$ for $1 \le j \le n$. Hence λ is a unifier of the termlists. Conversely let θ be a unifier of the termlists then for $x \in DOM\tau$ it is $x = x_i$ with $1 \le i \le m$ and $\theta\alpha x_i = \theta x_i =_T \theta t_i = \theta\tau x_i$ and for $x \in DOM\alpha$ $x = c_j$ with $1 \le j \le n$ and $\theta\tau c_j = \theta c_j =_T \theta r_j = \theta\alpha c_j$. ■

Since for simple equational theories $\mu U\Sigma_T(x, t) = \emptyset$ if $x \in \mathbf{V}(t)$ and $\mu U\Sigma_T(x, t) = \{(x \leftarrow t)\}$ otherwise the most general unifier (if it exists) of the termlists in Lemma 4.1 is just the composition of $\{x_i \leftarrow t_i\}$ and $\{c_j \leftarrow r_j\}$. Hence we define $\tau_0 = \tau$ and $\tau_j = \sigma_j\tau_{j-1}$ and $\sigma_j = \{c_j \leftarrow \tau_{j-1}r_j\}$ for $1 \le j \le n$.

Lemma 4.2: In a simple theory T the termlists $(x_1, \ldots, x_m, c_1, \ldots, c_n)$ and $(t_1, \ldots, t_m, r_1, \ldots, r_n)$ are T-unifiable iff τ_n is the most general unifier.

Proof: We show the "only if" direction by induction on n: let θ be a unifier then $\theta \le_T \tau_n [V_n]$ with $V_n = \mathbf{V}(x_1, \ldots, x_m, c_1, \ldots, c_n, t_1, \ldots, t_m, r_1, \ldots, r_n)$.

Base step: Since $\theta x_i =_T \theta t_i = \theta\tau x_i$ it is $\theta =_T \theta\tau$ and hence we get $\theta \le_T \tau_0 = \tau [V_0]$.

Induction step: Since $\theta \le_T \tau_n [V_n]$ there exists λ such that $\theta =_T \lambda\tau_n [V_n]$ and with Lemma 2.1 we have $\theta =_T \lambda\tau_n [V_{n+1}]$. But then $\lambda c_{n+1} = \lambda\tau_n c_{n+1} = \theta c_{n+1} =_T \theta r_{n+1} = \lambda\tau_n r_{n+1}$ (since $\tau_n c_{n+1} = c_{n+1}$) i.e. λ unifies c_{n+1} and $\tau_n r_{n+1}$. Hence $c_{n+1} \notin \mathbf{V}(\tau_n r_{n+1})$ and $\mu U\Sigma_T(c_{n+1}, \tau_n r_{n+1}) = \{\sigma_{n+1}\}$ with $\sigma_{n+1} = \{c_{n+1} \leftarrow \tau_n r_{n+1}\}$ and $\lambda \le_T \sigma_{n+1} [\mathbf{V}(c_{n+1}, \tau_n r_{n+1})]$. Therefore we have $\lambda \le_T \sigma_{n+1} [\mathbf{V}(c_{n+1}, \tau_n r_{n+1}) \cup VCOD\tau_n]$ and $\theta =_T \lambda\tau_n \le_T \sigma_{n+1}\tau_n = \tau_{n+1} [DOM\tau_{n+1}]$ with Lemma 2.2. But then $\theta \le_T \tau_{n+1} [V_{n+1}]$ again with Lemma 2.1.

Hence τ_{n+1} exists and is the most general unifier of the termlists.

The other direction is trivial. ■

Corollary 4.1: Let T be a simple theory. If the substitutions τ and α are T-unifiable then τ_n is a single most general unifier of τ and α and $\tau * \alpha = \{\tau_n\}$ with $\tau_n = \tau_n\alpha = \tau_n\tau$ is the merge of τ and α. If the substitutions τ and α are not T-unifiable then there exists j, $1 \le j \le n$ with $c_j \in \mathbf{V}(\tau_{j-1}r_j)$.

For non-simple theories the unification problem $\langle c_j = \tau_{j-1} r_j \rangle_T$ is not trivial as we have seen in chapter 3. It can produce a set of substitutions different from $\{c_j \leftarrow \tau_{j-1} r_j\}$. Hence we cannot directly state the set of most general unifiers of τ and α. But we shall show that it is sufficient to compute the set of most general unifiers of two smaller termlists. We later use these termlists to show the termination of our main algorithm.

Lemma 4.3: (i) If τ and α are T-unifiable then the termlists $(\tau c_1, \ldots, \tau c_n) = (c_1, \ldots, c_n)$ and $(\tau r_1, \ldots, \tau r_n) = (s_1, \ldots, s_n)$ are T-unifiable and vice versa.

(ii) For $\lambda \in \mu U\Sigma_T(\,(c_1, \ldots, c_n), (s_1, \ldots, s_n)\,)$ there exists $\theta \in \mu U\Sigma_T(\tau, \alpha)$ with $\lambda\tau \equiv_T \theta\,[\,\mathbf{V}(\tau) \cup \mathbf{V}(\alpha)\,]$ and vice versa.

Proof: (i) Let λ be a unifier of τ and α then λ is a unifier of the termlists since $\lambda c_i = \lambda\tau c_i =_T \lambda\alpha c_i = \lambda\alpha\alpha c_i = \lambda\alpha r_i =_T \lambda\tau r_i = \lambda s_i$ for $1 \le i \le n$. The first equation holds by condition (4.1). Conversely let λ be a unifier of the termlists then for $x \in DOM\tau$ it is $\lambda\tau\tau x = \lambda\tau x = \lambda\tau\alpha x$ (the last equation again by (4.1)) and for $c_i \in DOM\alpha$ it is $\lambda\tau\alpha c_i = \lambda\tau r_i =_T \lambda\tau c_i = \lambda\tau\tau c_i$, hence $\lambda\tau$ unifies τ and α.

(ii) If $\lambda \in \mu U\Sigma_T(\,(c_1, \ldots, c_n), (s_1, \ldots, s_n)\,)$ then $\lambda\tau$ is a unifier of τ and α. Hence there exists $\theta \in \mu U\Sigma_T(\tau, \alpha)$ with $\lambda\tau \le_T \theta\,[V_1]$ where $V_1 = \mathbf{V}(\tau) \cup \mathbf{V}(\alpha)$. But by (i) θ is a unifier of $(c_1, \ldots, c_n)$ and $(s_1, \ldots, s_n)$ and therefore there exists $\lambda' \in \mu U\Sigma_T(\,(c_1, \ldots, c_n), (s_1, \ldots, s_n)\,)$ with $\theta \le_T \lambda'\,[V_2]$ where $V_2 = \mathbf{V}(\,(c_1, \ldots, c_n), (s_1, \ldots, s_n)\,) = \mathbf{V}(\,\tau(\mathbf{V}(\alpha))\,) \subseteq V_1$. Hence we have $\lambda\tau \le_T \theta\,[V_2]$ and $\theta \le_T \lambda'\,[V_2]$ and since $\lambda\tau = \lambda\,[V_2]$ by $V_2 \cap DOM\tau = \emptyset$ it is $\lambda = \lambda\tau \le_T \theta \le_T \lambda'\,[V_2]$ and by the minimality $\lambda = \lambda'$. Hence $\lambda = \lambda\tau \equiv_T \theta\,[V_2]$. In addition we have $\theta\tau =_T \theta$ since for $x \in DOM\tau$ $\theta\tau x =_T \theta\alpha x = \theta x$ by (4.1) and $\lambda\tau \le_T \theta \le_T \lambda\,[V_1]$. But as $VCOD\tau \subseteq V_1$ it is $\theta\tau \le_T \lambda\tau\,[V_1]$ and therefore we finally have $\lambda\tau \equiv_T \theta\,[V_1]$.

