Deciding the Word Problem in the Union of Equational Theories Sharing Constructors

Franz Baader[1*] and Cesare Tinelli[2]

[1] LuFg Theoretical Computer Science, RWTH Aachen,
Ahornstraße 55, 52074 Aachen, Germany
baader@informatik.rwth-aachen.de
[2] Department of Computer Science, University of Illinois at Urbana-Champaign,
1304 W. Springfield Ave, Urbana, IL 61801 – USA
tinelli@cs.uiuc.edu

Abstract. The main contribution of this paper is a new method for combining decision procedures for the word problem in equational theories sharing "constructors." The notion of constructors adopted in this paper has a nice algebraic definition and is more general than a related notion introduced in previous work on the combination problem.

1 Introduction

The integration of constraint solvers (that is, specialized decision procedures for restricted classes of problems) into general purpose deductive systems (such as Knuth-Bendix completion procedures, resolution-based theorem provers, or Logic Programming systems) aims at combining the efficiency of the specialized method with the universality of the general one. Many applications of the constraint-based systems obtained by such an integration require a combination of more than one constraint language, and thus a solver for the resulting mixed constraints. The development of general combination methods for constraint solvers tries to avoid the necessity of designing a new specialized decision procedure for each new combination of constraint languages.

For equational theories, one is usually interested in solvers for the following decision problems: the word problem, the matching problem, and the unification problem. In this setting, the research on combination of constraint solvers is mainly concerned with finding conditions under which the following question can be answered affirmatively: given two equational theories E_1 and E_2 with decidable word/matching/unification problems, is the word/matching/unification problem for $E_1 \cup E_2$ also decidable?

A very effective (but also rather strong) restriction is to require that E_1 and E_2 be equational theories over disjoint signatures. Under this restriction, decision procedures for the word problems in E_1 and E_2 can always be combined into a decision procedure for the word problem in $E_1 \cup E_2$ [10, 14, 13, 8, 7]. For the matching and the unification problem, there also exist very general combination

* Partially supported by the EC Working Group CCL II.

results under the disjointness restriction (see [12] for matching, and, e.g., [13, 4, 1] for unification). It is not hard to extend these results to theories sharing constant symbols [11, 7, 2]. The only work we are aware of that presents a general combination approach for the union of equational theories having more than constant symbols in common is [6], where the problem of combining algorithms for the unification, matching, and word problem is investigated for theories sharing so-called "constructors."

In this paper, we restrict our attention to the word problem. The combination result we obtain improves on the corresponding result in [6] in the following respects. Firstly, we introduce a notion of constructors, modeled after the one introduced in [15], which is strictly more general than the one in [6]. Whereas [6] does not allow for nontrivial identities between constructor terms, we only require the constructor theory to be collapse-free. Secondly, the definition of constructors in [6] depends strongly on technical details such as the choice of an appropriate well-founded and monotonic ordering. In contrast, our definition uses only abstract algebraic properties. Finally, the combination procedure described in [6], like the ones for the disjoint case [10, 13, 8, 7], directly transforms the terms for which the word problem is to be decided, by applying collapse equations[1] and abstracting alien subterms. This transformation process must be carried on with a rather strict strategy (in principle, going from the leaves of the terms to their roots) and it is not easy to describe. In contrast, our procedure extends the rule-based combination procedure for the word problem introduced in [2] for the case of shared constants. It works on a set of equations rather than terms, and its transformation rules can be applied in arbitrary order, that is, no strategy is needed. We claim that this difference makes the method more flexible and easier to describe and comprehend.

The next section introduces the word problem and describes a reduction of the word problem in the union of equational theories to satisfiability of a conjunction of two pure formulae. Before we can describe our combination procedure, we must introduce our notion of constructors (Section 3). Section 3 also contains some results concerning the union of theories sharing constructors. In Section 4 we describe the new combination procedure for theories sharing constructors, and prove its correctness. Section 5 investigates the connection between our notion of constructors and the one introduced in [6], and includes some remarks on how this work relates to the research on modularity properties of term rewriting systems. Because of the page limit, we cannot give detailed proofs of our results. They can be found in [3].

2 Word Problems and Satisfiability Problems

We will use V to denote a countably infinite set of variables, and $T(\Omega, V)$ to denote the set of all Ω-terms, that is, terms over the signature Ω with variables in V. An equational theory E over the signature Ω is a set of (implicitly universally

[1] i.e., equations of the form $x \equiv t$, where x is a variable occurring in the non-variable term t.

quantified) equations between Ω-terms. We use $s \equiv t$ to denote an equation between the terms s, t. For an equational theory E, the *word problem* is concerned with the validity in E of quantifier-free formulae of the form $s \equiv t$. Equivalently, the word problem asks for the (un)satisfiability of the *disequation* $s \not\equiv t$ in E— where $s \not\equiv t$ is an abbreviation for the formula $\neg(s \equiv t)$. As usual, we often write "$s =_E t$" to express that the formula $s \equiv t$ is valid in E. An equational theory E is *collapse-free* iff $x \neq_E t$ for all variables x and non-variable terms t.

Given an Ω-term s, an Ω-algebra $\mathcal{A}$, and a valuation α (of the variables in s by elements of $\mathcal{A}$), we denote by $[\![s]\!]^{\mathcal{A}}_{\alpha}$ the interpretation of the term s in $\mathcal{A}$ under the valuation α. Also, if Σ is a subsignature of Ω, we denote by $\mathcal{A}^{\Sigma}$ the reduct of $\mathcal{A}$ to the subsignature Σ. An Ω-algebra $\mathcal{A}$ is a model of E iff every equation in E is valid in $\mathcal{A}$. The equational theory E over the signature Ω defines an Ω-*variety*, i.e., the class of all models of E. When E is *non-trivial* i.e., has models of cardinality greater than 1, this variety contains free algebras for any set of generators. We will call these algebras *E-free algebras*. Given a set of generators (or variables) X, an E-free algebra with generators X can be obtained as the quotient term algebra $\mathcal{T}(\Omega, X)/{=_E}$. It is well-known that two E-free algebras with sets of generators of the same cardinality are isomorphic.

