RWTH
LTCS–Report

Aachen University of Technology
Research group for
Theoretical Computer Science

A New Approach for Combining Decision Procedures for the Word Problem, and Its Connection to the Nelson-Oppen Combination Method

Franz Baader Cesare Tinelli

LTCS-Report 96-01

To appear in *Proc. CADE'97.*

RWTH Aachen
LuFg Theoretische Informatik
http://www-lti.informatik.rwth-aachen.de

Ahornstr. 55
52074 Aachen
Germany

A New Approach for Combining Decision Procedures for the Word Problem, and Its Connection to the Nelson-Oppen Combination Method

Franz Baader*
LuFG Theoretical Computer Science, RWTH Aachen
Ahornstraße 55, 52074 Aachen, Germany
email: baader@informatik.rwth-aachen.de

Cesare Tinelli†
Department of Computer Science
University of Illinois at Urbana-Champaign
1304 W. Springfield Ave, Urbana, IL 61801, USA
email: tinelli@cs.uiuc.edu

Abstract

The Nelson-Oppen combination method can be used to combine decision procedures for the validity of quantifier-free formulae in first-order theories with disjoint signatures, provided that the theories to be combined are stably infinite. We show that, even though equational theories need not satisfy this property, Nelson and Oppen's method can be applied, after some minor modifications, to combine decision procedures for the validity of quantifier-free formulae in equational theories. Unfortunately, and contrary to a common belief, the method cannot be used to combine decision procedures for the word problem. We present a method that solves this kind of combination problem. Our method is based on transformation rules and also applies to equational theories that share a finite number of constant symbols.

1 Introduction

Equational theories, that is, theories defined by a set of (implicitly universally quantified) equational axioms of the form $s \equiv t$, and their appropriate treatment in theorem provers play an important rôle in research on automated deduction. On the one hand, equational axioms occur in many

*Supported by DFG SPP "Deduktion" and the EC Working Group CCL II.
†Supported by grant DACA88-94-0014 from the US Army Construction Engineering Laboratories

axiom sets handled by theorem provers since they define common mathematical properties of operators (such as associativity, commutativity). On the other hand, the straightforward approach for treating equality (namely, axiomatizing the special properties of equality, and adding these axioms to the input axioms of the prover) often leads to unsatisfactory results. This explains the interest in developing special inference methods and decision procedures for handling equational theories.

The word problem, that is, the problem of whether an equation $s \equiv t$ is entailed by a given equational theory E, is the most basic decision problem for equational theories. It is, of course, undecidable, as exemplified by the undecidability of the word problem for finitely presented semigroups [6]. Nevertheless, there are decidability results for certain classes of equational theories (such as theories defined by a finite set of ground equations [8]), and there are general approaches for tackling the word problem (such as Knuth-Bendix completion [4], which tries to generate a confluent and terminating term rewriting system for the theory).

The present paper is concerned with the question of whether the decidability of the word problem is a modular property of equational theories: given two equational theories E_1 and E_2 with decidable word problems, is the word problem for $E_1 \cup E_2$ also decidable? In this general formulation, the answer is obviously no, with the word problem for semigroups again providing a counterexample. In fact, consider a finitely presented semigroup with undecidable word problem. The set of equational axioms corresponding to the semigroup's presentation can be seen as the union of a set A axiomatizing the associativity of the semigroup operation, and a set G of ground equations corresponding to the defining relations of the presentation. The word problem for G is decidable, since G is a finite set of ground equations, and it is quite obvious that the word problem for A is decidable as well. But the word problem for $A \cup G$ is just the word problem for the presented semigroup, which is undecidable by assumption.

The theories A and G of this example share a function symbol (the binary semigroup operation). What happens if we assume that there are no shared symbols, that is, the theories to be combined are built over disjoint signatures? Modularity properties for term rewriting systems over disjoint signatures have been studied in detail. It has turned out that confluence is a modular property [18], but unfortunately termination is not. In [17] it is shown that there exist two confluent and terminating rewrite systems over disjoint signatures such that their union is not terminating. Thus, the union of systems that provide a decision procedure for the word problem in the single theories does not yield a decision procedure for the word problem in the combined theory.

Nevertheless, decision procedures for the word problem can be combined in the case of disjoint signatures (independently of where these decision procedures come from), that is, if E_1 and E_2 are equational theories over disjoint signatures, and both have a decidable word problem, then $E_1 \cup E_2$

has a decidable word problem as well. This was shown in [14, 13, 9, 3]. Surprisingly, this combination result does not appear to be widely known, possibly because it was obtained and presented as a side result of the research on combining matching and unification algorithms. As a matter of fact, although the result in principle follows from a technical lemma in [14], it is not explicitly stated there; in [13, 3] it is stated as a corollary, but not mentioned in the abstract or the introduction; only [9] explicitly refers to the result in the abstract. The combination methods used in these articles are essentially identical, the main differences lying in the proofs of correctness. They all directly transform 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 and comprehend.

In this paper, we introduce a new method for combining decision procedures for the word problem that works on a set of equations rather than terms. Its transformation rules can be applied in arbitrary order, that is, no strategy is needed. Thus, the difference between this new approach and the old ones is similar to the difference between Martelli and Montanari's transformation-based unification algorithm [5] and Robinson's original one [12]. We claim that, as in the unification case, this difference makes the method more flexible, easier to describe and comprehend, and thus also easier to generalize. This claim is supported by the fact that the approach is not restricted to the disjoint signature case: the theories to be combined are allowed to share a finite set of constant symbols.

There is a persistent rumor that combining decision procedures for the word problem (in the disjoint case) is a special case of Nelson and Oppen's combination method [7]. At first sight, the idea is persuasive: the Nelson-Oppen method combines decision procedures for the validity of quantifier-free formulae in first-order theories, and the word problem is concerned with the validity of quantifier-free formulae of the form $s \equiv t$ in equational theories. Considered more closely, the rumor turns out to be not quite true for two reasons. First, Nelson and Oppen require the single theories to be stably infinite, and equational theories need not satisfy this property.[2] Second, although we are only interested in the word problem for the combined theory, Nelson and Oppen's method generates more general validity problems in the single theories. Thus, just knowing that the word problems in the single theories are decidable is not sufficient. We shall show, however, that our method for combining decision procedures for the word problem follows an approach very similar to Nelson and Oppen's.

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

[2] It turns out, however, that they satisfy a somewhat weaker property, which in principle suffices to apply Nelson and Oppen's combination approach.

2 Nelson and Oppen's Combination Method

We shall first recall the general procedure, and then investigate whether it can be applied to equational theories.

