Unification in Extensions of Shallow Equational Theories

Florent Jacquemard*, Christoph Meyer, Christoph Weidenbach**

Max-Planck-Institut für Informatik
Im Stadtwald
66123 Saarbrücken, Germany
email: {florent,meyer,weidenb}@mpi-sb.mpg.de

Abstract. We show that unification in certain extensions of shallow equational theories is decidable. Our extensions generalize the known classes of shallow or standard equational theories. In order to prove decidability of unification in the extensions, a class of Horn clause sets called sorted shallow equational theories is introduced. This class is a natural extension of tree automata with equality constraints between brother subterms as well as shallow sort theories. We show that saturation under sorted superposition is effective on sorted shallow equational theories. So called semi-linear equational theories can be effectively transformed into equivalent sorted shallow equational theories and generalize the classes of shallow and standard equational theories.

1 Introduction

Algorithms to solve unification and word problems in an equational theory play a crucial role in many areas of computer science like automated deduction, logic and functional programming, and symbolic constraint solving. Many algorithms are dedicated to particular theories and often semantic conditions are assumed. In addition, a lot of progress has been made towards syntactic characterizations of classes of equational theories or rewrite systems in which these problems are decidable. The class of *shallow* theories, axiomatized by equations in which variables occur at most at depth one, has been shown by Comon, Haberstrau & Jouannaud (1994) to have a decidable unification problem. They exploit a transformation of the system into an equivalent cycle-syntactic presentation (Kirchner 1986). By a termination analyses under basic superposition Nieuwenhuis (1996) generalized the result to so-called standard theories.

Furthermore, tree automata and tree grammars have also been used for unification purposes. Limet & Réty (1997) use Tree Tuple Synchronized Grammars to generate solutions to unification problems by a simulation of narrowing. In (Kaji, Toru & Kasami 1997) it is shown that the closure with respect to some kind of term rewriting system of the (recognizable set) of ground instances of a

* This work was partially supported by the CONSOLE project.
** This work was supported by the German science foundation program Deduktion.

linear term is recognizable. Similar techniques based on the completion of tree automata are presented by Comon (1995) and Jacquemard (1996) for linear shallow TRS and a generalization called linear growing TRS. The decidability of the word problem as well as restricted cases of unifiability in the concerned theories can be derived from these results.

In this paper we show the decidability of unification in so called semi-linear equational theories which strictly extend shallow theories. Informally, a semi-linear system contains equations in which non-linear variables only appear in the same subterms. For example, the equation $f(f(x,x),y) \approx g(f(x,x))$ is semi-linear whereas $f(g(x),h(x),h(g(y))) \approx h(g(x))$ is not. Our techniques are influenced by tree automata, sorted unification and saturation-based methods.

Sorted shallow equational theories naturally generalize tree automata with equality constraints (Bogaert & Tison 1992) as well as shallow sort theories (Weidenbach 1998). Throughout the paper, we consider the following example of Nieuwenhuis (1996). The equational theory is given by the equations $f(g(x),y) \approx h(y)$ and $f(x,x) \approx g(x)$. The definition of standard theories does not include this case. The closure of the theory under basic superposition leads to an infinite set of equations $g(h^n(g(x))) \approx h^{n+1}(g(x))$. The infinite expansion can be avoided by abstracting the linear (semi-linear) term $g(x)$ into a sort declaration $S(g(x))$. The theory is then transformed into a sorted shallow equational theory consisting of the Horn clauses $\| \rightarrow S(g(x))$, $S(x) \| \rightarrow f(x,y) = h(y)$, $\| \rightarrow f(x,x) = g(x)$. Our notation for clauses is of the form $Sort\ Constraint \| Antecedent \rightarrow Succedent$ where the sort constraint atoms are particular, monadic antecedent atoms for which special inference rules are provided by our sorted superposition calculus.

The paper is organized as follows: Section 3 starts with a discussion on tree automata with brother constraints. We prove that they are not sufficient for our purpose. Then sorted shallow equational theories are studied. It is shown that saturation under sorted superposition terminates and that unifiability modulo the saturated theory is decidable. A procedure which transforms a sorted semi-linear equational theory into an equivalent sorted shallow one is given in Section 4. This implies the decidability of unifiability modulo a set of (sorted) semi-linear equations. This result strictly embeds previous ones concerning shallow theories by Comon et al. (1994). We show in Section 5 that with similar techniques, we can treat a generalization of standard theories as proposed by Nieuwenhuis (1996). In the same section, we also consider some other extensions for which our method does not work and discuss some related work on E-unification. For more details consider our technical report (Jacquemard, Meyer & Weidenbach 1998).

2 Preliminaries

We adhere to the usual definitions for variables, terms, substitutions, equations, atoms, (positive and negative) literals, multisets, and clauses, see (Dershowitz &

Jouannaud 1990) for what concerns equational theories. We give just the most important definitions for our purpose.

The algebra of terms over a finite set of function symbols $\mathcal{F}$ and a set $\mathcal{X}$ of variables is denoted $\mathcal{T}(\mathcal{F},\mathcal{X})$ and $\mathcal{T}(\mathcal{F})$ is its subalgebra of ground terms. An *equation* is an unoriented pair of terms of $\mathcal{T}(\mathcal{F},\mathcal{X})$ denoted $s \approx t$. For sake of simplicity, we may apply to equations or other atoms the same following notations as for terms. The function *vars* maps terms, atoms, literals, clauses and sets of such objects to the set of variables occurring in these objects. A *position* p in a term (equation, atom) is a word over the natural numbers. For a term (equation, atom) t we define $t|_p$ of t at position p by $t|_\epsilon = t$ and $t|_{i.p} = t_i|_p$ where $t = f(t_1,\ldots,t_n)$ and $1 \leq i \leq n$. We write $t[s]_p$ to denote that $t|_p = s$ and $t[p/s']$ is the term obtained from t by replacing its subterm at position p by s'.