In this paper, we are interested in *combined* equational theories, that is, equational theories E of the form $E := E_1 \cup E_2$, where E_1 and E_2 are equational theories over two (not necessarily disjoint) signatures Σ_1 and Σ_2. The elements of $\Sigma_1 \cap \Sigma_2$ are called *shared* symbols. We call 1-*symbols* the elements of Σ_1 and 2-*symbols* the elements of Σ_2. A term $t \in T(\Sigma_1 \cup \Sigma_2, V)$ is an *i-term* iff its *top symbol* $t(\epsilon) \in V \cup \Sigma_i$, i.e., if t is a variable or has the form $t = f(t_1, ..., t_n)$ for some i-symbol f $(i = 1, 2)$. Note that variables and terms t with $t(\epsilon) \in \Sigma_1 \cap \Sigma_2$ are both 1- and 2-terms. A subterm s of a 1-term t is an *alien subterm* of t iff it is not a 1-term and every proper superterm of s in t is a 1-term. Alien subterms of 2-terms are defined analogously. For $i = 1, 2$, an i-term s is *pure* iff it contains only i-symbols and variables. A (dis)equation $s \equiv t$ $(s \not\equiv t)$ is i-pure iff s and t are pure i-terms. It is called pure iff it is i-pure for some $i \in \{1, 2\}$.

A given disequation $s \not\equiv t$ between $(\Sigma_1 \cup \Sigma_2)$-terms s, t can be transformed into an equisatisfiable formula $\varphi_1 \wedge \varphi_2$, where φ_i is a conjunction of i-pure equations and disequations $(i = 1, 2)$. This can be achieved by the usual *variable abstraction* process in which alien subterms are replaced by new variables (see, e.g., [1, 3] for a detailed description of the process). Obviously, if we know that $\varphi_1 \wedge \varphi_2$ is satisfiable in a model $\mathcal{A}$ of $E_1 \cup E_2$, then φ_i is satisfiable in the reduct $\mathcal{A}^{\Sigma_i}$, which is a model of E_i $(i = 1, 2)$. However, the converse need not be true, that is, if φ_i is satisfiable in a model $\mathcal{A}_i$ of E_i $(i = 1, 2)$, then we cannot necessarily deduce that the conjunction $\varphi_1 \wedge \varphi_2$ is satisfiable in some model $\mathcal{A}$ of $E_1 \cup E_2$. One case in which we can is described by the proposition below.

Proposition 1. *Let $\mathcal{A}_i$ be a model of E_i $(i = 1, 2)$, and $\Sigma := \Sigma_1 \cap \Sigma_2$. Assume that the reducts ${\mathcal{A}_1}^{\Sigma}$ and ${\mathcal{A}_2}^{\Sigma}$ are both free in the same Σ-variety and their respective sets of generators Y_1 and Y_2 have the same cardinality. If φ_i is satisfiable in $\mathcal{A}_i$ with the variables in $\mathcal{V}ar(\varphi_1) \cap \mathcal{V}ar(\varphi_2)$ taking distinct values over Y_i for $i = 1, 2$, then there is a model of $E_1 \cup E_2$ in which $\varphi_1 \wedge \varphi_2$ is satisfiable.*

This proposition is a special case of more general results in [15]. A simpler direct proof in the special case can also be found in [3].

In the following, we will consider the case where the algebras $\mathcal{A}_i$ are E_i-free. Unfortunately, the property of being a free algebra is not preserved under signature reduction. The problem is that the reduct of an algebra may need more generators than the algebra itself. For example, consider the signature $\Omega := \{\mathsf{p}, \mathsf{s}\}$ and the equational theory E axiomatized by the equations

$$E := \{x \equiv \mathsf{p}(\mathsf{s}(x)),\ x \equiv \mathsf{s}(\mathsf{p}(x))\}\,.$$

The integers $\mathcal{Z}$ are a free model of E over a set of generators of cardinality 1 when s and p are interpreted as the successor and the predecessor function, respectively. Now, if $\Sigma := \{\mathsf{s}\}$, then $\mathcal{Z}^{\Sigma}$ is definitely not free because it does not even admit a non-redundant set of generators, which is a necessary condition for an algebra to be free.

Nonetheless, there are free algebras admitting reducts that are also free, although over a possibly larger set of generators. These algebras are models of equational theories that admit *constructors* in the sense explained in the next section.

3 Theories Admitting Constructors

In the following, Ω will be an at most countably infinite functional signature, and Σ a subset of Ω. For a given equational theory E over Ω we define the *Σ-restriction of* E as $E^{\Sigma} := \{s \equiv t \mid s, t \in T(\Sigma, V) \text{ and } s =_E t\}$.

Definition 2 (Constructors). *The subsignature Σ of Ω is a* set of constructors for E *if the following two properties hold:*

1. *The Σ-reduct of the countably infinitely generated E-free Ω-algebra is an E^{Σ}-free algebra.*
2. *E^{Σ} is collapse-free.*

This definition is a rather abstract formulation of our requirements on the theory E. In the following, we develop a more concrete characterization[2] of theories admitting constructors, which will make it easier to show that a given theory admits constructors. But first, we must introduce some more notation.

Given a subset G of $T(\Omega, V)$, we denote by $T(\Sigma, G)$ the set of terms over the "variables" G. To express this construction we will denote any such term by $s(\bar{r})$ where $\bar{r}$ is the tuple made of the terms of G that replace the variables of s. Notice that this notation is consistent with the fact that $G \subseteq T(\Sigma, G)$. In fact, every $r \in G$ can be represented as $s(r)$ where s is a variable of V. Also notice that $T(\Sigma, V) \subseteq T(\Sigma, G)$ whenever $V \subseteq G$. In this case, every $s \in T(\Sigma, V)$ can be trivially represented as $s(\bar{v})$ where $\bar{v}$ are the variables of s.

[2] This *characterization* of constructors is a special case of the *definition* of constructors in [15].