Now we consider the mapping $\Phi: \mu U\Sigma_T(\tau, \alpha) \longrightarrow \mu U\Sigma_T(\,(c_1, \ldots, c_n), (s_1, \ldots, s_n)\,)$ with $\Phi(\theta) = \lambda_\theta$ with $\lambda_\theta\tau \equiv_T \theta\,[V_1]$. First we show that Φ is a mapping. Suppose $\Phi(\theta) = \lambda_1$ and $\Phi(\theta) = \lambda_2$ then $\lambda_1 \equiv_T \lambda_2\,[V_2]$ and hence $\lambda_1 = \lambda_2$. For the injectivity of Φ let $\Phi(\theta_1) = \Phi(\theta_2)$ then it is $\theta_1 \equiv_T \lambda_{\theta 1}\tau = \lambda_{\theta 2}\tau \equiv_T \theta_2\,[V1]$ and by the minimality $\theta_1 = \theta_2$. By the above surjectivity of Φ is obvious, hence Φ is bijective and the converse holds. ∎

We shall now describe an algorithm that computes a complete set $cU\Sigma_T((s_1, \ldots, s_n), (t_1, \ldots, t_n))$ of unifiers of two termlists. We assume the existence of our main algorithm UNIFY since LIST-UNIFY and UNIFY are mutually

recursive.

FUNCTION LIST-UNIFY

INPUT: Two arbitrary termlists $(s_1, ..., s_n)$ and $(t_1, ..., t_n)$ of length n

$\Sigma_0 := \{\varepsilon\}$
for i := 1 ,..., n do
$\Sigma_i := \{ \tau_i \circ \sigma_{i-1} \mid \sigma_{i-1} \in \Sigma_{i-1}$ and $\tau_i \in UNIFY(\sigma_{i-1}s_i, \sigma_{i-1}t_i) \}$
od

OUTPUT The set of unifiers Σ_n away from $Z \supseteq \mathbf{V}((s_1, ..., s_n), (t_1, ..., t_n))$

ENDOF LIST-UNIFY

Lemma 4.4: If UNIFY is a correct and complete algorithm which terminates then LIST-UNIFY is a correct and complete algorithm for two termlists.

Proof: Let $\sigma_n \in \Sigma_n$ then $\sigma_n = \tau_n \circ ... \circ \tau_1$ where τ_i is a correct unifier of $\tau_{i-1} \circ ... \circ \tau_1 s_i$ and $\tau_{i-1} \circ ... \circ \tau_1 t_i$ by the hypothesis for UNIFY and hence $\sigma_n s_i =_T \sigma_n t_i$ for all $1 \le i \le n$, i.e. LIST-UNIFY is a correct algorithm.
Now let θ be a unifier of $(s_1, ..., s_n)$ and $(t_1, ..., t_n)$ then θ is a unifier of s_1 and t_1. By the completeness of UNIFY there exists $\tau_1 \in \Sigma_1 = UNIFY(s_1, t_1)$ with $\theta \le_T \tau_1[\mathbf{V}(s_1, t_1)]$. By Lemma 2.1 we have $\theta =_T \lambda_1\tau_1[W]$ with $W = \mathbf{V}(s_1, ..., s_n, t_1, ..., t_n)$ and hence λ_1 is unifier of $(\tau_1 s_1, ..., \tau_1 s_n)$ and $(\tau_1 t_1, ..., \tau_1 t_n)$. By induction it is easy to see that $\theta =_T \lambda_n\tau_n \circ ... \circ \tau_1[W]$, i.e. LIST-UNIFY is complete. ■

Termination of our main algorithm and hence of LIST-UNIFY is shown in section 6. In order to compute the merge it is sufficient to take a complete (not necessarily minimal) set of unifiers of τ and α. We then have some redundant unifiers which can be eliminated in a minimizing step.

5. AN EXAMPLE

Given the unification problem

$$\langle f(x\ f(x\ f(y\ g(x\ u)))) \;=\; f(z\ f(g(a\ b)\ g(a\ b))) \rangle_T$$

where $T = T_1 \cup T_2$ with T_1 the theory of associativity and commutativity (AC) for the function symbol f denoted as AC_f and T_2 the empty theory $\emptyset$ for g, a, b are two uninterpreted constants. To ease the notation we drop the unnecessary function symbols f and represent the terms as abelian strings. The immediate alien subterms for $s = f(x^2\ y\ g(x\ u))$ and $t = f(z\ g(a\ b)\ g(a\ b))$ are I-ALIEN(s, t) = { g(x u), g(a b) }. The only subproblem is therefore (s', t') = (g(x u), g(a b)) with the most general unifier $\sigma' = \{x \leftarrow a, u \leftarrow b\}$. The constant abstractions of s and t are $\underline{s} = f(x^2\ y\ c_1)$ and $\underline{t} = f(z\ c_2^2)$ with $\alpha = \{c_1 \leftarrow g(x\ u), c_2 \leftarrow g(a\ b)\}$. The set of most general unifiers for $\underline{s}$ and $\underline{t}$ is (see [St 81][HS 85]):

$$\begin{aligned}\mu U\Sigma_T(\underline{s}, \underline{t}) \;=\; \{ &\{x \leftarrow f(z_1\ c_2),\ y \leftarrow z_2,\ z \leftarrow f(z_1\ z_2^2\ c_1)\},\\ &\{x \leftarrow c_2,\ y \leftarrow u_2,\ z \leftarrow f(u_2^2\ c_1)\},\\ &\{x \leftarrow v_1,\ y \leftarrow f(v_2 c_2^2),\ z \leftarrow f(v_1 v_2^2 c_1 c_2)\},\\ &\{x \leftarrow w_1,\ y \leftarrow f(c_2^2),\ z \leftarrow f(w_1 c_1 c_2)\}\ \}.\end{aligned}$$