A term is called *complex* if it is neither a constant nor a variable. A term t is called *shallow* if t is a variable or is of the form $f(x_1,\ldots,x_n)$ where the x_i are not necessarily different. An equation $s \approx t$ is called *shallow* if both s and t are shallow. Note that shallow variables in $s \approx t$ can be arbitrarily shared by s and t. A term t is called *linear* if every variable occurs at most once in t. A term t is called *semi-linear* if it is a variable or of the form $f(t_1,\ldots,t_n)$ such that every t_i is semi-linear and whenever $vars(t_i) \cap vars(t_j) \neq \emptyset$ we have $t_i = t_j$ for all i, j. An equation $s \approx t$ is semi-linear if (i) s and t are variables or (ii) $s = f(s_1,\ldots,s_n)$ and if t is a variable and $t \in vars(s_i)$ then $s_i = t$ for all i or if $t = g(t_1,\ldots,t_m)$ and $vars(s_i) \cap vars(t_j) \neq \emptyset$ then $s_i = t_j$ for all i, j. For instance, the term $f(g(x),g(x),h(y,y))$ and the equations $h(g(x),g(x),y) \approx f(y,g(x),y)$ and $f(g(x),g(x),y) \approx y$, are semi-linear, but $f(g(x),g(x),h(x,y))$, $h(g(x),x)$ and $h(g(x),g(x)) \approx x$ are not.

Atoms formed from unary predicates are called *monadic*. For the purpose of this paper, a Horn clause is written in the form $\Theta \parallel \Gamma \rightarrow \Delta$ where the *sort constraint* Θ consists of monadic atoms representing the sort restrictions. Γ and Δ denote the antecedent and succedent atoms of the clause, respectively. In the initial clause set we assume the arguments of all sort constraint atoms to be variables. Furthermore, for the theories we consider here, it is always the case that either the antecedent or the succedent of a clause is empty.

We call a sort constraint Θ *solved* in a Horn clause $\Theta \parallel \Gamma \rightarrow \Delta$, if $vars(\Theta) \subseteq vars(\Gamma \cup \Delta)$ and all terms occurring in Θ are variables. A Horn clause $T_1(x_1),\ldots,T_n(x_n) \parallel \rightarrow S(t)$ is called a *declaration* if $T_1(x_1),\ldots,T_n(x_n)$ is solved. In case t is a variable, a declaration is called a *subsort* declaration. A declaration $T_1(x_1),\ldots,T_n(x_n) \parallel \rightarrow S(t)$ is *shallow* (*linear*, *semi-linear*) if t is shallow (linear, semi-linear). A *sort theory* is a finite set of declarations. It is called *shallow* (*linear*, *semi-linear*) if all declarations are shallow (linear, semi-linear). A *sorted equation* (*sorted disequation*) is a clause $\Theta \parallel \rightarrow l \approx r$ (a clause $\Theta \parallel l \approx r \rightarrow$) where Θ is solved. A *sorted equational theory* is a finite set of sorted equations and declarations. It is called *shallow* (*semi-linear*) if all equations and all declarations are shallow (semi-linear).

A *substitution* is a mapping from $\mathcal{X}$ to $\mathcal{T}(\mathcal{F},\mathcal{X})$. As usual, we do not distinguish between a substitution and its homomorphic extension in the free algebra

$\mathcal{T}(\mathcal{F},\mathcal{X})$. Given an *equational theory* E, i.e., a finite set of equations, we write $s \underset{E}{\leftrightarrow} t$ iff there exists an equation $l \approx r \in E$ and a substitution σ such that $s|_p = l\sigma$ and $t = s[p/r\sigma]$. The reflexive symmetric transitive closure of the binary relation $\underset{E}{\leftrightarrow}$ is denoted $\underset{E}{\overset{*}{\leftrightarrow}}$. Two terms s and t are called *unifiable* modulo an equational theory E iff there exists a substitution σ such that $s\sigma \underset{E}{\overset{*}{\leftrightarrow}} t\sigma$. Note that this is equivalent to stating that the clause set consisting of the equational theory E and the clause $\| s \approx t \rightarrow$ is unsatisfiable.

For a set of Horn clauses $\mathcal{A}$ and a clause C, $\mathcal{A} \models C$ denotes the usual semantic entailment relation where all variables of $\mathcal{A}$ and C are assumed to be universally quantified.

3 Decidability of Sorted Shallow Equational Theories

In this section we show how saturation-based methods for sorted shallow equational theories succeed in the decision of unifiability in (non-linear) shallow theories whereas tree automata (with constraints) techniques fail.

3.1 Tree automata and linear shallow theories

We adopt here a definition of tree automata by means of Horn clauses. This definition, though non-standard, is equivalent to the usual ones, e.g., (Bogaert & Tison 1992). A systematic correspondence between various types of Horn clause sets and known classes of tree automata with constraints has been studied by one of the authors (Weidenbach 1998).

Definition 1. A tree automaton $\mathcal{A}$ is a finite set of linear shallow declarations of the form $S_1(x_1), \ldots, S_n(x_n) \| \rightarrow S\big(f(x_1, \ldots, x_n)\big)$.

Following tree automata terminology, the unary predicates are called *states* and the Horn clauses of $\mathcal{A}$ are *transition rules* or just transitions.

A term $t \in \mathcal{T}(\mathcal{F})$ is *recognized* by $\mathcal{A}$ in some state S if $\mathcal{A} \models S(t)$. If we fix in $\mathcal{A}$ a subset $\mathcal{S}$ of *final states* (final predicates), then $t \in \mathcal{T}(\mathcal{F})$ is recognized by $\mathcal{A}$ (with respect to $\mathcal{S}$) if t is recognized by $\mathcal{A}$ in some final state. A set $L \subseteq \mathcal{T}(\mathcal{F})$ is a recognizable language if L is the set of ground terms which are recognized by a tree automaton $\mathcal{A}$ (with respect to some set of final states).