For every equational theory E over the signature Ω and every subset Σ of Ω, we define the following subset of $T(\Omega, V)$:

$$G_E(\Sigma, V) := \{r \in T(\Omega, V) \mid r \neq_E f(\bar{t}) \text{ for all } f \in \Sigma \text{ and } \bar{t} \text{ in } T(\Omega, V)\}.$$

We will show that, if Σ is a set of constructors for E, then $G_E(\Sigma, V)$ determines a set of free generators for the Σ-reduct of the countably infinitely generated E-free algebra. But first, let us point out the following properties of $G_E(\Sigma, V)$:

Lemma 3. *Let E be an equational theory over Ω and $\Sigma \subseteq \Omega$.*

1. *$G_E(\Sigma, V)$ is nonempty iff $V \subseteq G_E(\Sigma, V)$;*
2. *If $V \subseteq G_E(\Sigma, V)$, then E^Σ is collapse-free.*

Theorem 4 (Characterization of constructors). *Let $\Sigma \subseteq \Omega$, E a non-trivial equational theory over Ω, and $G := G_E(\Sigma, V)$. Then Σ is a* set of constructors for *E iff the following holds:*

1. *$V \subseteq G$.*
2. *For all $t \in T(\Omega, V)$, there is an $s(\bar{r}) \in T(\Sigma, G)$ such that $t =_E s(\bar{r})$.*
3. *For all $s_1(\bar{r}_1), s_2(\bar{r}_2) \in T(\Sigma, G)$,*

$$s_1(\bar{r}_1) =_E s_2(\bar{r}_2) \text{ iff } s_1(\bar{v}_1) =_E s_2(\bar{v}_2),$$

where $\bar{v}_1, \bar{v}_2$ are fresh variables abstracting $\bar{r}_1, \bar{r}_2$ so that two terms in $\bar{r}_1, \bar{r}_2$ are abstracted by the same variable iff they are equivalent in E.

Actually, the proof of the theorem—which can be found in [3]—provides a little more information than stated in the formulation of the theorem.

Corollary 5. *Let Σ be a set of constructors for E, $\mathcal{A}$ an E-free Ω-algebra with the countably infinite set of generators X, and α a bijective valuation of V onto X. Then, the reduct $\mathcal{A}^\Sigma$ is an E^Σ-free algebra with generators $Y := \{[\![r]\!]^{\mathcal{A}}_\alpha \mid r \in G_E(\Sigma, V)\}$, and $X \subseteq Y$.*

Condition 2 of Theorem 4 says that, when Σ is a set of constructors for E, every Ω-term t is equivalent in E to a term $s(\bar{r}) \in T(\Sigma, G)$ where $G := G_E(\Sigma, V)$. We will call $s(\bar{r})$ *a normal form of t in E*—in general, a term may have more than one normal form. We will say that a term t is *in normal form* if it is already of the form $t = s(\bar{r}) \in T(\Sigma, G)$. Because $V \subseteq G$, it is immediate that Σ-terms are in normal form, as are terms in G. We will say that a term t is *E-reducible* if it is not in normal form. Otherwise, it is *E-irreducible*.

We will make use of normal forms in our combination procedure. In particular, we will consider normal forms that are computable in the following sense.

Definition 6 (Computable Normal Forms). *Let Σ be a set of constructors for the equational theory E over the signature Ω. We say that* normal forms are computable for Σ and E *if there is a computable function*

$$\mathrm{NF}^\Sigma_E \colon T(\Omega, V) \longrightarrow T(\Sigma, G)$$

such that $\mathrm{NF}^\Sigma_E(t)$ is a normal form of t, i.e., $\mathrm{NF}^\Sigma_E(t) =_E t$.

Notice that Definition 6 does not entail that the variables of $\mathrm{NF}_E^\Sigma(t)$ are included in the variables of t. However, if $V_0 := \mathcal{V}ar(\mathrm{NF}_E^\Sigma(t)) \setminus \mathcal{V}ar(t)$ is nonempty, then $\pi(\mathrm{NF}_E^\Sigma(t))$ is also a normal form of t for any injective renaming π of the variables in V_0. Consequently, if V_1 is a given finite subset of V, we can always assume without loss of generality that $\mathcal{V}ar(\mathrm{NF}_E^\Sigma(t)) \setminus \mathcal{V}ar(t)$ and V_1 are disjoint.[3] As a rule then we will *always* assume that the variables occurring in a normal form $\mathrm{NF}_E^\Sigma(t)$ but not in t, if any, are *fresh* variables.

An important consequence of Definition 6 is that, when normal forms are computable for Σ and E, it is always possible to tell whether a term is in normal form or not.

Proposition 7. *Let Σ be a set of constructors for the equational theory E over the signature Ω and assume that normal forms are computable for Σ and E. Then, the E-reducibility of terms in $T(\Omega, V)$ is decidable.*

We provide below two examples of equational theories admitting constructors in the sense of Definition 2. But first, let us consider some counter-examples:

- The signature $\Sigma := \Omega := \{f\}$ is not a set of constructors for the theory E axiomatized by $\{x \equiv f(x)\}$ because Definition 2(2) is not satisfied.
- The signature $\Sigma := \{f\} \subseteq \{f, g\} =: \Omega$ is not a set of constructors for the theory E axiomatized by $\{g(x) \equiv f(g(x))\}$ because Theorem 4(2) is not satisfied. In fact, the term $g(x)$ does not have a normal form. (The signature $\{f, g\}$, however, is a set of constructors for the same theory.)
- Finally, take $\Omega := \{f, g\}$ and $\Sigma := \{f\}$ and consider the theory $E := \{f(g(x)) \equiv f(f(g(x)))\}$. Then we have $G_E(\Sigma, V) = V \cup \{g(t) \mid t \in T(\Omega, V)\}$. It is easy to see that conditions (1) and (2) of Theorem 4 hold. However, condition (3) does not hold since $f(g(x)) =_E f(f(g(x)))$, although $f(y) \neq_E f(f(y))$.