The General Method

This method is concerned with combining decision procedures for the validity of quantifier-free formulae. Let Γ be a first-order theory over the signature Σ, which consists of a set Σ_F of function symbols and a set Σ_P of relation symbols. We treat equality $\equiv$ as a logical symbol, i.e., it is always present and thus needs not be included in the signature. A *quantifier-free formula* is a Boolean combination of Σ-atoms, i.e., of formulae of the form $P(s_1, \ldots, s_n)$, where $P \in \Sigma_P \cup \{\equiv\}$ is an n-ary predicate symbol and $s_1, \ldots, s_n \in T(\Sigma_F, V)$ are Σ_F-terms with variables from a (countably infinite) set of variables V. As usual, we say that a quantifier-free formula φ is *valid* in Γ iff it holds in all models of Γ, i.e., iff for all Σ-structures $\mathcal{A}$ that satisfy Γ and all valuations α of the variables in φ by elements of $\mathcal{A}$ we have $\mathcal{A}, \alpha \models \varphi$. Since a formula is valid in Γ iff its negation is unsatisfiable in Γ, we can turn the validity problem for Γ into an equivalent *satisfiability problem:* we know that a formula φ is not valid in Γ iff there exist a Σ-model $\mathcal{A}$ of Γ and a valuation α such that $\mathcal{A}, \alpha \models \neg\varphi$.

When considering the satisfiability problem, we may (without loss of generality) restrict our attention to *conjunctive* quantifier-free formulae, i.e., conjunctions of Σ-atoms and negated Σ-atoms. In fact, a given quantifier-free formula can be transformed into an equivalent formula in disjunctive normal form (i.e., a disjunction of conjunctive quantifier-free formulae), and this disjunction is satisfiable in Γ iff one of its disjuncts is satisfiable in Γ.

Now assume that Σ_1 and Σ_2 are two disjoint signatures and that Γ is obtained as the union of a Σ_1-theory Γ_1 and a Σ_2-theory Γ_2. How can decision procedures for validity (equivalently: satisfiability) in Γ_i $(i = 1, 2)$ be used to obtain a decision procedure for validity (equivalently: satisfiability) in Γ? *Nelson and Oppen's method for combining decision procedures* considers the satisfiability problem in Γ. Given a conjunctive quantifier-free $(\Sigma_1 \cup \Sigma_2)$-formula φ to be tested for satisfiability, it proceeds in three steps:

1. *Generate a conjunction $\varphi_1 \wedge \varphi_2$ that is equivalent to φ, where φ_i is a pure Σ_i-formula $(i = 1, 2)$.*
   Here equivalent means that φ and $\varphi_1 \wedge \varphi_2$ are satisfiable in exactly the same models of Γ. This is achieved by replacing alien subterms by variables and adding appropriate equations (see the example below).

2. *Test the pure formulae for satisfiability in the respective theories.*
   If φ_i is unsatisfiable in Γ_i for $i = 1$ or $i = 2$, then return "unsatisfiable." Otherwise proceed with the next step.

3. *Propagate equalities between different shared variables (i.e., variables* $u_j \neq v_j$ *occurring in both* φ_1 *and* φ_2*), if a disjunction of such equalities can be deduced from the pure parts.*
   A disjunction $u_1 \equiv v_1 \vee \ldots \vee u_k \equiv v_k$ of equations between different shared variables can be deduced from φ_i in Γ_i iff $\varphi_i \wedge u_1 \not\equiv v_1 \wedge \ldots \wedge u_k \not\equiv v_k$ is unsatisfiable in Γ_i. Since the satisfiability problem in Γ_i was assumed to be decidable, and since there are only finitely many shared variables, it is decidable whether such a disjunction exists.
   If no such disjunctions can be deduced, return "satisfiable." Otherwise, take any of them,[3] and propagate its equations as follows. For every disjunct $u_j \equiv v_j$, proceed with the second step for the formula $\varphi_1\sigma_j \wedge \varphi_2\sigma_j$, where $\sigma_j := \{u_j \mapsto v_j\}$ (for $j = 1, \ldots, k$). Return "satisfiable" iff one of these cases yields "satisfiable."

Example 2.1 Consider the (equational) theories $\Gamma_1 := \{\forall x. f(x, x) \equiv x\}$ and $\Gamma_2 := \{\forall x. g(g(x)) \equiv g(x)\}$ over the signatures $\Sigma_1 := \{f\}$ and $\Sigma_2 := \{g\}$. If we want to know whether the (mixed) quantifier-free formula

$$g(f(g(z), g(g(z)))) \equiv g(z)$$

is valid in $\Gamma_1 \cup \Gamma_2$, we can apply the Nelson-Oppen procedure to its negation $g(f(g(z), g(g(z)))) \not\equiv g(z)$.

In *Step 1*, $f(g(z), g(g(z)))$ is an alien subterm in $g(f(g(z), g(g(z))))$ (since $g \in \Sigma_2$ and $f \in \Sigma_1$). In addition, $g(z)$ and $g(g(z))$ are alien subterms in $f(g(z), g(g(z)))$. Replacing these subterms by variables yields the conjunction $\varphi_1 \wedge \varphi_2$, where

$$\varphi_1 := u \equiv f(v, w) \quad \text{and} \quad \varphi_2 := g(u) \not\equiv g(z) \wedge v \equiv g(z) \wedge w \equiv g(g(z)).$$

In *Step 2*, it is easy to see that both pure formulae are satisfiable in their respective theories. The equation $u \equiv f(v, w)$ is obviously satisfiable in the trivial model of Γ_1 (of cardinality 1). The formula φ_2 is, for example, satisfiable in the Γ_2-free algebra with two generators, where u is interpreted by one generator, z by the other, and v, w as required by the equations.

In *Step 3*, we can deduce $w \equiv v$ from φ_2 in Γ_2 since φ_2 contains $v \equiv g(z) \wedge w \equiv g(g(z))$ and Γ_2 contains the universally quantified equation $g(g(x)) \equiv g(x)$. Propagating the equality $w \equiv v$ yields the pure formulae

$$\varphi_1' := u \equiv f(v, v) \quad \text{and} \quad \varphi_2' := g(u) \not\equiv g(z) \wedge v \equiv g(z) \wedge v \equiv g(g(z)),$$

which again turn out to be separately satisfiable in *Step 2* (with the same models as used above).

In *Step 3*, we can now deduce the equality $u \equiv v$ from φ_1' in Γ_1, and its propagation yields

$$\varphi_1'' := v \equiv f(v, v) \quad \text{and} \quad \varphi_2'' := g(v) \not\equiv g(z) \wedge v \equiv g(z) \wedge v \equiv g(g(z)).$$

[3] For efficiency reasons, one should take a disjunction with minimal k.