Merging these unifiers with α we get:

$$\begin{aligned}\mu U\Sigma_T(\underline{s}, \underline{t})*\alpha \;=\; \{ &\\ &\{x \leftarrow f(z_1\ g(a\ b)),\ y \leftarrow z_2,\ z \leftarrow f(z_1\ z_2^2\ g(\ f(z_1\ g(a\ b))\ z_3\ )),\ u \leftarrow z_3\ \},\\ &\{x \leftarrow g(a\ b),\ y \leftarrow u_2,\ z \leftarrow f(u_2^2\ g(\ g(a\ b)\ u_3)),\ u \leftarrow u_3\ \},\\ &\{x \leftarrow v_1,\ y \leftarrow f(v_2\ g(a\ b)\ g(a\ b)),\ z \leftarrow f(v_1\ v_2^2\ g(a\ b)\ g(v_1 v_3)),\ u \leftarrow v_3\ \},\\ &\{x \leftarrow w_1,\ y \leftarrow f(g(a\ b)\ g(a\ b)),\ z \leftarrow f(w_1\ g(a\ b)\ g(w_1\ w_3),\ u \leftarrow w_3\ \}\ \}\end{aligned}$$

The only unifier of (s', t') is $\sigma = \{x \leftarrow a, u \leftarrow b\}$ and hence $\sigma s = f(a^2\ y\ g(a\ b))$ and $\sigma t = f(z\ g(a\ b)\ g(a\ b))$ have the set of most general T-unifiers $\mu U\Sigma_T(\sigma s, \sigma t) = \{\tau_1, \tau_2\}$ with $\tau_1 = \{y \leftarrow f(x_1\ g(a\ b)),\ z \leftarrow f(x_1\ a^2)\}$ and $\tau_2 = \{y \leftarrow g(a\ b),\ z \leftarrow f(a^2)\}$. Hence

$$\tau_1\sigma = \{x \leftarrow a, y \leftarrow f(x_1\ g(a\ b)),\ z \leftarrow f(x_1\ a^2), u \leftarrow b\}$$
$$\tau_2\sigma = \{x \leftarrow a, y \leftarrow g(a\ b),\ z \leftarrow f(\ a^2), u \leftarrow b\}$$

are two more most general T-unifier of s and t. So finally we have

$$\mu U\Sigma_T(s, t) \;=\; \mu U\Sigma_T(\underline{s}, \underline{t})*\alpha \cup \{\tau_1\sigma, \tau_2\sigma\}.$$

6. TERMINATION

In order to prove the termination of Stickel's AC-unification algorithm, F. Fages [Fa 84] gave a complexity measure for two terms which can be used in the more general case [Ye 85][Ti 85]. We shall use a slightly modified version of that measure for showing that our algorithm terminates. The following are prerequisites for the definition of that measure.

We define the immediate function symbols of a term r in a term s by

$$Op(r, s) = \{ hd(t) \mid t \text{ is an immediate superterm of } r \text{ and a subterm of } s \}.$$

We write $Op(r, S)$ for $\cup\{ Op(r, s) \mid s \in S \}$ and we omit the parentheses in $Op(r, \{s, t\})$ and write $Op(r; s, t)$. The set of theories by which s is constrained in t is denoted as

$$T\text{-}Op(r, s) = \{ T' \mid f \in \mathbf{F}(T') \text{ and } f \in Op(r, s) \}$$

and $T\text{-}Op(r, S)$ as $\cup\{ T\text{-}Op(r, s) \mid s \in S \}$. The set of shared variables of a set of terms S is then defined as the set of those variables constrained by at least two different theories

$$\mathbf{V}_s(S) = \{ x \in \mathbf{V}(S) \mid\mid T\text{-}Op(x, S) \mid > 1 \}.$$

The complexity of a pair of terms for two terms s and t, which we shall use to show the termination of our algorithm is:

$$\mathbb{C}(s, t) = (\nu, \tau) \quad \text{where} \quad \nu = |\mathbf{V}_s(s, t)| \quad \text{and} \quad \tau = |ALIEN(s, t)|.$$

where the set ALIEN(s, t) of alien subterms is as defined in section 2.3. To illustrate the definitions we take the example of the previous chapter: let $s = f(x\ f(x\ f(y\ g(x\ u))))$ and $t = f(z\ f(g(a\ b)\ g(a\ b)))$ then $Op(x; s, t) = \{f, g\}$ and $T\text{-}Op(x; s, t) = \{AC_T, \emptyset\}$. Since the other variables only occur immediately under one function symbol we have $\mathbf{V}_s(s,t) = \{x\}$. The set of alien subterms of s and t is $ALIEN(s, t) = \{s, t, g(x\ u), g(a\ b)\}$. Note that an uninterpreted constant is not an alien subterm in our definition whereas in the definitions of Fages, Tiden and Yelick it is.

Taking the lexicographic order on the complexities we obtain a Noetherian order. For this section we always assume that one of the given terms s and t is not a variable or an uninterpreted constant since the other cases were treated in section 3. The following lemma states that the complexity of alien subterms is less than the complexity of the terms itself.

Lemma 6.1: Given two terms s and t. If $s', t' \in ALIEN(s, t) \cup \mathbf{V}(s, t)$ are proper subterms of s or t and not both are variables then

$$\mathbb{C}(s', t') < \mathbb{C}(s, t).$$

Proof: Let $C(s, t) = (\nu, \tau)$ and $C(s', t') = (\nu', \tau')$. Since $V(s', t') \subseteq V(s, t)$ and $Op(x; s', t') \subseteq Op(x; s, t)$ we have $V_s(s', t') \subseteq V_s(s, t)$ and hence $\nu' \leq \nu$. If $\nu' = \nu$ we have to show that $\tau' < \tau$. Now consider the mapping Φ: ALIEN(s', t') $\longrightarrow$ ALIEN(s, t) with $\Phi(r') := r$ and $r =_T r'$. Then Φ is an inclusion and since at least s or t is not contained in ALIEN(s', t') it is $\tau' < \tau$. ■

To show the termination of our main algorithm UNIFY we have to show $C(\sigma s, \sigma t) < C(s, t)$ if σ unifies some immediate alien subterms of s and t, i.e. we have to show that unifiers produced by the algorithm decrease the complexity of the original terms.
We say a substitution σ is **monotone for s and t** iff $C(\sigma s, \sigma t) \leq C(s, t)$ and **strictly monotone for s and t** iff $C(\sigma s, \sigma t) < C(s, t)$. In the following lemmata we show the monotony of certain substitutions. We call a substitution σ **alien for s and t** iff $\sigma = \{x \leftarrow r\}$ with $x \in V(s, t)$, $r \in$ ALIEN(s, t) and $x \notin V(r)$.

Lemma 6.2: If a substitution σ is alien for two terms s and t then σ is monotone for s and t.