Example 8. The theory of the natural numbers with addition is the most immediate example of a theory with constructors. Consider the signature $\Sigma_1 := \{0, \mathsf{s}, +\}$ and the equational theory E_1 axiomatized by the equations below:

$$x + (y + z) \equiv (x + y) + z, \quad x + y \equiv y + x, \quad x + \mathsf{s}(y) \equiv \mathsf{s}(x + y), \quad x + 0 \equiv x.$$

It can be shown that the signature $\Sigma := \{0, \mathsf{s}\}$ is a set of constructors for E_1 in the sense of Definition 2. The proof in [3] uses the fact that orienting the third and fourth equation from left to right yields a canonical term rewrite system modulo the first two equations. Note that the restriction of E_1 to Σ (i.e., the theory ${E_1}^\Sigma$) is the syntactic equality of Σ-terms.

Example 9. Consider the signature $\Sigma_2 := \{0, 1, \mathsf{rev}, \cdot\}$ and the equational theory E_2 axiomatized by the equations below:

$$\begin{array}{lll} x \cdot (y \cdot z) \equiv (x \cdot y) \cdot z, & \mathsf{rev}(0) \equiv 0, & \mathsf{rev}(1) \equiv 1, \\ \mathsf{rev}(x \cdot y) \equiv \mathsf{rev}(y) \cdot \mathsf{rev}(x), & \mathsf{rev}(\mathsf{rev}(x)) \equiv x. & \end{array}$$

[3] Otherwise, we apply an appropriate renaming that produces a normal form of t satisfying such disjointness condition.

The signature $\Sigma' := \{0, 1, \cdot\}$ is a set of constructors for E_2 in the sense of Definition 2. The proof in [3] depends on the fact that orienting the equations from left to right yields a canonical term rewriting system. This example differs from the previous one in that the restriction of the theory to the constructor signature is no longer syntactic equality: ${E_2}^{\Sigma}$ expresses associativity of "$\cdot$".

Combination of Theories Sharing Constructors

For the next results, in which we go back to the problem of combining equational theories, we will consider two non-trivial equational theories E_1, E_2 with respective countable signatures Σ_1, Σ_2 such that $\Sigma := \Sigma_1 \cap \Sigma_2$ is a set of constructors for E_1 and for E_2, and ${E_1}^{\Sigma} = {E_2}^{\Sigma}$.

The proposition below—which is important in the proof of correctness of our combination procedure—is an easy consequence of Proposition 1 and Corollary 5.

Proposition 10. *For $i = 1, 2$, let $\mathcal{A}_i$ be an E_i-free Σ_i-algebra with a countably infinite set X_i of generators, and let $Y_i := \{[\![r]\!]^{\mathcal{A}_i}_{\alpha_i} \mid r \in G_E(\Sigma_i, V)\}$, where α_i is any bijective valuation of V onto X_i. Let φ_1, φ_2 be conjunctions of equations and disequations of respective signature Σ_1, Σ_2. If φ_i is satisfiable in $\mathcal{A}_i$ with $\mathcal{V}ar(\varphi_1) \cap \mathcal{V}ar(\varphi_2)$ taking distinct values over Y_i for $i = 1, 2$, then $\varphi_1 \wedge \varphi_2$ is satisfiable in $E_1 \cup E_2$.*

The following theorem shows that being a set of constructors is a modular property. Thus, the application of the combination procedure described in the next section can be iterated.

Theorem 11. *Let E_1, E_2 be two non-trivial equational theories with respective signatures Σ_1, Σ_2 such that $\Sigma := \Sigma_1 \cap \Sigma_2$ is a set of constructors for E_1 and for E_2, ${E_1}^{\Sigma} = {E_2}^{\Sigma}$, the word problem for E_i is decidable, and normal forms are computable for Σ and E_i for $i = 1, 2$. Then, the following holds:*

1. *Σ is a set of constructors for $E := E_1 \cup E_2$.*
2. *$E^{\Sigma} = {E_1}^{\Sigma} = {E_2}^{\Sigma}$.*
3. *Normal forms are computable for Σ and E.*

The (quite involved) proof in [3] shows that the three conditions in Theorem 4 are satisfied. It depends on an appropriate characterization of $G_E(\Sigma, V)$. Modulo E, this set is identical to the set G' defined below.

Definition 12. *For $i = 1, 2$, let $G_i := G_{E_i}(\Sigma, V)$. The set G' is inductively defined as follows:*

1. *Every variable is an element of G', that is, $V \subseteq G'$.*
2. *Assume that $r(\bar{v}) \in G_i$ for $i \in \{1, 2\}$ and $\bar{r}$ is a tuple of elements of G' such that the following conditions are satisfied:*
   (a) $r(\bar{v}) \neq_E v$ for all variables $v \in V$;
   (b) $r_k(\epsilon) \notin \Sigma_i$ for all components r_k of $\bar{r}$;
   (c) the tuple $\bar{v}$ consists of all variables of r without repetitions;

(d) *the tuples* $\bar{v}$ *and* $\bar{r}$ *have the same length;*
(e) $r_k \neq_E r_\ell$ *if* r_k, r_ℓ *occur at different positions in the tuple* $\bar{r}$.
Then $r(\bar{r}) \in G'$.

Notice that $G_i \subseteq G'$ for $i = 1, 2$ because the components of $\bar{r}$ above can also be variables. Also notice that no element r of G' can have a shared symbol as top symbol since r is either a variable or a term "starting" with an element of G_i.

4 A Combination Procedure for the Word Problem

In this section, we will present a combination procedure that allows us to derive the following decidability result for the word problem in the union of equational theories sharing constructors:

Theorem 13. *Let* E_1, E_2 *be two non-trivial equational theories of signature* Σ_1, Σ_2, *respectively, such that* $\Sigma := \Sigma_1 \cap \Sigma_2$ *is a set of constructors for both* E_1 *and* E_2, *and* ${E_1}^{\Sigma} = {E_2}^{\Sigma}$. *If for* $i = 1, 2$,

- *normal forms are computable for* Σ *and* E_i, *and*
- *the word problem in* E_i *is decidable,*

then the word problem in $E_1 \cup E_2$ *is also decidable.*

From Theorem 11 it follows that, given the right conditions, the combination procedure applies immediately by recursion to more than two theories:

Corollary 14. *Let* Σ *be a signature and* $E_1, \ldots, E_n$ *be* n *equational theories of signature* $\Sigma_1, \ldots, \Sigma_n$, *respectively, such that* $\Sigma = \Sigma_i \cap \Sigma_j$ *and* ${E_i}^{\Sigma} = {E_j}^{\Sigma}$ *for all distinct* $i, j \in \{1, \ldots, n\}$. *Also, assume that* Σ *is a set of constructors for every* E_i. *If for all* $i \in \{1, \ldots, n\}$,

- *normal forms are computable for* Σ *and* E_i, *and*
- *the word problem in* E_i *is decidable,*

then the word problem in $E_1 \cup \cdots \cup E_n$ *is decidable and normal forms are computable for* Σ *and* $E_1 \cup \cdots \cup E_n$.