In *Step 2*, it turns out that φ_2'' is not satisfiable in Γ_2, and thus the answer is "unsatisfiable," which shows that $g(f(g(z), g(g(z)))) \equiv g(z)$ is valid in $\Gamma_1 \cup \Gamma_2$. In fact, $v \equiv g(z)$ and the equation $g(g(x)) \equiv g(x)$ of Γ_2 imply that $g(v) \equiv g(z)$, which contradicts $g(v) \not\equiv g(z)$.

Obviously, the procedure terminates since there are only finitely many shared variables to be identified. In addition, it is easy to see that satisfiability is preserved at each step. This implies that the procedure is complete, that is, if it answers "unsatisfiable" (because variable propagation has made one of the pure subformulae unsatisfiable in the corresponding component theory), then the original formula is in fact unsatisfiable.

For arbitrary theories Γ_1 and Γ_2, the combination procedure need not be sound (see Example 2.3 below). One must assume that each Γ_i is *stably infinite*, that is, such that a quantifier-free formula φ_i is satisfiable in Γ_i iff it is satisfiable in an infinite model of Γ_i. This restriction was not mentioned in Nelson and Oppen's original article [7]; it was introduced in [10]. Two new and simple proofs of soundness and completeness of the procedure are given in [11, 15]. In essence, both depend on the following proposition (see [15] for a proof). For a finite set of variables X, let $\Delta(X)$ denote the conjunction of all disequations $x \not\equiv y$ for $x, y \in X, x \neq y$.

Proposition 2.2 *Let Γ_1 and Γ_2 be two stably infinite theories over the disjoint signatures Σ_1 and Σ_2, respectively; let φ_i be a quantifier-free Σ_i-formula ($i = 1, 2$), and let X be the set of variables occurring in both φ_1 and φ_2. If $\varphi_i \wedge \Delta(X)$ is satisfiable in Γ_i for $i = 1, 2$, then $\varphi_1 \wedge \varphi_2$ is satisfiable in $\Gamma_1 \cup \Gamma_2$.*

It is easy to see that this proposition yields soundness of the procedure, that is, if the procedure answers "satisfiable" then the original formula was satisfiable. In fact, if in Step 3 no disjunction of equalities between shared variables can be derived from the pure formulae, the prerequisite for the proposition is satisfied (take the disjunction of all $x \not\equiv y$ for $x, y \in X, x \neq y$). We shall use a similar proposition to prove the correctness of our combination procedure.

Its Application to Equational Theories

We now turn to the question of whether the Nelson-Oppen method applies to equational theories (that is, sets of universally quantified equations). For this purpose, we will consider only functional signatures, which means that the only predicate symbol in our formulae will be the equality symbol.

First, note that a trivial equational theory E (i.e., a theory that has only the trivial 1-element model, or equivalently a theory that entails $x \equiv y$ for distinct variables x, y) cannot be stably infinite. However, this is not a real problem since satisfiability and validity in the trivial model are obviously decidable. In addition, if E_1 or E_2 are trivial, then their union is trivial, and thus one does not need a combination procedure to decide satisfiability in

$E_1 \cup E_2$. The next example shows that non-trivial equational theories need not be stably infinite either and that Nelson and Oppen's procedure is not correct for such theories.

Example 2.3 Let $E_1 := \{\forall x, y. f(g(x), g(y)) \equiv x, \forall x, y. f(g(x), h(y)) \equiv y\}$. It is easy to see that E_1 is non-trivial. In fact, by orienting the equations from left to right, one obtains a canonical term rewriting system, in which any two distinct variables have a different normal form. Now, consider the quantifier-free formula $g(x) \equiv h(x)$. Obviously, this formula is satisfiable in the trivial model of E_1. In every model $\mathcal{A}$ of E_1 that satisfies $g(x) \equiv h(x)$, there exists an element e such that $g^{\mathcal{A}}(e) = h^{\mathcal{A}}(e)$. But then we have that

$$a = f^{\mathcal{A}}(g^{\mathcal{A}}(a), g^{\mathcal{A}}(e)) = f^{\mathcal{A}}(g^{\mathcal{A}}(a), h^{\mathcal{A}}(e)) = e$$

for every element a of $\mathcal{A}$, which entails that $\mathcal{A}$ is the trivial model. Thus, $g(x) \equiv h(x)$ is only satisfiable in the trivial model of E_1, which show that the (non-trivial) equational theory E_1 is not stably infinite.

To see that this really leads to an incorrect behavior of the Nelson-Oppen method, let $E_2 := \{\forall x. k(x) \equiv k(x)\}$, and consider the conjunction $g(x) \equiv h(x) \wedge k(x) \not\equiv x$. Clearly, $k(x) \not\equiv x$ is satisfiable in E_2 (for instance, in the E_2-free algebra with 1 generator) and, as we saw earlier, $g(x) \equiv h(x)$ is satisfiable in E_1. In addition, no equations between distinct shared variables can be deduced (since there is only one shared variable). It follows that Nelson and Oppen's procedure would answer "satisfiable" on input $g(x) \equiv h(x) \wedge k(x) \not\equiv x$. However, since $g(x) \equiv h(x)$ is only satisfiable in the trivial model of E_1, and no disequation can be satisfied in a trivial model, $g(x) \equiv h(x) \wedge k(x) \not\equiv x$ is unsatisfiable in $E_1 \cup E_2$.

The problem pointed out by the example is solely due to the fact that one of the pure subformulae is only satisfiable in the trivial model, whereas the other is not satisfiable in the trivial model. Given a quantifier-free formula φ, it is obviously decidable whether φ is satisfiable in the trivial model of $E_1 \cup E_2$: just replace all equations by "true" and all negated equations by "false." To test satisfiability in a non-trivial model of $E_1 \cup E_2$, one can then consider satisfiability in $E_1' \cup E_2'$, where $E_i' := E_i \cup \{\exists x, y. x \not\equiv y\}$. Obviously, φ is satisfiable in $E_1 \cup E_2$ iff it is satisfiable in the trivial model or in a model of $E_1' \cup E_2'$. We shall show in the following:

Lemma 2.4 *Let E_i be a non-trivial equational theory.*

1. *The theory $E_i' := E_i \cup \{\exists x, y. x \not\equiv y\}$ is stably infinite.*
2. *If satisfiability of quantifier-free formulae is decidable for E_i, then it is also decidable for E_i'.*

Consequently, Nelson and Oppen's procedure can be applied to the combined theory $E_1' \cup E_2'$. This shows

Theorem 2.5 *Let E_1 and E_2 be two equational theories over disjoint signatures. If satisfiability of quantifier-free formulae is decidable for E_i ($i = 1, 2$), then it is also decidable for $E_1 \cup E_2$.*

The *second statement of the lemma* is trivial. In fact, let φ_i be a quantifier-free Σ_i-formula. Then φ_i is satisfiable in E_i' iff $\varphi_i \wedge x \not\equiv y$ is satisfiable in E_i, where x, y are two distinct variables not occurring in φ_i.