The class of recognizable languages is closed under Boolean operations. Every recognizable language is recognized by one *deterministic* tree automaton $\mathcal{A}$ such that a ground term cannot be recognized by $\mathcal{A}$ in more than one state. Every recognizable language is recognized by one *completely specified* tree automaton $\mathcal{A}$, such that every ground term is recognized by $\mathcal{A}$ at least in one state. It is decidable in polynomial time whether a given term $t \in \mathcal{T}(\mathcal{F})$ is recognized by a tree automaton $\mathcal{A}$. It is decidable in linear time whether the language recognized by some tree automaton $\mathcal{A}$ is empty or not.

Tree automata and grammars have been used by Kaji et al. (1997) and Limet & Réty (1997) to solve word and unifiability problems. In the first paper as well as in the papers by Comon (1995) and Jacquemard (1996) the recognizability

of the closure of some recognizable set L with respect to term rewriting systems of restricted classes is investigated. In the following we denote the closure of a set of terms $L \subseteq \mathcal{T}(\mathcal{F})$ with respect to an equational system E: $(\overset{*}{\underset{E}{\leftrightarrow}})(L) := \{s \in \mathcal{T}(\mathcal{F}) \mid \exists t \in L\ t \overset{*}{\underset{E}{\leftrightarrow}} s\}$. For a given system E we can reduce the word problem $s \overset{*}{\underset{E}{\leftrightarrow}} t$ to the membership problem for $s \in (\overset{*}{\underset{E}{\leftrightarrow}})(\{t\})$ if the closure set and L is recognizable. For a goal $s = t$ where s and t are both linear and $vars(s) \cap vars(t) = \emptyset$, unifiability modulo E is equivalent to $\{s\sigma \mid \sigma \text{ ground}\} \cap (\overset{*}{\underset{\mathcal{R}}{\leftrightarrow}})(\{t\sigma \mid \sigma \text{ ground}\}) = \emptyset$. Since the set of ground instances of s and t are both recognizable unifiability in this case can be reduced to an emptiness decision problem for tree automata.

Theorem 2. *(Comon 1995). Let E be a linear shallow equational system and L be a recognizable language. Then $(\overset{*}{\underset{E}{\leftrightarrow}})(L)$ is a recognizable language.*

The principle of the construction for linear shallow equational systems is the following. We start with a tree automaton $\mathcal{A}_0$ which recognizes L and contains one state S_{l_i} for each direct ground subterm l_i in equations $f(l_1, \ldots, l_n) \approx r$ in E such that l_i (and only l_i) is recognized by $\mathcal{A}_0$ in S_{l_i}. In some sense these subterms are abstracted by $\mathcal{A}_0$. Then $\mathcal{A}_0$ is completed with respect to inference rules like the one below. Note that the construction of $\mathcal{A}_0$ is similar to our transformation process in Section 4.

$$\text{Inf}\ \frac{\begin{array}{r} \| \rightarrow f(l_1, \ldots, l_n) \approx g(r_1, \ldots, r_m) \quad \in E \\ S_1(x_1), \ldots, S_n(x_n) \| \rightarrow S\big(f(x_1, \ldots, x_n)\big) \end{array}}{T_1(x_1), \ldots, T_m(x_m) \| \rightarrow S\big(g(x_1, \ldots, x_m)\big)}$$

This inference rule is applied providing that for each $i \leq m$ such that r_i is a ground term, $T_i = S_{r_i}$, and for each $i \leq m$, $j \leq n$ such that l_j and r_i are the same variables, then $T_i = S_j$. If we apply paramodulation to the premises of the above inference rule we obtain the clause $S_1(l_1), \ldots, S_n(l_n) \| \rightarrow S\big(g(r_1, \ldots, r_n)\big)$. With the two above conditions (abstraction of r_i by T_i and equalities between states according to variables), and since E is linear shallow, this clause is equivalent to $T_1(y_1), \ldots, T_m(y_m) \| \rightarrow S\big(g(y_1, \ldots, y_m)\big)$. This relates the automata theoretic approach and its generalization presented in Section 3.3. Unfortunately, the above recognizability result of Theorem 2 cannot be extended to non-linear systems, as the following example shows. Assume f is a binary function symbol, s is unary and a is a constant, and let $L = \{a\}$, $E = \{f(x, x) \approx a\}$. Then it is well known that languages of the form $\{f(t, t)\}$ are not recognizable by tree automata.

3.2 Brothers automata and the non-linearities

Bogaert & Tison (1992) introduce tree automata with constraints which define a strict superclass of recognizable languages to deal with non-linear rewrite systems.

Definition 3. A tree automaton with equality constraints between brother subterms is a finite set of shallow declarations of the form $S_1(x_1), \ldots, S_n(x_n) \| \rightarrow S(f(x_1, \ldots, x_n))$ where the x_i are not necessarily different.

We call $Rec_=$ this class of recognizers as well as the class of recognized languages; the notion of recognized terms and languages is the same for tree automata with equality constraints between brother subterms as for (standard) tree automata.

The class $Rec_{\neq}$ (Bogaert & Tison 1992) is strictly larger than $Rec_=$ because (syntactic) disequations between variables $x_i \neq x_j$ are also allowed in the antecedent of clauses. The nice closure properties of tree automata still apply here, namely closure under Boolean operations, under determinism and complete specification. The emptiness problem is also decidable for $Rec_{\neq}$ though EXPTIME-hard. However, disequalities are not necessary for our purpose (see the conclusion for a discussion about this extension), but we can show that neither $Rec_=$ nor $Rec_{\neq}$ suffice to generalize Theorem 2 to the case of non-linear shallow systems.