Proof: Let $C(\sigma s, \sigma t) = (\nu_\sigma, \tau_\sigma)$, $C(s, t) = (\nu, \tau)$ and $\sigma = \{x \leftarrow r\}$. Since $V_s(\sigma s, \sigma t) \subseteq V_s(s, t)$ we have $\nu_\sigma \leq \nu$.
If $\nu_\sigma < \nu$ we are done. If $\nu_\sigma = \nu$ we want to show that $\tau_\sigma \leq \tau$. We construct an injective mapping Ψ from ALIEN($\sigma s, \sigma t$) to ALIEN(s, t) with $\Psi(p) = p'$ and $\sigma p' =_T p$. For $p = r$ we define $\Psi(r) = r$ with $\sigma r = r$ since $r \in$ ALIEN(s, t) and $x \notin V(r)$. For $p \neq r$ in ALIEN($\sigma s, \sigma t$) there exists an $p' \in$ ALIEN(s, t) with $\sigma p' =_T p$; we define $\Psi(p) = p'$. Hence we have $\Psi(p) = p'$ with $\sigma p' =_T p$.
The injectivity of Ψ is easy to see: $q_1 = \Psi(p_1) = \Psi(p_2) = q_2$ implies $p_1 =_T \sigma q_1 = \sigma q_2 =_T p_2$ and by definition of ALIEN $p_1 = p_2$ (note that for s' and t' in ALIEN(s, t) we have: if $s' =_T t'$ then $s' = t'$). Hence $|\text{ALIEN}(\sigma s, \sigma t)| = \tau_\sigma \leq \tau = |\text{ALIEN}(s, t)|$. ■

A substitution σ is called **T-pure for s and t** iff $DOM\sigma \subseteq V(s, t)$, $VCOD\sigma \cap V(s, t) = \emptyset$ and the following two conditions are satisfied

- $T \in T\text{-}Op(x; s, t)$ for all $x \in DOM\sigma$ and
- σx is a T-pure term (i.e. $COD\sigma \subseteq T(F(T) \cup C_\emptyset, V)$).

We can asssume that the algorithms T-UNIFY only generate T-pure substitutions for T-pure terms; we simply speak of pure substitutions and pure terms if there are no ambiguities.

Lemma 6.3: If a substitution σ is T-pure for two terms s and t then σ is monotone for s and t.

Proof: Let $\mathcal{C}(\sigma s, \sigma t) = (\nu_\sigma, \tau_\sigma)$, $\mathcal{C}(s, t) = (\nu, \tau)$ and σ be T-pure for s and t. We shall construct an injective mapping Φ from $\mathbf{V}_s(\sigma s, \sigma t)$ to $\mathbf{V}_s(s, t)$. Let $x \in \mathbf{V}_s(\sigma s, \sigma t)$: if $x \notin \mathrm{VCOD}\sigma$ then $x \in \mathbf{V}_s(s, t)$ and we choose $\Phi(x) = x$. Note that $x \notin \mathrm{DOM}\sigma$. For $x \in \mathrm{VCOD}\sigma$ consider the set of variables $V_1 = \{y \mid x \in \mathbf{V}(\sigma y)\}$. Suppose that for all $y \in V_1$ $\sigma y \neq x$. Since σ is T-pure x occurs only under the function symbols of T in σs and σt ($x \notin \mathbf{V}(s, t)$), which is a contradiction. Now consider the set $V_2 = \{y \in V_1 \mid \sigma y = x\}$. But then there exists $y \in V_2 \cap \mathbf{V}_s(s, t)$ and we can choose $\Phi(x) = y$. Otherwise again all y would only occur under the function symbols of T in s and t which contradicts $x \in \mathbf{V}_s(\sigma s, \sigma t)$. Hence we have $\sigma y = x$ for all x with $\Phi(x) = y$. The injectivity of Φ is easy to see: $y_1 = \Phi(x_1) = \Phi(x_2) = y_2$ implies $x_1 = \sigma y_1 = \sigma y_2 = x_2$. Hence $\nu_\sigma \leq \nu$.

If $\nu_\sigma < \nu$ nothing is to show. If $\nu_\sigma = \nu$ the mapping Φ is bijective and there exists the inverse Φ^{-1} from $\mathbf{V}_s(s, t)$ to $\mathbf{V}_s(\sigma s, \sigma t)$ with $\Phi^{-1}(x) = \sigma x = y$. We want to show that $\tau_\sigma \leq \tau$. Again we construct a mapping Ψ from ALIEN($\sigma s, \sigma t$) to ALIEN(s, t) with $\Psi(p) = p'$ and $\sigma p' =_T p$.

By the bijectivity of Φ for all $x \in \mathrm{DOM}\sigma$ with $\sigma x \notin \mathbf{V}$ we have $x \notin \mathbf{V}_s(s, t)$. Let p be in ALIEN($\sigma s, \sigma t$) then p is not introduced by σ since for all x with $\sigma x \notin \mathbf{V}$ T-Op(x, s, t) = {T} and $\{\mathrm{hd}(\sigma x)\} \in T$ or σx is an uninterpreted constant. Hence there must exist a subterm $p' \in$ ALIEN(s, t) with $\sigma p' =_T p$ and we define $\Psi(p) = p'$. The injectivity of Ψ is again easy to see: $q_1 = \Phi(p_1) = \Phi(p_2) = q_2$ implies $p_1 =_T \sigma q_1 = \sigma q_2 =_T p_2$ and as above $p_1 = p_2$. Hence we have $|\mathrm{ALIEN}(\sigma s, \sigma t)| = \tau_\sigma \leq \tau = |\mathrm{ALIEN}(s, t)|$. ■

In the termination proof we shall often use the fact that a substitution which is pure or alien for s' and t' is pure or alien for s and t as well if s' and t' are alien subterms of s and t:

Lemma 6.4: Let s and t be two terms and let $s', t' \in \mathrm{ALIEN}(s, t) \cup \mathbf{V}(s, t)$ be proper subterms of s or t and not both variables. If σ is alien for s' and t' then σ is alien for s and t. If σ is a pure substitution for s' and t' and $\mathrm{VCOD}(\sigma) \cap \mathbf{V}(s, t) = \emptyset$ then σ is pure for s and t.

The proof is obvious. The next lemma is the key for the termination proof.

Lemma 6.5: Let σ be a pure or alien substitution for s and t which are not both variables and let $s', t' \in \mathrm{ALIEN}(s, t)$ be proper and distinct subterms of s or t. If σ unifies s' and t' then σ is strictly monotone.

Proof: By the lemmata 6.2 and 6.3 we have $C(\sigma s, \sigma t) \leq C(s, t)$. Suppose $C(\sigma s, \sigma t) = C(s, t)$, i.e. the mapping Ψ from ALIEN($\sigma s, \sigma t$) to ALIEN(s, t) constructed in the above proofs is bijective. Hence the inverse Ψ^{-1} exists with $\Psi^{-1}(p) = q$ and $\sigma p =_T q$. But then $\Psi^{-1}(s') = \Psi^{-1}(t')$ since σ unifies s' and t' which contradicts the bijectivity of Ψ^{-1}. ■

Since in the algorithm the substitutions are built up by composition we say a substitution σ is **elementary for the problem** $\langle s = t \rangle_T$ (or short for s and t) iff it is a composition of pure or alien substitutions, i.e. $\sigma = \sigma_n \sigma_{n-1} \ldots \sigma_1$ where σ_i is pure or alien for $\sigma_{i-1} \ldots \sigma_1 s$ and $\sigma_{i-1} \ldots \sigma_1 t$ for $2 \leq i \leq n$ and σ_1 is pure or alien for s and t. By an induction argument we have:

Lemma 6.6: (i) If σ is an elementary substitution for s and t then σ is monotone for s and t.