The *first statement of the lemma* is an easy consequence of the fact that the class of models of an equational theory is closed under direct products. In fact, assume that the quantifier-free formula φ_i is satisfiable in E_i', i.e., φ_i is satisfiable in a non-trivial model $\mathcal{A}$ of E_i. Then the countably infinite direct product of $\mathcal{A}$ with itself is an infinite model of E_i (and of E_i'), and it is easy to see that it satisfies φ_i.

The Word Problem

For an equational theory E, the word problem is concerned with the validity in E of quantifier-free formulae of the form $s \equiv t$.[4] Equivalently, the word problem asks for the (un)satisfiability of $s \not\equiv t$ in E. Now, let E_1 and E_2 be two equational theories over disjoint signatures. Obviously, Theorem 2.5 implies that the word problem is decidable for $E_1 \cup E_2$, provided that the validity (equivalently: satisfiability) of (arbitrary) quantifier-free formulae is decidable for E_1 and E_2.

However, if we just assume that the word problem (equivalently: satisfiability of formulae of the form $s \not\equiv t$) is decidable for E_i ($i = 1, 2$), then such an assumption will be too weak to allow for a straightforward application of the Nelson-Oppen procedure. In fact, it is easy to see that the satisfiability tests in the second and third step of the procedure need not be of the specific form that can be handled by a procedure for the word problem.

In our method for combining decision procedures for the word problem, the main ideas to overcome these difficulties are in principle[5] the following:

- In Step 3, propagate only equalities that can be deduced with the help of a decision procedure for the word problem in E_i:
  - If we have $x \equiv s, y \equiv t$ and $s =_{E_i} t$, then propagate $x \equiv y$.
  - If we have $x \equiv s$, y occurs in s, and $s =_{E_i} y$, then propagate $x \equiv y$.
- In Step 2, return "unsatisfiable" only if equality propagation has generated a trivially unsatisfiable disequation of the form $x \not\equiv x$.

The main part of the proof of correctness will be to show that this restricted form of equality propagation and satisfiability test is sufficient for our purposes.

[4] As usual, we often write "$s =_E t$" to express that the formula $s \equiv t$ is valid in E.

[5] The rules of our combination approach are somewhat more complex for technical reasons, and the fact that we also take shared constants into account.

3 The Combination Procedure for the Word Problem

In the following, we consider the equational theory $E := E_1 \cup E_2$ where, for $i = 1, 2$, E_i is a non-trivial equational theory over the functional signature Σ_i. Furthermore, we assume that $\Sigma := \Sigma_1 \cap \Sigma_2$ is a finite (possibly empty) set of constant symbols, and that the word problem for each E_i is decidable.

To decide the word problem for E, we consider the satisfiability problem for quantifier-free formulae of the form $s_0 \not\equiv t_0$, where s_0 and t_0 are $(\Sigma_1 \cup \Sigma_2)$-terms. As in the Nelson-Oppen procedure, the first step of our procedure transforms this formula into a conjunction of pure formulae by means of variable abstraction.

Abstraction Systems

In the following, we use finite sets of atomic formulae in place of conjunctions of such formulae. We will then say that such a set is satisfiable in a theory iff the conjunction of its elements is satisfiable in that theory.

To define the abstraction process in more detail, we need some notation. We assume that all terms are built over the signature $\Sigma_1 \cup \Sigma_2$ with variables from a countably infinite set V. The elements of Σ_1 will be called *1-symbols* and the elements of Σ_2 *2-symbols*. A term t is called *i-term* iff it 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 shared constants are both 1- and 2-terms. A subterm s of a 1-term t is called *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. An i-term s is *pure* iff it contains only i-symbols and variables.

For the disequation $s_0 \not\equiv t_0$, the *abstraction procedure* starts with the set $S_0 := \{x \not\equiv y, x \equiv s_0, y \equiv t_0\}$, where x, y are distinct variables not occurring in s_0, t_0, if s_0 and t_0 are not variables. If s_0 (t_0) is a variable, we use s_0 in place of x (t_0 in place of y), and omit the corresponding (trivial) equation. Assume that a finite set S_i consisting of $x \not\equiv y$ and equations of the form $u \equiv s$ (where $u \in V$ and $s \in T(\Sigma_1 \cup \Sigma_2, V) \setminus V$) has already been constructed. If S_i contains an equation $u \equiv s$ such that s has an alien subterm t at position p, then S_{i+1} is obtained from S_i by replacing $u \equiv s$ by the equations $u \equiv s'$ and $v \equiv t$, where v is a variable not occurring in S_i, and s' is obtained from s by replacing t at position p by v. Otherwise, if all terms occurring in S_i are pure, then S_i is the output of the abstraction procedure. Obviously, this process terminates and yields a set $AS(s_0 \not\equiv t_0)$ which is satisfiable in E iff $s_0 \not\equiv t_0$ is satisfiable in E.

The set $AS(s_0 \not\equiv t_0)$ satisfies additional properties, whose importance will become clear later on.

Definition 3.1 *Let S be a set of equations of the form $(v \equiv t)$ where $v \in V$ and $t \in T(\Sigma_1 \cup \Sigma_2, V) \setminus V$. We define the relation $\prec$ on S as follows. For* *all* $(u \equiv s), (v \equiv t) \in S$,

$$(u \equiv s) \prec (v \equiv t) \quad \textit{iff} \quad v \in \mathcal{V}(s),$$

where $\mathcal{V}(s)$ *denotes the set of variables occurring in* s*. By* $\prec^+$ *we denote the transitive and by* $\prec^*$ *the reflexive-transitive closure of* $\prec$*. The relation* $\prec$ *is called* acyclic *iff* $\prec^+$ *is irreflexive.*

Definition 3.2 (Abstraction System) *We say that the set* $A := \{x \not\equiv y\} \cup S$ *is an* abstraction system with initial formula $x \not\equiv y$ *iff the following holds:*

1. S *is a finite subset of* $\{v \equiv t \mid v \in V,\ t \in (T(\Sigma_1, V) \cup T(\Sigma_2, V)) \setminus V\}$*;*
2. *the relation* $\prec$ *on* S *is acyclic;*
3. *for all* $(u \equiv s), (v \equiv t) \in S$,
   (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* $t \notin T(\Sigma_i, V)$*.*

Condition (1) states that S consists of equations between variables and pure non-variable terms; condition (2) implies that for all $(u \equiv s), (v \equiv t) \in S$, if $(u \equiv s) \prec^* (v \equiv t)$ then $u \notin \mathcal{V}(t)$; condition (3a) implies that a variable appears only once in S as the left-hand side of an equation; condition (3b) implies that the elements of every $\prec$-chain of S have alternating signatures. The following proposition is an easy consequence of the definition of the abstraction procedure.