Lemma 4. *There exists some recognizable set L and (non-linear) shallow equational system E such that the set $(\overset{*}{\underset{E}{\leftrightarrow}})(L)$ is not in $Rec_{\neq}$.*

Proof. Let f, g be two binary function symbols, a be some constant and consider the system $E := \{f(x,x) \to g(x,x)\}$ and language $L := \{g(s_1,s_2) \mid s_1, s_2 \in \mathcal{T}(\mathcal{F})\}$. Assume that $L' := (\overset{*}{\underset{E}{\leftrightarrow}})(L)$ is recognized by some $\mathcal{A} \in Rec_{\neq}$ with respect to the distinguished set of final states $\mathcal{S}$. We may assume without loss of generality that $\mathcal{A}$ is deterministic and completely specified. Let N be the number of states of $\mathcal{A}$. Let us define a sequence of well-balanced ground terms of $\mathcal{T}(\mathcal{F})$ by $t_1 := f(a,a)$ and for all $i \geq 1, t_{i+1} := f(t_i,t_i)$. It is easy to check that for all $i \geq 1$, the cardinal of the equivalence class of t_i modulo $\overset{*}{\underset{E}{\leftrightarrow}}$ is 2^{2^i-1}. Thus there exists an integer $i_0 = \lceil \log(\log(|Q|+2)) \rceil$ such that for all $i \geq i_0$, we have two distinct ground terms s_i, s_i' both equivalent to t_i modulo $\overset{*}{\underset{E}{\leftrightarrow}}$ and both recognized by $\mathcal{A}$ in the same state called S_i. Moreover, by construction, we have that $f(s_i, s_i') \in L'$. Thus this term is recognized by $\mathcal{A}$ in some final state $S_i^{\mathrm{f}} \in \mathcal{S}$. By determinism of $\mathcal{A}$, there exists a clause $C_i = S_i(x_1), S_i(x_2) \,\|\, \Gamma_i \to S_i^{\mathrm{f}}(f(x_1,x_2)) \in \mathcal{A}$, ($\Gamma_i$ is a set of syntactic disequations between variables) such that $\mathcal{A} \models C_i\sigma$ with $\sigma = \{x_1 \mapsto s_j, x_2 \mapsto s_{j'}\}$. Note that for all $i \geq i_0$, the variables x_1 and x_2 are distinct because $s_i \neq s_i'$. On the other hand, there exist two distinct integers $j, j' \geq i_0$ such that $S_j = S_{j'}$. Thus, $\mathcal{A} \models C_j\sigma$ where $\sigma = \{x_1 \mapsto s_j, x_2 \mapsto s_{j'}\}$, because $x_1 \neq x_2$ in C_j and thus $f(s_j, s_{j'})$ is recognized by $\mathcal{A}$ in the final state S_j^{f}. This is a contradiction because this term is not in L'.

We can conclude from Lemma 4 that the syntactic equality constraints of the automata in $Rec_=$ are too rough for our purpose. The sorted shallow equational theories studied in the following section are a strict generalization of $Rec_=$. An important achievement of this approach is that semantic equality tests are possible.

3.3 Saturation

The following inference rules form a sound and refutationally complete calculus for Horn clause sets consisting of declarations and sorted (dis)equations.

They are mainly an adaption of basic superposition with selection (Bachmair, Ganzinger, Lynch & Snyder 1995, Nieuwenhuis & Rubio 1995) to the particular form of the Horn clauses considered here, where the sort constraints are subject to the basic restriction and are solved by a particular selection strategy. This strategy is expressed by the rule Sort Constraint Resolution, see below. As usual, we assume a reduction ordering $\succ$ that is total on ground terms. We call the calculus consisting of the inference rules Sort Constraint Resolution, Superposition Right, Superposition Left and Equality Resolution plus the usual reduction rules subsumption and condensing the *sorted* superposition calculus. Note that the basic restriction does not interfere with subsumption or condensing, because sort constraint atoms are solely subsumed (condensed) by other sort constraint atoms and in considered clause sets no non-variable terms occur in the sort constraint.

Definition 5 (Sort Constraint Resolution). The inference

$$\text{Inf}\ \frac{\begin{array}{rl} T_1(t), \ldots, T_n(t), \Psi \,\|\, \Gamma & \rightarrow \Delta \\ \Theta_1 \,\|\, & \rightarrow T_1(t_1) \\ \vdots & \\ \Theta_n \,\|\, & \rightarrow T_n(t_n) \end{array}}{\bigcup_i \Theta_i\sigma, \Psi\sigma \,\|\, \Gamma\sigma \rightarrow \Delta\sigma}$$

where t is either a non-variable term or t is a variable $t = x$ with $x \notin vars(\Gamma \cup \Delta)$ and no non-variable term occurs in Ψ; no further atom $S(t)$ occurs in Ψ, σ is the simultaneous mgu of t, t_1, ..., t_n and all Θ_i are solved is called a *Sort Constraint Resolution* inference.

Definition 6 (Superposition Right). The inference

$$\text{Inf}\ \frac{\begin{array}{rl} \Psi \,\|\, & \rightarrow s \approx t \\ \Theta \,\|\, & \rightarrow A[s']_{1.p} \end{array}}{\Psi\sigma, \Theta\sigma \,\|\, \rightarrow A[1.p/t]\sigma}$$

where σ is the mgu of s and s', $t\sigma \not\succ s\sigma$, s' is not a variable, if A is an equation $l \approx r$ with $l|_p = s'$ then $r\sigma \not\succ l\sigma$ and the sort constraints Ψ, Θ are solved is called a *Superposition Right* inference.

Definition 7 (Superposition Left). The inference

$$\text{Inf}\ \frac{\begin{array}{rl} \Psi \,\|\, & \rightarrow s \approx t \\ \Theta \,\|\, l[s']_p \approx r & \rightarrow \end{array}}{\Psi\sigma, \Theta\sigma \,\|\, l[p/t]\sigma \approx r\sigma \rightarrow}$$

where σ is the mgu of s and s', $t\sigma \not\succ s\sigma$, s' is not a variable, $r\sigma \not\succ l\sigma$ and the sort constraints Ψ, Θ are solved is called a *Superposition Left* inference.