(ii) If in addition σ unifies two distinct and proper subterms s', t' $\in$ ALIEN(s, t) then σ is strictly monotone for s and t.

(iii) Let s', t' $\in$ ALIEN(s, t) $\cup$ $\mathbf{V}$(s, t) be proper subterms of s or t and not both are variables. If σ is elementary for s' and t' then σ is elementary for s and t provided the newly introduced variables are away from $\mathbf{V}$(s, t).

To summarize: first we introduced monotone substitutions. Then we have shown that alien and pure substitutions (the elements of the generated unifiers) and their composition are monotone.
The main termination proof is by Noetherian induction on the complexities of the input terms. We show UNIFY(s, t) terminates and generates substitutions elementary for s and t. Therefore it is sufficient:

for two terms s and t the complexity of the input terms s' and t' in every recursive call of UNIFY in UNIFY(s, t) is smaller than the complexity of the original terms i.e. $C(s', t') < C(s, t)$ (hence we can apply the induction hypothesis) and the substitutions generated by UNIFY(s, t) are elementary for the original terms s and t.

First we prove that the merge operation terminates by showing that every call of UNIFY in LIST-UNIFY($(c_1, \ldots, c_n), (\tau r_1, \ldots, \tau r_n)$) terminates and yields substitutions which are elementary for s and t.

Theorem 6.1: Given s and t with the corresponding abstractions $\underline{s}$ resp. $\underline{t}$, the substitutions $\alpha = \{c_1 \leftarrow r_1, \ldots, c_n \leftarrow r_n\}$ reversing the abstraction and $\tau \in$ TH(s)-UNIFY($\underline{s}, \underline{t}$) unifying the abstractions, then the merge operation $\tau * \alpha$ terminates and the merges are elementary for s and t.

Proof: We show by induction on n that for $i = 1, \ldots, n$ $C(\tau_{i-1}c_i, \tau_{i-1}r_i) < C(s, t)$ and τ_i is elementary for s and t with $\tau_0 = \tau$, $\tau_i = \sigma_i\tau_{i-1}$ and $\sigma_i \in \text{UNIFY}(\tau_{i-1}c_i, \tau_{i-1}r_i)$. Since $\tau \in \text{TH(s)-UNIFY}(\underline{s}, \underline{t})$ $\tau_0 = \tau$ is pure and hence elementary for s and t; w.l.o.g. we can assume $\text{VCOD}\tau \supseteq \{c_1, \ldots, c_n\}$.

Base step: first we have $\tau_0 c_1 = c_1 \in \mathbf{V}(\tau_0 s, \tau_0 t)$. Since $r_1 \in \text{I-ALIEN}(s, t)$ it is $\tau_0 r_1 \in \text{I-ALIEN}(\tau_0 s, \tau_0 t)$ and with Lemma 6.1 we have $C(\tau_0 c_1, \tau_0 r_1) < C(\tau_0 s, \tau_0 t) \leq C(s, t)$. To show that τ_1 is elementary for s and t we distinguish two cases:

CASE 1: $c_1 \in \mathbf{V}(\tau_0 r_1)$: Let $\sigma_1 \in \text{UNIFY}(\tau_0 c_1, \tau_0 r_1)$ then by the main Noetherian induction hypothesis σ_1 is elementary for $\tau_0 c_1$ and $\tau_0 r_1$ and by Lemma 6.6 (iii) σ_1 is elementary for $\tau_0 s$ and $\tau_0 t$.

CASE 2: $c_1 \notin \mathbf{V}(\tau_0 r_1)$: with $\sigma_1 = \{c_1 \leftarrow \tau_0 r_1\}$ we have $\{\sigma_1\} = \text{UNIFY}(\tau_0 c_1, \tau_0 r_1)$ and with Lemma 6.4 σ_1 is alien for $\tau_0 s$ and $\tau_0 t$.

Summarizing $\tau_1 = \sigma_1\tau_0$ is elementary for s and t.

Induction step: If $\tau_{n-1}c_n = c_n$ then the proof is analogue to the base step. Now let $\tau_{n-1}c_n \neq c_n$. If $\text{TH}(\tau_{n-1}c_n) = \text{TH(s)}$ then by Corollary 2.1 $\tau_{n-1}c_n$ and $\tau_{n-1}r_n$ are not unifiable since $\text{TH}(\tau_{n-1}r_n) \neq \text{TH}(\tau_{n-1}c_n)$. Now if $\text{TH}(\tau_{n-1}c_n) \neq \text{TH(s)}$ $\tau_{n-1}c_n$ is in $\text{ALIEN}(\tau_{n-1}s, \tau_{n-1}t)$ by $\text{TH(s)} \in \text{T-Op}(c_i, \tau_0 s, \tau_0 t)$. By the same argument as above $\tau_{n-1}r_n \in \text{ALIEN}(\tau_{n-1}s, \tau_{n-1}t)$. Hence with Lemma 6.1 and the induction hypothesis (τ_{n-1} is elementary for s and t) we get that $C(\tau_{n-1}c_n, \tau_{n-1}r_n) < C(\tau_{n-1}s, \tau_{n-1}t) \leq C(s, t)$. Hence by the main Noetherian induction $\sigma_n \in \text{UNIFY}(\tau_{n-1}c_n, \tau_{n-1}r_n)$ is elementary for $\tau_{n-1}c_n$ and $\tau_{n-1}r_n$ and with Lemma 6.6 (iii) for $\tau_{n-1}s$ and $\tau_{n-1}t$. Finally we have $\tau_n = \sigma_n\tau_{n-1}$ is elementary for s and t. ■

We now state the two main theorems:

Theorem 6.2: For a variable x and a term t VARIABLE-TERM-UNIFY(x, t) terminates and generates substitutions which are elementary for x and t.

The proof is analogue to that of the next theorem:

Theorem 6.3: For two terms s and t at least one of which is not a variable, UNIFY(s, t) terminates and generates substitutions which are elementary for s and t.

Proof: In STEP 1 termination is established by Theorem 6.2 as well as the property of the generated substitutions being elementary.