Proposition 3.3 *The set* $AS(s_0 \not\equiv t_0)$ *obtained by applying the abstraction procedure to the disequation* $s_0 \not\equiv t_0$ *is an abstraction system which is satisfiable in* E *iff* $s_0 \not\equiv t_0$ *is satisfiable in* E*.*

The Combination Procedure

The main idea of the procedure, which is described in Fig. 1, is to see whether the disequation between the two input terms is satisfiable in E by turning the disequation into an abstraction system, and then propagating some equations between variables which are deduced using the decision procedures for the single theories. The transformations the initial system goes through will eventually produce an abstraction system whose initial formula has the form $(v \not\equiv v)$ iff the two input terms are in $=_E$ (i.e., iff the corresponding disequation is unsatisfiable in E).

During the execution of the procedure on input $\{s_0, t_0\}$, S, the set on which the procedure works, is repeatedly modified by the application of the transformations at step (2). In essence, transformations (2a) and (2b) remove possible collapse equations of E_1 or E_2 from S. In these transformations we have used the notation "$s_1[v_2]$" to express that the variable v_2 occurs in the term s_1. Transformation (2c) instantiates with a shared constant any variable equated to a Σ_i-term that is equivalent to that constant and discards the corresponding equation. Transformation (2d) identifies any two variables equated to equivalent Σ_i-terms and then discards one of the corresponding equations. Transformation (2e) recognizes an inconsistency in S and replaces it with (a symbol for) the unsatisfiable set.

Input: $\{s_0, t_0\} \subset T(\Sigma_1 \cup \Sigma_2, V)$.

1. Let $S := AS(s_0 \not\equiv t_0)$.
2. Repeatedly apply the following transformations to S until none of them is applicable.
   (a) $\{u \not\equiv v\} \cup \{v_1 \equiv s_1[v_2]\} \cup \{v_2 \equiv s_2\} \cup T$
   $\longrightarrow \{u \not\equiv v\}\{v_1 \mapsto v_2\} \cup \{v_2 \equiv s_2\} \cup T\{v_1 \mapsto s_2\}$
   if $v_2 =_{E_i} s_1$ for $i = 1$ or $i = 2$.
   (b) $\{v_1 \equiv s_1[v_2]\} \cup T \longrightarrow T\{v_1 \mapsto v_2\}$
   if for no s_2, $\{v_2 \equiv s_2\} \subseteq T$
   and
   $v_2 =_{E_i} s_1$ for $i = 1$ or $i = 2$.
   (c) $\{u \not\equiv v\} \cup \{x \equiv s\} \cup T \longrightarrow \{u \not\equiv v\} \cup \{x \equiv c\} \cup T\{x \mapsto c\}$
   if $c \in \Sigma$,
   $s \notin \Sigma$,
   and
   $c =_{E_i} s$ for $i = 1$ or $i = 2$.
   (d) $\{x \equiv s\} \cup \{y \equiv t\} \cup T \longrightarrow \{y \equiv t\} \cup T\{x \mapsto y\}$
   if $(y \equiv t) \not\prec^* (x \equiv s)$
   and
   $s =_{E_i} t$ for $i = 1$ or $i = 2$.
   (e) $\{v \not\equiv v\} \cup T \longrightarrow \bot$.
3. Succeed if $\bot$ has been generated. Fail otherwise.

Figure 1: The Combination Procedure.

We prove in Section 5 that this combination procedure decides the word problem for $E = E_1 \cup E_2$ by showing that the procedure is partially correct (i.e., sound and complete) and terminates on all inputs. For the proof of correctness, we need to extend the result in Proposition 2.2 to the case of signatures sharing constants. We will do this in the next section.

4 Satisfiability in the Union of Theories Sharing Constants

The results of this section concern arbitrary first-order theories. They are not restricted to equational theories.

Let $\Sigma' \subseteq \Sigma$ be signatures, and $\mathcal{A}$ be a Σ-structure. The Σ'-reduct of $\mathcal{A}$, denoted by $\mathcal{A}^{\Sigma'}$, is the Σ'-structure obtained from $\mathcal{A}$ by just forgetting about the interpretation of the symbols in $\Sigma \setminus \Sigma'$. In this situation, $\mathcal{A}$ is called an expansion of $\mathcal{A}^{\Sigma'}$ to the larger signature.

Lemma 4.1 *Let Γ_1 be a Σ_1-theory, and Γ_2 be a Σ_2-theory. Their union $\Gamma_1 \cup \Gamma_2$ is consistent iff there is a model $\mathcal{A}_1$ of Γ_1 and a model $\mathcal{A}_2$ of Γ_2 such that their reducts to $\Sigma_1 \cap \Sigma_2$ are isomorphic.*

A proof of this lemma can be obtained as an easy consequence of Craig's interpolation theorem [15], or by a direct model-theoretic construction [11].

Let X, Y be finite sets of variables or constants. We denote by $\Delta(X)$ the conjunction of all the disequations $x \not\equiv y$ for $x, y \in X, x \neq y$, and by $\Delta(X, Y)$ the conjunction of all the disequations $x \not\equiv y$ for $x \in X, y \in Y$.

Proposition 4.2 *Let Γ_1 be a Σ_1-theory and Γ_2 be a Σ_2-theory, and assume that $\Sigma = \Sigma_1 \cap \Sigma_2$ is a finite set of constant symbols. For $i = 1, 2$, let $\mathcal{A}_i$ be a model of Γ_i, φ_i a quantifier-free Σ_i-formula and X the set of variables occurring in both φ_1 and φ_2. If $\mathcal{A}_1$ and $\mathcal{A}_2$ have the same cardinality and, for $i = 1, 2$,*

$$\varphi_i \wedge \Delta(X) \wedge \Delta(\Sigma) \wedge \Delta(X, \Sigma)$$

is satisfiable in $\mathcal{A}_i$, then $\varphi_1 \wedge \varphi_2$ is satisfiable in $\Gamma_1 \cup \Gamma_2$.