As shown in Section 2, the word problem for $E := E_1 \cup E_2$ can be reduced to the satisfiability problem for disequations of the form $s_0 \not\equiv t_0$, where s_0 and t_0 are $(\Sigma_1 \cup \Sigma_2)$-terms. By variable abstraction, this disequation can be transformed into an equisatisfiable formula $\varphi_1 \wedge \varphi_2$, where φ_i is a conjunction of i-pure equations and disequations ($i = 1, 2$). We will use finite sets of (dis)equations in place of conjunctions of such formulae, and say that a set of (dis)equations is satisfiable in a theory iff the conjunction of its elements is satisfiable in that theory. It turns out that the finite set of (dis)equations obtained by applying variable abstraction is what we call an *abstraction system*. Before we can define this notion, we must introduce some notation.

Let $x, y \in V$ and T be a set of equations of the form $v \equiv t$ where $v \in V$ and $t \in T(\Sigma_1 \cup \Sigma_2, V) \setminus V$. The relation $\prec$ is the smallest binary relation on $\{x \not\equiv y\} \cup T$ such that, for all $u \equiv s, v \equiv t \in T$,

$$(x \not\equiv y) \prec (v \equiv t) \text{ iff } v \in \{x, y\},$$
$$(u \equiv s) \prec (v \equiv t) \text{ iff } v \in \mathcal{V}ar(s).$$

By $\prec^+$ we denote the transitive and by $\prec^*$ the reflexive-transitive closure of $\prec$. The relation $\prec$ is *acyclic* if there is no equation $v \equiv t$ in T such that $(v \equiv t) \prec^+ (v \equiv t)$.

Definition 15 (Abstraction System). *The set* $S := \{x \not\equiv y\} \cup T$ *is an* abstraction system with initial formula $x \not\equiv y$ *iff* $x, y \in V$ *and the following holds:*

1. *T is a finite set of equations of the form $v \equiv t$ where $v \in V$ and $t \in (T(\Sigma_1, V) \cup T(\Sigma_2, V)) \setminus V$;*
2. *the relation $\prec$ on S is acyclic;*
3. *for all $(u \equiv s), (v \equiv t) \in T$,*
   (a) *if $u = v$ then $s = t$;*
   (b) *if $(u \equiv s) \prec (v \equiv t)$ and $s \in T(\Sigma_i, V)$ with $i \in \{1, 2\}$ then $t(\epsilon) \notin \Sigma_i$.*

Condition (1) above states that T consists of equations between variables and pure non-variable terms; Condition (2) implies that for all $(u \equiv s), (v \equiv t) \in T$, if $(u \equiv s) \prec^* (v \equiv t)$ then $u \notin \mathcal{V}ar(t)$; Condition (3a) implies that a variable cannot occur as the left-hand side of more than one equation of T; Condition (3b) implies, together with Condition (1), that the elements of every $\prec$-chain of T have *strictly* alternating signatures $(\ldots, \Sigma_1, \Sigma_2, \Sigma_1, \Sigma_2, \ldots)$.

Every abstraction system S induces a finite graph $\mathcal{G}_S := (S, \prec)$ whose set of *nodes* is S and whose set of *edges* consists of all pairs $(n_1, n_2) \in S \times S$ such that $n_1 \prec n_2$. According to Definition 15, $\mathcal{G}_S$ is in fact a directed acyclic graph (or *dag*). Assuming the standard definition of path between two nodes and of length of a path in a dag, the *height* $\mathsf{h}(n)$ of the node n is the maximum of the lengths of all the paths in the dag that end with n.[4]

We say that an equation of an abstraction system S is *reducible* iff its right-hand side is E_i-reducible (i.e., not in normal form) for $i = 1$ or $i = 2$. The disequation in S is always irreducible. In the previous section, we would have represented the normal form of a term in $T(\Sigma_i, V)$ $(i = 1, 2)$ as $s(\bar{q})$ where s was a term in $T(\Sigma, V)$ and $\bar{q}$ a tuple of terms in $G_{E_i}(\Sigma, V)$. Considering that $G_{E_i}(\Sigma, V)$ contains V because of the assumption that Σ is a set of constructors, we will now use a more descriptive notation. We will distinguish the variables in $\bar{q}$ from the non-variables terms and write $s(\bar{y}, \bar{r})$ instead, where $\bar{y}$ collects the elements of $\bar{q}$ that are in V and $\bar{r}$ those that are in $G_{E_i}(\Sigma, V) \setminus V$.

The combination procedure described in Fig. 1 decides the word problem for the theory $E := E_1 \cup E_2$ by deciding the satisfiability in E of disequations of the form $s_0 \not\equiv t_0$ where s_0, t_0 are $(\Sigma_1 \cup \Sigma_2)$-terms. During the execution of the

[4] Since $\mathcal{G}_S$ is acyclic and finite, this maximum exists.

Input: $(s_0, t_0) \in T(\Sigma_1 \cup \Sigma_2, V) \times T(\Sigma_1 \cup \Sigma_2, V)$.

1. Let S be the abstraction system obtained by applying variable abstraction to $s_0 \not\equiv t_0$.
2. Repeatedly apply (in any order) **Coll1**, **Coll2**, **Ident**, **Simpl**, **Shar1**, **Shar2** to S until none of them is applicable.
3. Succeed if S has the form $\{v \not\equiv v\} \cup T$ and fail otherwise.