For STEP 2 termination of CONSTANT-UNIFY is obvious. The empty substitution and unifiers of pure terms are elementary for s and t.
For STEP 3 nothing is to prove.
In STEP 4 termination follows from termination of TH(s)-UNIFY and the generated substitutions are elementary since they are T-pure by the remark before Lemma 6.3.
As in STEP 4 TH(s)-UNIFY($\underline{s}$, $\underline{t}$) terminates. By Theorem 6.1 the merge operation terminates and the merges are elementary for s and t. So the theorem is shown for STEP 5.
In STEP 6 let $(s', t') \in SP(s,t)$ then by Lemma 6.1 $C(s', t') < C(s, t)$ and hence UNIFY(s', t') terminates and the substitutions $\sigma' \in$ UNIFY(s', t') are elementary for s' and t' by the main induction hypothesis. With Lemma 6.6 (iii) σ' is elementary for s and t since $s', t' \in$ ALIEN(s, t). Since σ' unifies s' and t' we know by Lemma 6.6 (ii) that σ' is strictly monotone for s and t, i.e. $C(\sigma's, \sigma't) < C(s, t)$. Hence by induction hypothesis UNIFY($\sigma's$, $\sigma't$) terminates and produces substitutions σ'' which are elementary for $\sigma's$ and $\sigma't$, i.e. $\sigma = \sigma''\cdot\sigma'$ is elementary for s and t. ■

7. CORRECTNESS AND COMPLETENESS

All the proofs of this chapter are by induction on the recursion depth of the term pair which is a Noetherian order by the last chapter. The set of substitutions returned by the unification algorithm is a correct set of unifiers:

Theorem 7.1: For $s, t \in \mathbf{T}$ UNIFY(s, t) returns a correct set of unifiers.

Proof: Consider each step in UNIFY in succesion:
STEP 1: The theorem follows from the theorem below (correctness for the variable-term-case).
STEP 2: Correctness is obvious (confer chapter 3).
STEP 3: Nothing is to show.
STEP 4: By assumption T_i-UNIFY is correct.
STEP 5: Let $\underline{s}$, $\underline{t}$ be the constant abstractions of s and t with $\alpha\underline{s} =_T s$ and $\alpha\underline{t} =_T t$.
By induction hypothesis let τ be a correct T-unifier of $\underline{s}$ and $\underline{t}$. Since for $\theta \in \tau * \alpha$ $\theta = \lambda\alpha =_T \lambda\tau$ for some λ we have (using the idempotence of τ and α):

$$\theta\cdot\tau =_T \lambda\cdot\tau\cdot\tau = \lambda\cdot\tau =_T \lambda\cdot\alpha = \lambda\cdot\alpha\cdot\alpha =_T \theta\cdot\alpha$$

Hence
$$\begin{aligned} \theta s &=_T \theta\cdot\alpha\underline{s} \\ &=_T \theta\cdot\tau\underline{s} \\ &=_T \theta\cdot\tau\underline{t} && \text{by assumption} \\ &=_T \theta\cdot\alpha\underline{t} \\ &=_T \theta t\,. \end{aligned}$$

STEP6: Let (s', t') be a subproblem of s and t. By induction hypothesis let σ' be a correct unifier of s' and t' and σ'' be a correct T-unifier of σ's and σ't. Then for $\sigma = \sigma'' \circ \sigma' \in \Sigma_T$ we have

$$\begin{aligned} \sigma s &= \sigma''(\sigma' s) \\ &=_T \sigma''(\sigma' t) && \text{by hypothesis} \\ &= \sigma t \end{aligned}$$

■

Theorem 7.2: Let $x \in \mathbf{V}$ and $t \in \mathbf{T}$ then VARIABLE-TERM-UNIFY(x, t) returns a correct and complete set of unifiers.

The correctness proof is analogous to the one above, completeness is shown as below. The following theorem shows that the main algorithm returns a complete set of unifiers. The technical lemmata can be found below the main proof.

Theorem 7.3: Let s, t be terms and let θ be a T-unifier of s and t. Then there exists $\sigma \in \Sigma_T(s, t)$ (returned by UNIFY(s, t)) such that

$$\theta \leq_T \sigma \ [V] \quad \text{with } V = \mathbf{V}(s, t).$$

Proof: Again we consider each step in turn.
STEP 1: The theorem follows from the next theorem (completeness for the variable-term-case).
STEP 2: By Lemma 3.1.
STEP 3: By Corollary 2.1.
STEP 4: By Lemma 2.6.
STEP 5: Now I-ALIEN(s, t) $\neq \emptyset$ and assume for all $(s', t') \in SP(s, t)$ it is $\theta s' \neq_T \theta t'$ (else STEP 6 applies). Then by Lemma 7.3 there exists $\underline{\theta}$ with $\underline{\theta}\,\underline{s} =_T \underline{\theta}\,\underline{t}$ and $\theta^* \in \underline{\theta} \circ \alpha$ with $\theta \leq_T \theta^*$ [V], where $\underline{s} = \mathfrak{a}s$ and $\underline{t} = \mathfrak{a}t$ are the constant abstractions of s and t and α the substitution reversing the constant abstraction $\mathfrak{a}$. By Lemma 2.6 there exists $\theta' \in \Sigma_T(\underline{s}, \underline{t})$ (returned by TH($\underline{s}$)-UNIFY($\underline{s}$, $\underline{t}$)) such that

$$\underline{\theta} \leq_T \theta' \ [\underline{V}] \quad \text{where } \underline{V} = \mathbf{V}(\underline{s}, \underline{t}).$$

Using Lemma 2.1 we get

$$\underline{\theta} \leq_T \theta' \ [V].$$

With Lemma 7.4 we have for $\sigma \in \theta' \circ \alpha \subseteq \Sigma_T(s, t)$ (returned by UNIFY(s, t))

$$\theta \leq_T \sigma \ [V].$$

STEP 6: In this step the subproblems are considered and moreover there exists $(s', t') \in SP(s, t)$ with $\theta s' =_T \theta t'$. By Noetherian induction there exists $\sigma' \in \Sigma_T(s', t')$ (returned by UNIFY(s', t')) such that

$$\theta \leq_T \sigma' \ [V'] \quad \text{with } V' = \mathbf{V}(s', t').$$

In other words there exists λ with $\theta =_T \lambda\sigma' \ [V]$ using Lemma 2.1.