Proof. For $i = 1, 2$, let X_i be the set of variables occurring in φ_i, and let $X = X_1 \cap X_2 = \{v_1, \ldots, v_n\}$. Let α_i be a valuation of the variables in X_i by elements of $\mathcal{A}_i$ such that

$$\mathcal{A}_i, \alpha_i \models \varphi_i \wedge \Delta(X) \wedge \Delta(\Sigma) \wedge \Delta(X, \Sigma).$$

Then choose a set $C := \{c_1, \ldots, c_n\}$ of pairwise distinct constant symbols not appearing in $\Sigma_1 \cup \Sigma_2$, let $\rho := \{v_1 \mapsto c_1, \ldots, v_n \mapsto c_n\}$ and, for $i = 1, 2$, consider the $(\Sigma_i \cup C)$-theory

$$\widehat{\Gamma}_i \quad := \quad \Gamma_i \cup \{\tilde{\exists}\,(\varphi_i \rho)\} \cup \{\Delta(C), \Delta(\Sigma), \Delta(C, \Sigma)\},$$

where $\tilde{\exists}\,(\varphi_i \rho)$ denotes the existential closure of the formula $\varphi_i \rho$.

Obviously, $\mathcal{A}_i$ can be expanded to a model $\mathcal{B}_i$ of $\widehat{\Gamma}_i$ by interpreting each c_j in C as $\alpha_i(v_j)$. For $i = 1, 2$, let $B'_i := \{c^{\mathcal{B}_i} \mid c \in \Sigma \cup C\}$, where $c^{\mathcal{B}_i}$ is the interpretation of the constant c in the structure $\mathcal{B}_i$. Because $\mathcal{B}_i$ satisfies $\Delta(C) \wedge \Delta(\Sigma) \wedge \Delta(C, \Sigma)$, we have $c^{\mathcal{B}_i} \neq d^{\mathcal{B}_i}$ for all $c, d \in \Sigma \cup C, c \neq d$. Thus, we can define a mapping $h_0 : B_1' \to B_2'$ by $h(c^{\mathcal{B}_1}) := c^{\mathcal{B}_2}$ for all $c \in \Sigma \cup C$, which is obviously a bijection. Since the domain of $\mathcal{B}_i$ coincides with the domain of $\mathcal{A}_i$, the structures $\mathcal{B}_1$ and $\mathcal{B}_2$ have the same cardinality, which implies that h_0 can be extended to a bijection h between the domains of $\mathcal{B}_1$ and $\mathcal{B}_2$. It is immediate by the construction of h_0 that h is an isomorphism of ${\mathcal{B}_1}^{\Sigma \cup C}$ onto ${\mathcal{B}_2}^{\Sigma \cup C}$.

It follows from Proposition 4.1 that $\widehat{\Gamma}_1 \cup \widehat{\Gamma}_2$ is consistent. Because the formulae $\varphi_1\rho$ and $\varphi_2\rho$ have no variables in common, $\widehat{\Gamma}_1 \cup \widehat{\Gamma}_2$ is equivalent to

$$\Gamma_1 \cup \Gamma_2 \cup \{\tilde{\exists}\,(\varphi_1 \wedge \varphi_2)\rho\} \cup \{\Delta(C), \Delta(\Sigma), \Delta(C, \Sigma)\}.$$

Consequently, $(\varphi_1 \wedge \varphi_2)\rho$, and therefore $\varphi_1 \wedge \varphi_2$, is satisfiable in $\Gamma_1 \cup \Gamma_2$. $\square$

5 The Correctness Proof

In the following, we denote by $S^{(0)}$ the abstraction system $AS(s_0 \not\equiv t_0)$ obtained by applying the abstraction procedure to the input disequation, and by $S^{(j)}$ $(j \geq 1)$ the set S after the j^{th} transformation step.

Lemma 5.1 (Termination) *The combination procedure halts on all inputs.*

Proof. As mentioned above, the abstraction procedure applied in the first part of the combination procedure terminates. In addition, every single transformation step in the second part is executable in finite time. All we need to show then is that the procedure performs only finitely many of these transformations. For $j \geq 0$, let n_j be the number of variables occurring on the left-hand side of an equation in $S^{(j)}$ plus the number of non-shared terms on the right-hand side of an equation in $S^{(j)}$. It is easy to see that $n_0 > n_1 > n_2 \ldots$, and thus the number of transformation steps is bounded by n_0. $\square$

Next, we want to show that all sets $S^{(j)} \neq \bot$ obtained during the run of the combination procedure are in fact abstraction systems. The proof of acyclicity (Condition 2 in Definition 3.2) will be facilitated by the following lemma, whose simple proof is omitted.

Lemma 5.2 *Let $<$ be a binary relation on a finite set A, and $b, c \in A$ be such that $c \not<^* b$. We denote the restriction of $<$ to $A \setminus \{b\}$ by $<_b$,*[6] *and consider the relations*

$$\begin{aligned} <_1 &:= <_b \cup \{\langle a, d\rangle \mid a < b, c < d\} \\ <_2 &:= <_b \cup \{\langle a, c\rangle \mid a < b\}. \end{aligned}$$

If $<$ is acyclic, then $<_1$ and $<_2$ are acyclic as well.

[6] That is $<_b := < \cap\, (A \setminus \{b\})^2$.

Lemma 5.3 *Given an execution of the combination procedure, let $S^{(n)}$ be the last set if the procedure fails, or the last but one, if the procedure succeeds. Then, for all $j = 0, \dots, n$, $S^{(j)}$ is an abstraction system.*

Proof. For all $j = 0, \dots, n$, let $\prec_j$ denote the relation $\prec$ (cf. Def. 3.1) on $S^{(j)}$. We prove the claim by induction on j. The induction base ($j = 0$) is trivial by construction of $S^{(0)}$ and Proposition 3.3.

Thus, assume that $0 < j \leq n$, and that we already know that $S^{(j-1)}$ is an abstraction system. Observing that, by the assumption on n, $S^{(j)}$ cannot be the result of transformation (2e), consider the following cases.

Transformation (2a). From the definition of transformation (2a), it is easy to see that $S^{(j-1)}$ and $S^{(j)}$ must have the following form:

$$\begin{array}{lllllllll} S^{(j-1)} & = & \{u \not\equiv v\} & \cup & \{v_1 \equiv s_1[v_2]\} & \cup & \{v_2 \equiv s_2\} & \cup & T \\ S^{(j)} & = & \{u \not\equiv v\}\{v_1 \mapsto v_2\} & \cup & & & \{v_2 \equiv s_2\} & \cup & T\{v_1 \mapsto s_2\} \end{array}$$

Let $\{u' \not\equiv v'\} := \{u \not\equiv v\}\{v_1 \mapsto v_2\}$. We show that $S^{(j)}$ is an abstraction system with initial formula $u' \not\equiv v'$.

If we take $\prec_{j-1}$ to be the relation $<$ of Lemma 5.2, $(v_1 \equiv s_1)$ to be b, and $(v_2 \equiv s_2)$ to be c, it is easy to see that $\prec_j$ coincides with $<_1$ (as defined in the lemma). Recalling that $\prec_{j-1}$ is acyclic by induction, it follows that $\prec_j$ is acyclic as well, and thus condition (2) of Definition 3.2 holds.