Definition 8 (Equality Resolution). The inference

$$\text{Inf}\ \frac{\Theta \,\|\, s \approx t \rightarrow}{\Theta\sigma \,\|\, \quad \rightarrow}$$

where σ is the mgu of s, t and Θ is solved is called a *Equality Resolution* inference.

Lemma 9. *Sorted shallow equational theories can be finitely saturated by sorted superposition.*

Proof. We shall show that the saturation process results in clauses of the form

$$T_1(t), \ldots, T_n(t), S_1(x_1), \ldots, S_m(x_m) \,\|\, \rightarrow A$$

where n, m are possibly zero, A is either a monadic atom $T(s)$ or an equation $l \approx r$ and t, s, l and r are always shallow terms. If the saturation process produces only clauses of this form, then it will terminate, because the depth of all these clauses as well as the length of variable chains between their literals are bound. Hence, there are only finitely many different clauses of this form modulo subsumption and condensing.

It remains to prove that all clauses generated by the saturation process have the above form. Obviously, shallow declaration clauses and sorted shallow equations are of the above form, where t as well as the x_i are variables occurring in A. For symmetry reasons it is sufficient to consider three cases of possible inferences: (i) The term t is a non-variable shallow term and we perform a sort constraint resolution inference. (ii) The term t is a variable that does not occur in A and we perform a sort constraint resolution inference. (iii) The sort constraint $T_1(t), \ldots, T_n(t), S_1(x_1), \ldots, S_m(x_m)$ is solved and we perform a superposition right inference. We separately consider these cases:
(i) The other clauses involved in the inference are all of the form $Q_1(y_1), \ldots, Q_{k_i}(y_{k_i}) \,\|\, \rightarrow T_i(t_i)$ where the y_j occur in t_i and t_i is a shallow term. The unifier σ only maps a variable to a non-variable shallow term if the variable is some t_i. Hence, the result of the inference is a clause of the desired form.
(ii) Again all other clauses involved in the inference are of the form $Q_1(y_1), \ldots, Q_{k_i}(y_{k_i}) \,\|\, \rightarrow T_i(t_i)$ where the y_j occur in t_i and t_i is a shallow term. The unifier σ possibly maps the variable t to a non-variable shallow term, but since t does not occur in A the result of the inference is again a clause of the desired form.
(iii) Since we do not superpose into variables and for any equation of the form $f(x_1, \ldots, x_n) \approx y$ either $y = x_i$ for some i and hence $f(x_1, \ldots, x_n) \succ x_i$ or y does not occur in $f(x_1, \ldots, x_n)$, a case analysis over the different combinations of the form of A and the involved sorted equation shows that the result is always of the desired form. Note that in the case of a Superposition Right inference, the involved clauses have a solved sort constraint.

For example, we apply the saturation process to the sorted shallow equational theory presented in Section 1:

(1) $S(x)$	$\,\|\, \rightarrow f(x,y) \approx h(y)$
(2)	$\,\|\, \rightarrow f(x,x) \approx g(x)$
(3)	$\,\|\, \rightarrow S(g(x))$

where we assume $f(x,y) \succ g(x) \succ h(x)$. Then the saturation process generates the additional clauses (4) and (5) by Superposition Right inferences.

$$(4)\ S(x) \quad \| \rightarrow g(x) \approx h(x)$$
$$(5)\ S(x) \quad \| \rightarrow S(h(x))$$

The clauses (1)–(5) are saturated by sorted superposition.

Lemma 10. *Unifiability with respect to finitely saturated sorted shallow equational theories is decidable.*

Proof. Two arbitrary terms t, s are unifiable iff we can derive the empty clause from the saturated theory and the goal clause $\| t \approx s \rightarrow$. Since the sorted shallow equational theory is saturated, no inferences inside the theory need to be considered. Furthermore, the goal is purely negative, so we can delete all clauses with an unsolved sort constraint from the saturated theory. We show that the sorted superposition calculus terminates on the goal clause. All generated clauses are of the form:

$$S_1(t_1), \ldots, S_m(t_m) \| t' \approx s' \rightarrow$$

where t', s' are terms resulting from inference rule applications to clauses inferred from the goal clause, m is possibly zero and $t' \approx s'$ does possibly not exist (after the application of an Equality Resolution inference). All inference rule applications to clauses of the above form, except Sort Constraint Resolution to a variable, are monotone in the well-founded ordering composed of the lexicographic combination of the number of non-linear variables occurring at different depth in some $t_1, \ldots, t_m, t' \approx s'$ and the maximal term depth of $t_1, \ldots, t_m, t' \approx s'$. The rule Sort Constraint Resolution is only applicable to a variable if the variable does not occur in $t' \approx s'$ and all other t_i are variables. Then an application generates at most one new shallow term in the sort constraint and following the argumentation in the proof of Lemma 9 the process of solving the generated sort constraints will eventually terminate. In summary, the term depth in all generated clauses is bound and solving sort constraints with variables that do not occur in the antecedent equation terminates. It remains to show that the length of variable chains in the generated clauses is bound. Obviously, such crucial chains can only occur in the sort constraint between sort constraint atoms that have a complex term as its argument. But this cannot happen, since Sort Constraint Resolution applied to a non-variable term is strictly monotone in the above ordering if the involved declarations have a succedent atom with a non-variable argument. With respect to subsort declarations, the saturation using Sort Constraint Resolution terminates anyway.

We evaluate two example queries with respect to the above saturated sorted shallow equational theory. First, we want to unify $f(x,y)$ and $h(y)$ starting with the goal clause

$$\| f(x,y) \approx h(y) \quad \rightarrow$$

We apply Superposition Left with (1) giving $S(x) \| h(y) \approx h(y) \rightarrow$. Next we apply Sort Constraint Resolution with (3) yielding $\| h(y) \approx h(y) \rightarrow$ and finally an application of Equality Resolution yields the empty clause. Therefore, $f(x,y)$ and $h(y)$ are unifiable in the considered shallow equational theory.