But then λ is a unifier of σ's and σ't. By Noetherian induction there exists $\sigma'' \in \Sigma_T(\sigma's, \sigma't)$ (returned by UNIFY($\sigma's, \sigma't$)) with

$$\lambda \leq_T \sigma'' \ [V''] \quad \text{with } V'' = \mathbf{V}(\sigma's, \sigma't)$$

and by Lemma 2.1 and 2.2 we obtain

$$\lambda\sigma' \leq_T \sigma''\sigma' \ [V]$$

and hence with $\sigma := \sigma''\sigma'$ there exists $\sigma \in \Sigma_T(s, t)$ (returned by UNIFY(s, t)) such that $\theta \leq_T \sigma \ [V]$. ■

While this completes the main result of this paragraph, some technical lemmata remain to be shown stating the existence of certain substitutions in STEP 5. Regarding the situation in STEP 5 we have two terms s and t constrained by the same theory, θ a unifier of s and t and for all subproblems $(s', t') \in SP(s, t)$ it is $\theta s' \neq_T \theta t'$. Let $\{r_1, ..., r_n\}$ be the immediate alien subterms of s and t, $\alpha = \{c_1 \leftarrow r_1, ..., c_n \leftarrow r_n\}$ and $\alpha_\theta = \{c_1 \leftarrow \theta r_1, ..., c_n \leftarrow \theta r_n\} = \theta\alpha|_{DOM\alpha}$. Since $\theta r_i \neq_T \theta r_j$ for $i \neq j$, $1 \leq i, j \leq n$ by assumption let $\mathfrak{a} = [r_1 \Leftarrow c_1, ..., r_n \Leftarrow c_n]$ and $\mathfrak{a}_\theta = [\theta r_1 \Leftarrow c_1, ..., \theta r_n \Leftarrow c_n]$ be the corresponding subterm replacements. If $\theta = \{x_1 \leftarrow p_1, ..., x_m \leftarrow p_m\}$ we define $\underline{\theta} = \{x_1 \leftarrow \mathfrak{a}_\theta p_1, ..., x_m \leftarrow \mathfrak{a}_\theta p_m\}$. Furthermore we denote by $\underline{s} = \mathfrak{a}s$ and $\underline{t} = \mathfrak{a}t$ the constant abstractions of s and t and by $V = \mathbf{V}(s, t)$ the set of variables in s and t.

Lemma 7.1: $\underline{\theta}(\mathfrak{a}s) = \mathfrak{a}_\theta\theta s$ and $\underline{\theta}(\mathfrak{a}t) = \mathfrak{a}_\theta\theta t$.

Proof: We only show the first equation for all subterms r in $\underline{s} = \mathfrak{a}s$.
For $r \in \mathbf{C}_\emptyset$ we distinguish the cases $r \neq c_i$ and $r = c_i$ for $1 \leq i \leq n$. For the first case $\underline{\theta}r = \mathfrak{a}_\theta\theta r$ is obvious. For $r = c_i$ we have $\underline{\theta}c_i = c_i$ and there exists a subterms r' in s with $r' =_T r_i$ and $\mathfrak{a}_\theta\theta r' =_T \mathfrak{a}_\theta\theta r_i = c_i$. Now let $r = x \in \mathbf{V}$. If $x \notin DOM\underline{\theta} = DOM\theta$ then $\underline{\theta}x = x = \theta x = \mathfrak{a}_\theta\theta x$ and for $x \in DOM\theta$ it is $\underline{\theta}x = \mathfrak{a}_\theta\theta x$. Since there occur no immediate alien subterms in $\underline{s}$ we have for all subterms r of $\underline{s}$ with $r \notin \mathbf{C}_\emptyset \cup \mathbf{V}$ $\underline{\theta}r = \mathfrak{a}_\theta\theta r$. ■

Lemma 7.2: For all terms q not containing $c_1, ..., c_n$: $\theta\alpha\mathfrak{a}_\theta q =_T \theta q$.

Proof: Suppose there exists a subterm r in q with $r =_T \theta r_i$ then r is replaced by $\mathfrak{a}_\theta$ to c_i. Applying $\theta\alpha$ to that c_i we again have θr_i. If $r \neq_T \theta r_i$ we have $\alpha\mathfrak{a}_\theta r = r$ since q does not contain any of the c_i and hence $\theta\alpha\mathfrak{a}_\theta q =_T \theta q$. ■

Lemma 7.3: $\underline{\theta}$ is a unifier of $\underline{s}$ and $\underline{t}$ and there exists $\theta^* \in \underline{\theta}\ast\alpha$ with

$$\theta \leq_T \theta^* \ [V] \quad \text{and} \quad V = \mathbf{V}(s, t).$$

Proof: First $\underline{\theta}$ is a unifier of $\underline{s}$ and $\underline{t}$ since by Lemma 7.1 and the fact that θ unifies s and t we have

$$\underline{\theta}\,\underline{s} = \underline{\theta}(\alpha s) = \alpha_\theta(\theta s) =_T \alpha_\theta(\theta t) = \underline{\theta}(\alpha t) = \underline{\theta}\,\underline{t}.$$

We now show that α and $\underline{\theta}$ are unifiable, i.e. the merge exists. Using Lemma 4.1 we define $h_1 = (x_1, \ldots, x_m, c_1, \ldots, c_n)$ and $h_2 = (\alpha_\theta p_1, \ldots, \alpha_\theta p_m, r_1, \ldots, r_n)$ then

$$\begin{aligned}\theta\alpha h_2 &= (\theta\alpha\alpha_\theta p_1, \ldots, \theta\alpha\alpha_\theta p_m, \theta\alpha r_1, \ldots, \theta\alpha r_n)\\ &=_T (\theta p_1, \ldots, \theta p_m, \theta r_1, \ldots, \theta r_n) \qquad \text{by Lemma 7.2}\\ &= (\theta\alpha x_1, \ldots, \theta\alpha x_m, \theta\alpha c_1, \ldots, \theta\alpha c_n) = \theta\alpha h_1.\end{aligned}$$

Hence $\theta\alpha \le_T \lambda\,[W]$ where λ is a most general unifier of α and $\underline{\theta}$ and W are the variables of α and $\underline{\theta}$. Therefore we have with $V \subseteq W$ and $\theta^* =_T \lambda\alpha =_T \lambda\tau$

$$\theta = \theta\alpha \le_T \lambda\alpha = \theta^*\,[V]. \qquad \blacksquare$$

Lemma 7.4: Let θ' be a unifier of $\underline{s}$ and $\underline{t}$ with $\underline{\theta} \le_T \theta'\,[V]$ and $VCOD\theta' \cap V = \emptyset$ then there exists a $\sigma \in \theta' * \alpha$ with $\theta \le_T \sigma\,[V]$.

Proof: We assume w.l.o.g. that $DOM\theta' = V$ (if not define $\theta' x = z$ for $x \in V \setminus DOM\theta'$ and z is a new variable which does not occur in the problem). Since $\underline{\theta} \le_T \theta'\,[V]$ there exists δ with $\underline{\theta} =_T \delta\theta'\,[V]$ and

(1) $\quad DOM\delta \cap V = \emptyset \quad$ and $\quad DOM\delta \subseteq VCOD\theta'$.

Furthermore using $VCOD\theta' \cap V = \emptyset$ we have

(2) $\quad DOM\delta \cap \mathbf{V}(\alpha) = \emptyset$.