Since $\{v_1 \mapsto s_2\}$ does not change the left-hand sides of equations in T, it is immediate that condition (3a) of Definition 3.2 holds as well.

Finally, observe that v_1 can appear in T only in an equation of the form $(v_0 \equiv s_0[v_1])$ and that $(v_0 \equiv s_0) \prec_{j-1} (v_1 \equiv s_1) \prec_{j-1} (v_2 \equiv s_2)$. By induction, we know that s_0 and s_2 are both non-variable Σ_i-terms, for $i = 1$ or $i = 2$; therefore, the application of $\{v_1 \mapsto s_2\}$ does not change the signature of the equations in T. It follows that $S^{(j)}$ satisfies both conditions (1) and (3b) of Definition 3.2.

Transformation (2b). The proof is essentially a special case of the one above, with s_2 replaced by v_2.

Transformation (2c). We know that $S^{(j-1)}$ and $S^{(j)}$ have the following form:

$$\begin{array}{lllllll} S^{(j-1)} & = & \{u \not\equiv v\} & \cup & \{x \equiv s\} & \cup & T \\ S^{(j)} & = & \{u \not\equiv v\} & \cup & \{x \equiv c\} & \cup & T\{x \mapsto c\} \end{array}$$

It is obvious that $\{x \equiv c\} \cup T\{x \mapsto c\}$ satisfies conditions (2) and (3a) of Definition 3.2. To see that it also satisfies conditions (1) and (3b), observe that the substitution $\{x \mapsto c\}$ does not change the signature of the terms in T, and that x does not appear in $T\{x \mapsto c\}$. It follows that $S^{(j)}$ is an abstraction system with initial formula $u \not\equiv v$.

Transformation (2d). We know that $S^{(j-1)}$ and $S^{(j)}$ have the form

$$\begin{array}{lllllll} S^{(j-1)} & = & (T \cup \{u \not\equiv v\}) & \cup & \{x \equiv s\} & \cup & \{y \equiv t\} \\ S^{(j)} & = & (T \cup \{u \not\equiv v\})\{x \mapsto y\} & \cup & & & \{y \equiv t\}, \end{array}$$

and it is *not* the case that $(y \equiv t) \prec_{j-1}{}^* (x \equiv s)$.

It is not difficult to see that this time $\prec_j$ is derivable from $\prec_{j-1}$ in the same way $<_2$ is derivable from $<$ in Lemma 5.2, where $(x \equiv s)$ is b and $(y \equiv t)$ is c. Again, it follows that $\prec_j$ satisfies condition (2) of Definition 3.2. By induction, we know that y appears as the left-hand side of no equations in $S^{(j-1)} \setminus \{y \equiv t\}$, and so it is immediate that $S^{(j)}$ satisfies condition (3a). It is also immediate that $S^{(j)}$ satisfies conditions (1) and (3b). In conclusion, $S^{(j)}$ is an abstraction system with initial formula $(u \not\equiv v)\{x \mapsto y\}$. □

The next lemma shows that the transformation rules preserve satisfiability.

Lemma 5.4 *For all $j > 0$ and all models $\mathcal{A}$ of $E = E_1 \cup E_2$, $S^{(j)}$ is satisfiable in $\mathcal{A}$ iff $S^{(j-1)}$ is satisfiable in $\mathcal{A}$.*

Proof. Transformation (2a). We know that $S^{(j-1)}$ and $S^{(j)}$ have the form

$$\begin{array}{lllllllll} S^{(j-1)} & = & T & \cup & \{u \not\equiv v\} & \cup & \{v_1 \equiv s_1[v_2]\} & \cup & \{v_2 \equiv s_2\} \\ S^{(j)} & = & T\{v_1 \mapsto s_2\} & \cup & \{u \not\equiv v\}\{v_1 \mapsto v_2\} & \cup & & & \{v_2 \equiv s_2\} \end{array}$$

and that $v_2 =_{E_i} s_1$ for $i = 1$ or $i = 2$. Assume that a valuation α satisfies $S^{(j-1)}$ in a model $\mathcal{A}$ of E. Since $v_2 \equiv s_1[v_2]$ is valid in E, for being valid in E_i, α must assign both v_1 and v_2 with the individual denoted by s_1. It follows immediately that α satisfies $S^{(j)}$ in $\mathcal{A}$.

Now, assume that a valuation α satisfies $S^{(j)}$ in a model $\mathcal{A}$ of E. Observe that, since $S^{(j-1)}$ is an abstraction system, v_1 does not occur in $v_2 \equiv s_2$ and, as a consequence, it does not occur in $S^{(j)}$ at all. Let α' be an extension of α such that $\alpha(v_1) = \alpha(v_2)$. It is immediate that α' satisfies the set $S_1 := T \cup \{v_1 \equiv s_2\} \cup \{u \not\equiv v\} \cup \{v_1 \equiv v_2\} \cup \{v_2 \equiv s_2\}$ in $\mathcal{A}$. Since $v_2 \equiv s_1[v_2]$ is valid in $\mathcal{A}$, it is also immediate that α' satisfies the set $S_2 := \{v_1 \equiv s_1[v_2]\}$ in $\mathcal{A}$. It follows that α' satisfies $S^{(j-1)}$, which is a subset of $S_1 \cup S_2$.

Transformation (2b) is a special case of the one above with s_2 replaced by v_2, *transformations (2c)* and *(2d)* can be treated similarly, and *transformation (2e)* is trivial (since the equation $v \not\equiv v$ is obviously unsatisfiable). □

It is now easy to show that the combination procedure is sound.

Proposition 5.5 *If the combination procedure succeeds on an input $\{s_0, t_0\}$, then $s_0 =_E t_0$.*

Proof. By the procedure's definition, we know that, if the procedure succeeds, there is an $n > 0$ such that $S^{(n)} = \bot$. By Lemma 5.4, this implies that $S^{(0)} = AS(s_0 \not\equiv t_0)$ is unsatisfiable in E. By Proposition 3.3, it follows that $s_0 \not\equiv t_0$ is unsatisfiable in E, which means that $s_0 =_E t_0$. □

In order to show completeness of the combination procedure, we need an additional assumption on the theories E_1, E_2, which is, however, not a severe restriction: the theories must be non-trivial and coincide on the shared constants.

Condition 5.6 *E_1 and E_2 are non-trivial and, for all $c, d \in \Sigma = \Sigma_1 \cap \Sigma_2$, we have $c =_{E_1} d$ iff $c =_{E_2} d$.*

If there are distinct shared constants c, d such that $c =_{E_i} d$ holds, then one can dispense with one of the two constants by replacing it everywhere with the other. For this reason, we may consider (without loss of generality) the following condition in place of Condition 5.6.