Second, consider the unification problem of $f(a,x)$ and $h(x)$ where a is some constant. The problem has no solution justified by the saturated clause set consisting of the clauses (1)–(5) and the clauses below:

$$S(a) \quad \begin{array}{ll} \| f(a,x) \approx h(x) & \rightarrow \\ \| & \rightarrow \\ \| g(a) \approx h(a) & \rightarrow \end{array}$$

4 Semi-Linear Sorted Equational Theories

In this section we prove that unification in semi-linear equational theories is decidable, too. We do so by transforming a semi-linear equational theory into a sorted shallow equational theory, preserving satisfiability. Then we apply Lemma 9 and Lemma 10 to obtain the decidability result. The following rule transforms sorted semi-linear equational theories into sorted shallow equational theories.

Definition 11. The transformation

$$\text{Red}\ \frac{\Psi, S_1(x_1), \ldots, S_m(x_m) \| \rightarrow A[t]_{p_1}}{\begin{array}{l} S_1(x_1), \ldots, S_m(x_m) \| \rightarrow T(t) \\ T(y), \Psi \| \rightarrow A[p_1, \ldots, p_n/y] \end{array}}$$

provided t is a non-variable subterm, $x_i \in vars(t)$ for all i, $vars(\Psi) \cap vars(t) = \emptyset$, $|p_i| = 2$ for all i, the positions $p_1, \ldots, p_n$ refer to all positions q of t in A with $|q| = 2$, T is a new monadic predicate and y is new to the replaced clause is called *flattening*.

Lemma 12. *Exhaustive application of flattening to a (sorted) semi-linear equational theory terminates, results in a sorted shallow equational theory and preserves satisfiability.*

Proof. Termination follows from the fact that the transformation replaces a clause by two clauses with fewer function symbols. No transformation is applicable to a clause that is a shallow declaration or a sorted shallow equation, since all terms at depth two of such atoms are always variables (if they exist). On the other hand, if the direct subterm of an atom is not shallow, it has a subterm at depth two which is not a variable and therefore the transformation applies. Hence, the transformation terminates in a sorted shallow equational theory.

By an induction argument it is sufficient to show that a single step of the transformation preserves satisfiability and results in a sorted semi-linear equational theory. The crucial property is that $vars(t) \cap vars(A[p_1, \ldots, p_n/y]) = \emptyset$. We show this by contradiction. Assume that after an application of the transformation there is a variable z occurring in t and $A[p_1, \ldots, p_n/y]$. By construction this can only be the case if z has an occurrence in A that is not inside an occurrence of t in A. So z occurs in some term $s \neq t$ with $A|_q = s$, $|q| = 2$, contradicting that the clause is semi-linear to which the transformation is applied. For the same reason, the result of an application of the transformation is again a sorted semi-linear equational theory and $x_j \notin vars(A[p_1, \ldots, p_n/y])$ for all j.

Theorem 13. *Unifiability in semi-linear equational theories is decidable.*

Proof. By Lemma 12 we can effectively translate semi-linear equational theories into sorted shallow equational theories preserving satisfiability. By Lemma 9 these theories can be effectively saturated by sorted superposition and by Lemma 10 unifiability is decidable with respect to saturated sorted shallow equational theories.

Application of the transformation to the example presented in the introduction yields the sorted shallow equational theory considered in the previous section.

4.1 Applications

Any equational theory E can be transformed into a semi-linear equational theory E' by replacing non-linear variable occurrences with fresh variables. Then E' is an upper approximation for E in the sense that $\overleftrightarrow{\overset{*}{E}} \subseteq \overleftrightarrow{\overset{*}{E'}}$, i.e., non-unifiability in E' implies non-unifiability in E. Furthermore, by Theorem 13, non-unifiability in E' is decidable. Ganzinger, Meyer & Weidenbach (1997) showed that in this case non-unifiability in E' can be used to effectively direct the search of a theorem prover in finding proofs with respect to E. One of our future goals is to improve the performance of SPASS (Weidenbach 1997) using this technology. Note that flattening applied to an arbitrary equational theory where we keep some $S_i(x_i)$ in the transformed clause if $x_i \in vars(A[p_1, \ldots, p_n/y])$ is already a transformation that generates an appropriate approximation.

5 Extensions, Limitations and Related Work

5.1 Extensions

A possible extension is to apply our method to compute the (eventual) solution of a unification problem in a semi-linear theory. Sort Constraint Resolution simulates sorted unification. Unification in shallow sort theories is known to be NP-complete and of unification type finitary. This implies that unification in sorted shallow equational theories is NP-hard and also of unification type finitary, if we consider well-sorted unifiers. The results of Theorem 13 in Section 4 obviously extend to sorted semi-linear equational theories.

The standard equations in (Nieuwenhuis 1996) include one form which is not embedded by the semi-linear case: the form $f(\ldots, g(x), \ldots) \approx x$ where g has to be a unary function symbol, assuming additional restrictions on the positions of linear terms and non-linear shallow variables in other equations. Obviously, the subterm $g(x)$ cannot be transformed into a sort declaration. However, we can show that unification in those theories can still be decided by sorted superposition using basic strategies on so-called *semi-standard* equations. We call an equation $f(t_1, \ldots, t_n) \approx x$ *semi-standard* if $f(t_1, \ldots, t_n)$ is semi-linear and moreover, there is one unary symbol g such that for all t_i with $x \in vars(t_i)$ we have

that $t_i = g(x)$. An equational theory E is called *semi-standard* if E only contains semi-linear equations or semi-standard equations of the form $f(t_1, \ldots, t_n) \approx x$ where only one t_i can be of the form $g(x)$.