We show that θ' and α are unifiable. Using again Lemma 4.1 we define $g_1 = (x_1, \ldots, x_m, c_1, \ldots, c_n)$ and $g_2 = (\theta' x_1, \ldots, \theta' x_m, r_1, \ldots, r_n)$. Now we have $\delta g_1 = g_1 = h_1$ by (1) and (2) (confer for the definition of h_1 and h_2 the proof of the last lemma) and $\delta g_2 = h_2$ by (2). Since h_1 and h_2 are unifiable by $\theta\alpha$ g_1 and g_2 are unifiable by $\theta\alpha\delta$ and therefore there exists $\sigma \in \theta' * \alpha$ with $\theta\alpha\delta \le_T \sigma\,[V]$. Hence with $\theta\alpha\delta = \theta\,[V]$ we have $\theta \le_T \sigma\,[V]$. $\blacksquare$

8. CONCLUSION

We presented a general unification algorithm that combines unification algorithms for regular finitary collapse free equational theories. Correctness, completeness, and termination are shown. Hence the combination of regular finitary collapse free equational theories is again finitary. The algorithm is not minimal, but the redundant unifiers can be eliminated in a minimizing step. Our method does not apply to theories with collapse axioms: for example given an idempotent function symbol f, i.e. the equational theory $I = \{ f(x\,x) = x \}$, an uninterpreted function symbol g, and the problem $\langle g(x, f(x, y)) = g(a, a) \rangle_{I \cup \emptyset}$, our algorithm would not find a unifier since the constant abstraction

$\langle g(x, c) = g(a, a) \rangle_{I \cup \emptyset}$ is not unifiable and there does not exist any further subproblem. But the original equation is solvable by the substitution $\sigma = \{x \leftarrow a, y \leftarrow a\}$. The reason is that in equational theories with collpase axioms terms can collapse to variables by instantiation. It is an open problem to find a terminating unification algorithm for the whole class of finitary theories.
Given a special unification algorithm for a regular collapse free theory we can extend this algorithm at once to handle uninterpreted function symbols by our method. To get an efficient implementation however we are not forced to compute the whole set of subproblems as defined in the abstract algorithm. Depending on the theory and the variables in the considered problem, that algorithm can be improved by only taking a subset of SP(s, t) in the iterative step. For common equational theories it is an open problem to find such sets.
The combination of unification algorithms for regular collpsefree theories as proposed by E. Tidén and K. Yelick are based on the same method: both abstract the immmediate alien subterms to variables, unify the variable-pure abstracted terms and then merge the resulting unifiers with the substitution reversing the abstraction. Our algorithm however uses a different approach. Just as Yelick's algorithm was motivated by the AC-unification algorithm of Stickel [St 81] and its extension by Fages [Fa 84] we started our work on extending the AC-unification algorithm of Livesay and Siekmann [LS 76] resulting in a unification algorithm for AC-function symbols and uninterpreted function symbols [HS 85]. This algorithm avoids the notoriously inefficient process of variable abstraction and its redundancy by a reduction of the case at hand to a variable **and** constant equation. These advantages carry over to our general approach: unifying the variable abstractions of the example in section 5 results in 69 unifiers [St 81], each of which has to be merged with the substitution reversing the variable abstraction. As we have seen, unifying the constant abstraction of the example yields only four unifiers, which have to be merged. In the iteration we also have to solve much simpler terms in order to compute the other two unifiers. Another point is that pure variable term pairs are almost always unifiable. Hence our constant abstraction process reduces the unifiability of the abstracted terms in comparison to the variable abstractions and so reduces the number of merges.

Acknowledgement: I would like to express my gratitude to Hans Jürgen Bürckert and Jörg Siekmann for their patience in endless discussions. Their support and constructive criticism have contributed much to the present form of this paper. I am also grateful to Manfred Schmidt-Schauß for a thorough reading of earlier draft of this paper.

REFERENCES:

[Bi 35] Birkhoff, G., 'On the Strucutre of Abstract Algebra', Proc. Cambrigde Phil. Soc., Vol. 31,433-454, (1935)

[BS 81] Burris, S. and Sankappanavar, H.P., 'A Course in Universal Algebra', Springer-Verlag,(1981)

[Fa 84] Fages, F., 'Associative-Commutative Unification', in Proc. of 7th CADE (ed. R.E. Shostak), Springer-Verlag, LNCS 170, 194-208, (1984)

[FH 83] Fages, F. and Huet, G., 'Unification and Matching in Equational Theories', Proc. of CAAP'83 (ed. G. Ausiello and M. Protasi), Springer-Verlag, LNCS 159, 205-220, (1983)

[GT 78] Goguen, J. A., Thatcher, J.W. and Wagner, E. G., 'An Initial Algebra Approach to the Specification, Correctness and Implementation of Abstract Data Types', in 'Current Trends in Programming Methodology, Vol.4, Data Structuring' (ed. R. T. Yeh), Prentice Hall, (1978)

[Gr 79] Grätzer, G., 'Universal Algebra', Springer-Verlag, (1979)

[HO 80] Huet, G. and Oppen, D. C., 'Equations and Rewrite Rules: A Survey', in 'Formal Languages: Perspectives and Open Problems (ed R. Book), Academic Press, (1980)

[HS85] Herold, A. and Siekmann, J., 'Unification in Abelian Semigroups', MEMO SEKI-85-III-KL, Universität Kaiserslautern,(1985)

[Hu 76] Huet, G., 'Résolution d'équations dans des langages d'ordre 1, 2 ,...., ω', Thèse de doctorat d'état, Université Paris VII, (1976)

[Ki 85] Kirchner, C., Methodes et outils de conception systematique d'algorithmes d'unification dans les théories équationelles', Thèse de doctorat d'état, Université de Nancy 1, (1985)

[Lo 78] Loveland, D., 'Automated Theorem Proving', North-Holland, (1978)

[LS 76] Livesay, M. and Siekmann, J. , 'Unification of Sets and Multisets', Universität Karlsruhe, Techn. Report, (1976)

[Mc 76] McNulty, G., 'The Decision Problem for Equational Bases of Algebras Annals of Mathematical Logic 10, 193-259, (1976)

[Ro 65] Robinson, J. A., 'A Machine-Oriented Logic Based on the Resolution Principle', JACM 12, Nº. 1, 23-41, (1965)

[Sc 86] Schmidt-Schauß, M., 'Unifcation under Associativity and Idempotence is of Type Nullary', MEMO SEKI, Universität Kaiserslautern, (1986)

[Si 84] Siekmann, J., 'Universal Unification', in Proc. of 7th CADE (ed R. E. Shostak), Springer-Verlag, LNCS 170, 1-42, (1984)

[St 81] Stickel, M.E., A Unification Algorithm for Associative-Commutative Functions', JACM 28, Nº. 3, 423-434, (1981)

[Ta 79] Taylor, W., 'Equational Logic', Houston Journal of Mathematics 5, (1979)

[Ti 85] Tidén, E., 'Unifiaction in Combinations of Theories with Disjoint Sets of Function Symbols', Royal Institute of Technology, Department of Computing Science, S-100 44 Stockholm, Sweden, (1985)

[Ye 85] Yelick, K., 'Combining Unification Algorithms for Confined Regular Equational Theories',in Proc. of 'Rewriting Techiques and Applications' (ed J.-P. Jouannaud), Springer-Verlag, LNCS 202, 365-380, (1985)