Fig. 1. The Combination Procedure.

procedure, the set S of formulae on which the procedure works is repeatedly modified by the application of one of the derivation rules defined in Fig. 2. We describe these rules in the style of a sequent calculus. The premise of each rule lists all the formulae in S before the application of the rule, where T stands for all the formulae not explicitly listed. The conclusion of the rule lists all the formulae in S after the application of the rule. It is understood that any two formulae explicitly listed in the premise of a rule are distinct.

In essence, **Coll1** and **Coll2** remove from S collapse equations that are valid in E_1 or E_2, while **Ident** identifies any two variables equated to equivalent Σ_i-terms and then discards one of the corresponding equations. The restriction that the height of $y \equiv t$ be not smaller than the height of $x \equiv s$ is there to preserve the acyclicity of $\prec$. In these rules we have used the notation $t[y]$ to express that the variable y occurs in the term t, and the notation $T[x/t]$ to denote the set of formulae obtained by substituting every occurrence of the variable x by the term t in the set T.

Simpl eliminates those equations that have become unreachable along a $\prec$-path from the initial disequation because of the application of previous rules. This rule is not essential but it reduces clutter in S by eliminating equations that do not contribute to the solution of the problem anymore. It can be used to obtain optimized, complete implementations of the combination procedure.

The main idea of **Shar1** and **Shar2** is to push shared symbols towards lower positions of the $\prec$-chains they belong to so that they can be processed by other rules. To do that the rules replace the reducible right-hand side t of an equation $x \equiv t$ by its normal form, and then plug the "shared part" of the normal form into all equations whose right-hand sides contain x. The exact formulation of the rules is somewhat more complex since we must ensure that the resulting system is again an abstraction system. In particular, the "alternating signature" condition (3b) of Definition 15 must be respected.

In the description of the rules, an expression like $\bar{z} \equiv \bar{r}$ denotes the set $\{z_1 \equiv r_1, \ldots, z_n \equiv r_n\}$ where $\bar{z} = (z_1, \ldots, z_n)$ and $\bar{r} = (r_1, \ldots, r_n)$, and $s(\bar{y}, \bar{z})$ denotes the term obtained from $s(\bar{y}, \bar{r})$ by replacing the subterm r_j with z_j for each $j \in \{1, \ldots, n\}$. Observe that this notation also accounts for the possibility that t reduces to a non-variable term of $G_{E_i}(\Sigma, V)$. In that case, s will be a variable, $\bar{y}$ will be empty, and $\bar{r}$ will be a tuple of length 1. Substitution expres-

$$\textbf{Coll1}\ \frac{T \quad u \not\equiv v \quad x \equiv t[y] \quad y \equiv r}{T[x/r] \quad (u \not\equiv v)[x/y] \quad y \equiv r}$$

if t is an i-term and $y =_{E_i} t$ for $i = 1$ or $i = 2$.

$$\textbf{Coll2}\ \frac{T \quad x \equiv t[y]}{T[x/y]}$$

if t is an i-term and $y =_{E_i} t$ for $i = 1$ or $i = 2$ and there is no $(y \equiv r) \in T$.

$$\textbf{Ident}\ \frac{T \quad x \equiv s \quad y \equiv t}{T[x/y] \quad y \equiv t}$$

if s, t are i-terms and $s =_{E_i} t$ for $i = 1$ or $i = 2$ and $x \neq y$ and $\mathsf{h}(x \equiv s) \leq \mathsf{h}(y \equiv t)$.

$$\textbf{Simpl}\ \frac{T \quad x \equiv t}{T}$$

if $x \notin \mathcal{V}ar(T)$.

$$\textbf{Shar1}\ \frac{T \quad u \not\equiv v \quad x \equiv t \quad \bar{y}_1 \equiv \bar{r}_1}{T[x/s(\bar{y}, \bar{z})[\bar{y}_1/\bar{r}_1]] \quad \bar{z} \equiv \bar{r} \quad u \not\equiv v \quad x \equiv s(\bar{y}, \bar{r}) \quad \bar{y}_1 \equiv \bar{r}_1}$$

if (a) t is an E_i-reducible i-term for $i = 1$ or $i = 2$,
(b) $\mathrm{NF}^{\Sigma}_{E_i}(t) = s(\bar{y}, \bar{r}) \notin V$,
(c) $\bar{r}$ non-empty,
(d) $\bar{z}$ fresh variables with no repetitions,
(e) $\bar{r}_1$ irreducible (for both theories),
(f) $\bar{y}_1 \subseteq \mathcal{V}ar(s(\bar{y}, \bar{r}))$ and $(x \equiv s(\bar{y}, \bar{r})) \prec (y \equiv r)$ for no $(y \equiv r) \in T$.

$$\textbf{Shar2}\ \frac{T \quad u \not\equiv v \quad x \equiv t \quad \bar{y}_1 \equiv \bar{r}_1}{T[x/s[\bar{y}_1/\bar{r}_1]] \quad u \not\equiv v \quad x \equiv s[\bar{y}_1/\bar{r}_1] \quad \bar{y}_1 \equiv \bar{r}_1}$$

if (a) t is an E_i-reducible i-term for $i = 1$ or $i = 2$,
(b) $\mathrm{NF}^{\Sigma}_{E_i}(t) = s \in T(\Sigma, V) \setminus V$,
(c) $\bar{r}_1$ irreducible (for both theories),
(d) $\bar{y}_1 \subseteq \mathcal{V}ar(s)$ and $(x \equiv s) \prec (y \equiv r)$ for no $(y \equiv r) \in T$.

Fig. 2. The Derivation Rules.

sions containing tuples are to be interpreted accordingly; e.g., $[\bar{z}/\bar{r}]$ replaces the variable z_j by r_j for each $j \in \{1, \ldots, n\}$.