Theorem 14. *Unifiability in semi-standard equational theories is decidable.*

The procedure in the proof of Lemma 12 which transforms a semi-linear theory into a sorted shallow theory can be extended to work for a semi-standard theory. The resulting system may contain clauses of the form $\Theta \parallel \rightarrow f(t_1, \ldots, t_n) \approx x$ where the equation is a so-called *semi-shallow* equation. The generalization of shallow equations to semi-shallow equations is similar to the extention of semi-linear equations to semi-standard equations. In the transformation procedure occurrences of subterms of the form $g(x)$ are not abstracted into sorts. Moreover, equations $s \approx t$ where $vars(s) \cap vars(t) = \emptyset$ are transformed into sorted clauses of the form $\Theta \parallel \rightarrow x \approx y$ to ensure that non-collapsing equations share at least one variable between both sides. The saturation of semi-shallow theories still terminates by imposing basic restrictions on subterms of the form $g(x)$ which have been introduced by unifiers into the equational part of a clause and which cannot be moved to the sort constraint.

E-unification remains decidable in the according saturated set shown by an analogous of Lemma 10. The problem is that the maximal term depth can be increased by one while the number of variables does not change. However, the according termination ordering can be generalized in a way that basic and non-basic regions of a clause are distinguished. Further detailes can be found in the technical report (Jacquemard et al. 1998).

5.2 Limitations

We present a generalization of semi-linear equational systems which cannot be treated with the methods of Sections 3.3 and 4.

The combination of associativity for one function symbol and a linear (!) shallow sort theory already yields an undecidable unification problem. This can be seen by a reduction of the emptiness of the intersection of context free languages to this problem.

Pseudo-linear theories generalize sorted semi-linear equational theories in a way that multiple occurrences of a variable in an equation are allowed, provided that they occur at the same depth. For instance $f(h(x), g(x)) \approx g(g(x))$ is pseudo-linear (though not semi-linear) and $f(h(x), g(x)) \approx g(x)$ is not. However, emptiness of some sort with respect to the combination of a linear (!) shallow sort theory and a pseudo-linear equational theory is already undecidable as shown in the following proposition.

Proposition 15 (Jacquemard et al. 1998). *The blank accepting problem for a non-deterministic Turing machine can be reduced to the emptiness of the intersection* $\left(\overleftarrow{E^*}\right)(L_1) \cap L_2$ *where* L_1 *and* L_2 *are two recognizable word languages and* E *is an equational word system with equations of the form* $aa' \approx bb'$.

Note that this kind of theory is an even simple case of a pseudo-linear equational system. The word problem in pseudo-linear word systems is decidable since pseudo-linear word equations are length preserving.

5.3 Related Work

Oyamaguchi (1990) shows that the word problem for right-ground TRS is undecidable whereas the word problem in left-linear and right-ground TRS is decidable in polynomial time. In the undecidability proof for right-ground systems Oyamaguchi used rewrite rules with non-linear variable occurrences at different depth.

Fassbender & Maneth (1996) investigate decidability of E-unification in theories induced by TRS called top-down tree transducers. Syntactic restrictions based on separated function and constructor alphabets are assumed. E-unification in top-down tree transducers with only one function symbol in the alphabet is shown to be decidable. Due to the constructor-based restrictions the results are difficult to compare to semi-linear theories. Otto, Narendran & Dougherty (1995) show that E-unification is decidable in equational theories axiomatized by monadic, confluent string-rewriting systems.

Kaji et al. (1997) show the recognizability of the right-closure of a certain class of right-linear, confluent TRS applied to a linear term. The variables occurring both in the left and right hand side of a rule $l \to r$ are assumed to be linear in l and, moreover, l and the subterms of r are related under additional restrictions which can be effectively computed. The techniques presented in Subsection 3.1 provide a decision method for some restricted unifiability problems modulo the above systems. Actually, the problem addressed in (Kaji et al. 1997) is more general because they deal with "constrained substitutions" which range in some recursively defined (recognizable) set of terms.

Comon et al. (1994) investigate the properties of non-linear shallow theories which are an instance of semi-linear equational theories. Shallow presentations can be transformed into equivalent cycle-syntactic presentations for which decidability of unification has been shown. The first-order theory of the quotient algebra $T(F)/_{=_E}$ is also shown to be decidable where F is finite and E is shallow. However, the proof techniques are entirely different to our approach.

Nieuwenhuis (1996) generalizes the result of Comon et al. (1994) to so-called standard theories. Standard theories extend non-linear shallow theories in a way that non-ground terms containing linear (non-significant) variables are allowed in certain restricted positions in both sides of the equations. An equation $f(s_1, \ldots, s_n) = g(t_1, \ldots, t_m)$ may contain linear terms s_i, respectively t_i, where all other equations with top symbol f, respectively g, must have linear terms in position i. Non-linear variable occurrences are limited to shallow positions [1]. The saturation-based methods are closely related to our work. The decidability results are also obtained by termination analyses of saturation under basic superposition.

[1] Another extension included in standard theories is discussed in Section 5.

Limet & Réty (1997) show the decidability of E-unification in theories represented by a particular class of confluent, constructor-based TRS. The set of possibly infinite solutions is represented by Tree Tuple Synchronized Grammars. A TRS is transformed into such a grammar which then simulates narrowing. The additional restrictions on the TRS are purely syntactic. However, semi-linear systems are difficult to compare to the constructor-based systems in this approach.

6 Conclusions and Future Work

We have shown that unifiability modulo a sorted shallow equational theory is decidable by means of saturation methods under sorted superposition. With the help of a transformation procedure this result extends to (sorted) semi-linear equational theories. Our result strictly embeds previous work concerning shallow theories by Comon et al. (1994). It can be obviously extended into sorted equational theories and also into a generalization of Nieuwenhuis (1996). However, we currently do not have any complexity results concerning the decision procedure or the number of generated mgus. The presented theory is already included in the first-order theorem prover SPASS (Weidenbach 1997) that can therefore be used for experiments with respect to the presented results.