In both **Shar** rules it is assumed that the normal form is not a variable. The reason for this restriction is that the case where an i-term is equal modulo E_i to a variable is already taken care of by the rules **Coll1** and **Coll2**. By requiring that $\bar{r}$ be non-empty, **Shar1** excludes the possibility that the normal form of the term t is a shared term. It is **Shar2** that deals with this case. The reason for a separate case is that we want to preserve the property that every $\prec$-chain is made of equations with alternating signatures (cf. Definition 15(3b)). When the equation $x \equiv t$ has immediate $\prec$-successors, the replacement of t by the Σ-term

s may destroy the alternating signatures property because $x \equiv s$, which is both a Σ_1- and a Σ_2-equation, may inherit some of these successors from $x \equiv t$.[5] **Shar2** restores this property by merging into s all the immediate successors of $x \equiv s$—which are collected, if any, in the set $\bar{y}_1 \equiv \bar{r}_1$. Condition (d) in **Shar2** makes sure that the tuple $\bar{y}_1 \equiv \bar{r}_1$ collects all these successors. The replacement of $\bar{y}_1$ by $\bar{r}_1$ in **Shar1** is done for similar reasons. In both **Shar** rules, the restriction that all the terms in $\bar{r}_1$ be in normal form is necessary to ensure termination.

A sketch of the Correctness Proof

As a first step to proving the correctness of the combination procedure, we can show that an application of one of the rules of Fig. 2 transforms abstraction systems into abstraction systems, preserves satisfiability, and leads to a decrease w.r.t. a certain well-founded ordering. This ordering can be obtained as follows: every node in the dag corresponding to the abstraction system S is associated with a pair (h, r), where h is the height of the node, and r is 1 if the corresponding (dis)equation is reducible, and 0 otherwise. The abstraction system S is associated with the multiset $M(S)$ consisting of all these pairs. Let $\sqsupset$ be the multiset ordering [5] induced by the lexicographic ordering on pairs.

Lemma 16. *Assume that S' is obtained from S by an application of one of the rules of Fig. 2.*

1. *If S is an abstraction system, then so is S'.*
2. *S is satisfiable in $E_1 \cup E_2$ iff S' is satisfiable in $E_1 \cup E_2$.*
3. *$M(S) \sqsupset M(S')$.*

The second point of the lemma implies soundness of our combination procedure, that is, if the combination procedure succeeds on an input (s_0, t_0), then $s_0 =_{E_1 \cup E_2} t_0$. Since the multiset ordering $\sqsupset$ is well-founded, the third point implies that the procedure always terminates. The first point implies that the final system obtained after the termination of the procedure is an abstraction system. This fact plays an important rôle in the proof of completeness of the procedure. The completeness of the combination procedure, meaning that the procedure succeeds on an input (s_0, t_0) whenever $s_0 =_{E_1 \cup E_2} t_0$, can be proved by showing that Proposition 10 can be applied (see [3] for details).

5 Related work

In this section, we investigate the connection between our notion of constructors and the one introduced in [6]. Before we can define the notion of constructors according to [6], called DKR-constructors in the following, we need to introduce

[5] Recall that we assume, without loss of generality, that the variables in $\mathcal{V}ar(s) \setminus \mathcal{V}ar(t)$ do not occur in the abstraction system (cf. the remark after Definition 6). Thus, the equations in $\bar{y} \equiv \bar{r}$ are in fact successors of $x \equiv t$.

the notion of a monotonic ordering. An ordering on $T(\Omega, V)$ is called monotonic if $s > t$ implies $f(\ldots, s, \ldots) > f(\ldots, t, \ldots)$ for all $s, t \in T(\Omega, V)$ and all function symbols $f \in \Omega$. In the rest of the section, we will consider a non-trivial equational theory E of signature Ω and a subsignature Σ of Ω.

Definition 17. *Let $>$ be a well-founded and monotonic ordering on $T(\Omega, V)$. The signature Σ is a* set of DKR-constructors for E w.r.t. $>$ *if*

1. *the $=_E$ congruence class of any term $t \in T(\Omega, V)$ contains a least element w.r.t. $>$, which we denote by $t{\downarrow}_E^>$, and*
2. *$f(t_1, \ldots, t_n){\downarrow}_E^> = f(t_1{\downarrow}_E^>, \ldots, t_n{\downarrow}_E^>)$ for all $f \in \Sigma$ and Ω-terms $t_1, \ldots, t_n$.*

We will call $t{\downarrow}_E^>$ the DKR-normal form of t, and then say that t is in DKR-normal form whenever $t = t{\downarrow}_E^>$. For the theory E_1 in Example 8, it is not hard to show that the signature Σ is set of DKR-constructors for E_1 w.r.t. an appropriate well-founded and monotonic ordering.

Example 9 shows that a set of constructors in the sense of Definition 2 need not be a set of DKR-constructors. In fact, as shown in [6], the definition of DKR-constructors implies that, if Σ is a set of DKR-constructors for E, then E^Σ is the theory of syntactic equality on Σ-terms. This implies that, in Example 9, the signature Σ' is not a set of DKR-constructors for E_2.

To show that the notion of DKR-constructors is a special case of our notion of constructors, we need a representation of the set $G_E(\Sigma, V)$.

Lemma 18. *Let Σ be a set of* DKR*-constructors for E w.r.t. $>$. Then $G_E(\Sigma, V) = \{r \in T(\Omega, V) \mid r{\downarrow}_E^>(\epsilon) \notin \Sigma\}$.*

Using this lemma, it is not hard to show the next proposition.

Proposition 19. *If Σ is a set of* DKR*-constructors for E w.r.t. $>$, then Σ is a set of constructors for E according to Definition 2.*

The definition of DKR-constructors does not assume that DKR-normal forms are computable. In [6], this is achieved by additionally assuming that the so-called symbol matching problem is decidable.