Let us conclude with another possible improvement of this work. Sorted shallow equational theories generalize $Rec_{=}$ tree automata. To subsume the whole class $Rec_{\neq}$ (Bogaert & Tison 1992), it is necessary to add syntactic disequations to clauses while preserving decidability results concerning membership and emptiness problems. This may have interesting applications in call-by-need normalization strategies for TRS. Durand & Middeldorp (1997) use tree automata techniques both to apply a call-by-need strategy based on the detection of needed redexes and to characterize the class of rewrite systems for which it is effective. The key idea is, given a rewrite system $\mathcal{R}$, to recognize the closure $(\frac{*}{\mathcal{S}}\!\!\rightarrow)(\mathrm{NF}_{\mathcal{S}})$ by $\mathcal{S}$ of the set of ground $\mathcal{S}$-normal-forms, where $\mathcal{S}$ is a certain approximation of $\mathcal{R}$. If we approximate $\mathcal{R}$ into a non-linear shallow system $\mathcal{S}$, the above set could be a recognized sorted shallow equational theory with syntactic disequalities, generalizing $Rec_{\neq}$ automata. Thus, with the appropriate extension of the theory of needed-redexes, more general call-by-need normalization strategies for some classes of non-linear rewrite systems could be obtained.

References

Bachmair, L., Ganzinger, H., Lynch, C. & Snyder, W. (1995), 'Basic paramodulation', *Information and Computation* **121**(2), 172–192.

Bogaert, B. & Tison, S. (1992), Equality and disequality constraints on direct subterms in tree automata, *in* A. Finkel & M. Jantzen, eds, 'Proceedings of 9th Annual Symposium on Theoretical Aspects of Computer Science, STACS92', Vol. 577 of *LNCS*, Springer, pp. 161–171.

Comon, H. (1995), Sequentiality, second order monadic logic and tree automata, *in* 'Proceedings 10th IEEE Symposium on Logic in Computer Science, LICS'95', IEEE Computer Society Press, pp. 508–517.

Comon, H., Haberstrau, M. & Jouannaud, J.-P. (1994), 'Syntacticness, cycle-syntacticness and shallow theories', *Information and Computation* **111**(1), 154 –191.

Dershowitz, N. & Jouannaud, J.-P. (1990), Rewrite systems, *in* J. van Leeuwen, ed., 'Handbook of Theoretical Computer Science', Vol. B, Elsevier Science Publishers, chapter 6, pp. 243–320.

Durand, I. & Middeldorp, A. (1997), Decidable call by need computations in term rewriting (extended abstract), *in* W. McCune, ed., 'Automated Deduction – CADE-14, 14th International Conference on Automated Deduction', LNCS 1249, Springer-Verlag, Townsville, North Queensland, Australia, pp. 4–18.

Fassbender, H. & Maneth, S. (1996), A strict border for the decidability of e-unification for recursive functions, *in* M. Hanus & M. Rodríguez-Artalejo, eds, 'Algebraic and Logic Programming (ALP-5) : 5th international conference, Aachen, Germany, September 25-27, 1996', Vol. 1139 of *Lecture notes in computer science*, Springer, Berlin.

Ganzinger, H., Meyer, C. & Weidenbach, C. (1997), Soft typing for ordered resolution, *in* 'Proceedings of the 14th International Conference on Automated Deduction, CADE-14', Vol. 1249 of *LNAI*, Springer, Townsville, Australia, pp. 321–335.

Jacquemard, F. (1996), Decidable approximations of term rewriting systems, *in* H. Ganzinger, ed., 'Rewriting Techniques and Applications, 7th International Conference, RTA-96', Vol. 1103 of *LNCS*, Springer, pp. 362–376.

Jacquemard, F., Meyer, C. & Weidenbach, C. (1998), Unification in extensions of shallow equational theories, MPI-Report MPI-I-98-2-002, Max-Planck-Institut für Informatik, Saarbrücken, Germany.

Kaji, Y., Toru, F. & Kasami, T. (1997), 'Solving a unification problem under constrained substitutions using tree automata', *Journal of Symbolic Computation* **23**, 79–117.

Kirchner, C. (1986), Computing unification algorithms, *in* 'Proceedings of the First Symposium on Logic in Computer Science', Cambridge, Massachusetts, pp. 206–216.

Limet, S. & Réty, P. (1997), E-unification by means of tree tuple synchronized grammars, *in* M. Bidoit & M. Dauchet, eds, 'TAPSOFT '97: Proceedings of the Seventh Joint Conference on Theory and Practice of Software Development, 7th International Joint Conference CAAP/FASE, Lille, France', Vol. 1214, Springer-Verlag, Lille, France.

Nieuwenhuis, R. (1996), Basic paramodulation and decidable theories (extended abstract), *in* 'Proceedings 11th IEEE Symposium on Logic in Computer Science, LICS'96', IEEE Computer Society Press, pp. 473–482.

Nieuwenhuis, R. & Rubio, A. (1995), 'Theorem proving with ordering and equality constrained clauses', *Journal of Symbolic Computation* **19**, 321–351.

Otto, F., Narendran, P. & Dougherty, D. J. (1995), Some independence results for equational unification, *in* J. Hsiang, ed., 'Rewriting Techniques and Applications, 6th International Conference, RTA-95', LNCS 914, Springer-Verlag, Kaiserslautern, Germany, pp. 367–381.

Oyamaguchi, M. (1990), 'On the word problem for right-ground term-rewriting systems', *The Transactions of the IEICE* **73**(5), 718–723.

Weidenbach, C. (1997), 'Spass version 0.49', *Journal of Automated Reasoning* **18**(2), 247–252.

Weidenbach, C. (1998), Sorted unification and tree automata, *in* W. Bibel & P. H. Schmitt, eds, 'Automated Deduction, A Basis for Applications', Vol. 1, Kluwer, chapter 2.