Definition 20. *We say that the* symbol matching problem on Σ modulo E *is decidable in $T(\Omega, V)$ if there exists an algorithm that decides, for all $t \in T(\Omega, V)$, whether there exists a function symbol $f \in \Sigma$ and a tuple of Ω-terms $\bar{t}$ such that $t =_E f(\bar{t})$. We say that t* matches onto Σ modulo E *if $t =_E f(\bar{t})$ for some $f \in \Sigma$ and some tuple $\bar{t}$ of Ω-terms.*

As pointed out in [6], if the symbol matching problem and the word problem are decidable for E, then a symbol $f \in \Sigma$ and a tuple of terms $\bar{t}$ satisfying $t =_E f(\bar{t})$ can be effectively computed, whenever it exists. In fact, once we know that an appropriate function symbol in Σ and a tuple of Ω-terms exists, we can simply enumerate all pairs consisting of a symbol $f \in \Sigma$ and a tuple $\bar{t}$ of Ω-terms, and test whether $t =_E f(\bar{t})$. We call an algorithm that realizes such a computation a *symbol matching algorithm on Σ modulo E*. Using such a symbol matching algorithm, we can define a function NF_E^Σ for E and Σ with the following recursive definition.

Definition 21. *Assume that Σ is set of* DKR*-constructors for E w.r.t. $>$, the word problem for E and the symbol matching problem on Σ modulo E are decidable, and let M be any symbol matching algorithm on Σ modulo E. Then, let NF_E^Σ be the function defined as follows: For every $t \in T(\Omega, V)$,*

1. *$\mathrm{NF}_E^\Sigma(t) := f(\mathrm{NF}_E^\Sigma(t_1), \ldots, \mathrm{NF}_E^\Sigma(t_n))$ if t matches onto Σ modulo E and f is the Σ-symbol and $(t_1, \ldots, t_n)$ the tuple of Ω-terms returned by M on input t.*
2. *$\mathrm{NF}_E^\Sigma(t) := t$, otherwise.*

Lemma 22. *Under the assumptions of Definition 21 the function NF_E^Σ is well-defined and satisfies the requirements of Definition 6.*

This lemma, together with Proposition 19, entails that Theorem 14 in [6] can be obtained as a corollary of our Theorem 13.

Corollary 23. *Let E_1, E_2 be non-trivial equational theories of signature Σ_1, Σ_2, respectively, such that $\Sigma := \Sigma_1 \cap \Sigma_2$ is a set of* DKR*-constructors for both E_1 and E_2. If for $i = 1, 2$, the symbol matching problem on Σ modulo E_i is decidable, and the word problem in E_i is decidable, then the word problem in $E_1 \cup E_2$ is also decidable.*

A third notion of constructors has been introduced in term rewriting in the context of modularity properties for term rewriting systems: a constructor is a function symbol that does not occur at the top of a left-hand side of a rule. It is easy to see that, for complete (i.e., confluent and strongly normalizing) term rewriting systems, this notion of constructors is a special case of the notion of DKR-constructors. A finite complete term rewriting system provides a decision procedure for the word problem. Although the union of two complete term rewriting systems sharing constructors need not be complete, this union is at least semi-complete (i.e., confluent and weakly normalizing), which is sufficient to obtain a decision procedure for the word problem (see, e.g., [9] for details). The main difference between this combination result and ours, in addition to the greater generality of our constructors, is that we do not assume that the word problem in the component theories can be decided by a complete or semi-complete term rewriting system, that is, our approach also applies in cases where the decision procedure is not based on term rewriting.

6 Future Work

As mentioned in the introduction, [6] also contains combination results for unification and matching, whereas the present paper is concerned only with the word problem. Thus, one direction for future research would be to extend our approach to the combination of decision procedures for the matching and the unification problem as well.

Another direction would be to extend the class of theories even further by relaxing the restriction that the equational theory over the constructors be collapse-free. A crucial artifact to our completeness proof is the set $G_E(\Sigma, V)$, which is used to obtain the (countably infinite) set of generators of a certain free algebra. When the equational theory over the constructors is not collapse-free, $G_E(\Sigma, V)$ is empty, and thus cannot be used to describe this set of generators. An appropriate alternative characterization of the set of generators might allow us to remove altogether the restriction that the equational theory over the constructors be collapse-free.

References

1. F. Baader and K.U. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. *J. Symbolic Computation* 21, 1996.
2. F. Baader and C. Tinelli. A new approach for combining decision procedures for the word problem, and its connection to the Nelson-Oppen combination method. In *Proc. CADE-14*, Springer LNAI 1249, 1997.
3. F. Baader and C. Tinelli. Deciding the word problem in the union of equational theories. UIUCDCS-Report 98-2073, Department of Computer Science, University of Illinois at Urbana-Champaign, 1998. Available at `http://www-lti.informatik.rwth-aachen.de/Forschung/Papers.html`.
4. A. Boudet. Combining unification algorithms. *J. Symbolic Computation* 16, 1993.
5. N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Communications of the ACM* 22, 1979.
6. E. Domenjoud, F. Klay, and C. Ringeissen. Combination techniques for non-disjoint equational theories. In *Proc. CADE-12*, Springer LNAI 814, 1994.
7. H. Kirchner and Ch. Ringeissen. Combining symbolic constraint solvers on algebraic domains. *J. Symbolic Computation* 18, 1994.
8. T. Nipkow. Combining matching algorithms: The regular case. In *Proc. RTA'89*, Springer LNCS 335, 1989.
9. E. Ohlebusch. Modular properties of composable term rewriting systems. *J. Symbolic Computation* 20, 1995.
10. D. Pigozzi. The join of equational theories. *Colloquium Mathematicum* 30, 1974.
11. Ch. Ringeissen. Unification in a combination of equational theories with shared constants and its application to primal algebras. In *Proc. LPAR'92*, Springer LNAI 624, 1992.
12. Ch. Ringeissen. Combination of matching algorithms. In *Proc. STACS-11*, Springer LNCS 775, 1994.
13. M. Schmidt-Schauß. Combination of unification algorithms. *J. Symbolic Computation* 8, 1989.
14. E. Tidén. *First-Order Unification in Combinations of Equational Theories.* Phd thesis, The Royal Institute of Technology, Stockholm, 1986.
15. C. Tinelli and Ch. Ringeissen. Non-disjoint unions of theories and combinations of satisfiability procedures: First results. Technical Report UIUCDCS-R-98-2044, Department of Computer Science, University of Illinois at Urbana-Champaign, April 1998. (Also available as INRIA research report no. RR-3402.)