Condition 5.7 *E_1 and E_2 are non-trivial and, for all $c, d \in \Sigma = \Sigma_1 \cap \Sigma_2$ such that $c \neq d$, we have $c \neq_{E_1} d$ and $c \neq_{E_2} d$.*

Proposition 5.8 *Assume that E_1, E_2 satisfy Condition 5.7, and let $E := E_1 \cup E_2$. If $s_0 =_E t_0$, then the combination procedure succeeds on input $\{s_0, t_0\}$.*

Proof. By Lemma 5.1, the procedure either succeeds or fails; therefore, we can prove the claim by proving that, if the procedure fails on input $\{s_0, t_0\}$, then $s_0 \neq_E t_0$. Thus, assume that the procedure fails, and let $S^{(n)}$ be the set obtained in the last transformation step. Given Lemma 5.4 and the construction of $S^{(0)}$, it is sufficient to show that $S^{(n)}$ is satisfiable in E.

From Lemma 5.3 we know that $S^{(n)}$ is an abstraction system with an initial formula of the form $x \not\equiv y$ for distinct variables x and y (otherwise the procedure would have succeeded). It follows that $S^{(n)} \setminus \{x \not\equiv y\}$ can be partitioned into the sets

$$S_1 \quad := \quad \{x_j \equiv s_j(\tilde{u}_j)\}_{j \in I} \quad \text{and} \quad S_2 \quad := \quad \{y_j \equiv t_j(\tilde{v}_j)\}_{j \in J},$$

where I and J are finite, $s_j \in T(\Sigma_1, V) \setminus V$, $t_j \in T(\Sigma_2, V) \setminus V$, and $\tilde{u}_j$ (resp. $\tilde{v}_j$) is the sequence of variables occurring in s_j (resp. t_j).[7] It is an easy consequence of Definition 3.2 that each x_j occurs only once in S_1,[8] each y_j occurs only once in S_2, and $\{x_i\}_{i \in I}$ and $\{y_j\}_{j \in J}$ are disjoint.

In the following, let X denote the set $\mathcal{V}(S_1) \cap \mathcal{V}(S_2)$ of all variables occurring in both S_1 and S_2. Let G_1 and G_2 be sets (of generators) of the same cardinality, and assume that this cardinality is infinite and greater than or equal to the cardinality of $\Sigma_1 \cup \Sigma_2$. For $i = 1, 2$, let $\mathcal{A}_i$ be an E_i-free algebra over the set of generators G_i. Since E_i was assumed to be non-trivial, one can deduce by simple cardinal arithmetic, that $\mathcal{A}_1$ and $\mathcal{A}_2$ have the same cardinality. We show below that

$$S_i \cup \{x \not\equiv y\} \cup \{\Delta(X), \Delta(X, \Sigma), \Delta(\Sigma)\}$$

[7] Each equation of the form $x \equiv c$ with $c \in \Sigma$, that is, $c \in T(\Sigma_1, V) \cap T(\Sigma_2, V)$, is put in either S_1 or S_2 arbitrarily.

[8] Note that condition (2) of Definition 3.2 entails that x_j cannot occur in $\tilde{u}_j$, whereas condition (3b) entails that x_j cannot occur in $\tilde{u}_{j'}$ for $j \neq j'$.

is satisfiable in $\mathcal{A}_i$, and thus, by Proposition 4.2, that $S^{(n)}$ is satisfiable in $E = E_1 \cup E_2$.

We restrict our attention to the case in which $i = 1$ ($i = 2$ can be treated analogously). First, note that Condition 5.7 obviously implies that the E_1-free algebra $\mathcal{A}_1$ satisfies $\Delta(\Sigma)$.

Now, let U denote the set of all variables occurring on the right-hand sides of equations in S_1 (that is, the variables in the sequences $\tilde{u}_j$). We consider a valuation α of $\mathcal{V}(S_1)$ into $\mathcal{A}_1$ assigning each $u \in U$ with a distinct element of $\mathcal{A}_1$'s set of generators, and each x_j with $s_j^{\mathcal{A}_1}[\alpha(\tilde{u}_j)]$ (i.e., the interpretation of the term s_j in $\mathcal{A}_1$ under this valuation of its variables). Notice that α is well-defined because all the x_j's are distinct and $x_j \notin U$, as we saw earlier. By construction, α satisfies S_1.

Next, we show that $\alpha(u) \neq \alpha(v)$ for all distinct $u, v \in \mathcal{V}(S_1)$, which will imply that α satisfies $\Delta(X)$.

If both u and v are in U, $\alpha(u) \neq \alpha(v)$ is obvious by the construction of α. Hence, let $u = x_j$ for some $j \in I$ and assume by contradiction that $\alpha(x_j) = \alpha(v)$. If $v = x_\ell$ for some $\ell \in I$, by the construction of α we have that $\mathcal{A}_1, \alpha \models s_j \equiv s_\ell$. Since α evaluates the variables in the equation $s_i \equiv s_j$ by distinct generators of $\mathcal{A}_1$, and since $\mathcal{A}_1$ is free for E_1, it follows that $s_j =_{E_1} s_\ell$; but then, since either $(x_\ell \equiv s_\ell) \not\prec^* (x_j \equiv s_j)$ or $(x_j \equiv s_j) \not\prec^* (x_\ell \equiv s_\ell)$, transformation (2d) applies to $S^{(n)}$, against the assumption that $S^{(n)}$ is the last set. If $v \in U$, similarly to the previous case we obtain that $v =_{E_1} s_j$. Now, if v does not occur in s_j, it is easy to see that E_1 only admits trivial models, against the assumption that E_1 is non-trivial. If v occurs in s_j, either transformation (2a) or (2b) applies to $S^{(n)}$, again against the assumption that $S^{(n)}$ is the last set.

Finally, we show, again by contradiction, that α satisfies $\Delta(X, \Sigma)$, that is, $\alpha(v) \neq c^{\mathcal{A}_1}$ for all $v \in X$ and $c \in \Sigma$. If $v \in U$, $\alpha(v)$ is a generator of $\mathcal{A}_1$, which certainly differs from $c^{\mathcal{A}_1}$ (otherwise, E_1 would be trivial). Therefore, suppose that $v = x_j$ for some $j \in I$, and that $\mathcal{A}_1, \alpha \models x_j \equiv c$. As before, this entails that $c =_{E_1} s_j$. Recalling our earlier observations on S_1 and S_2, and that v is shared by the two sets, we can deduce that S_2 contains an equation of the form $y \equiv t[x_j]$. Now, s_j cannot be a shared constant, otherwise $S^{(n)}$, for including $\{y \equiv t[x_j], x_j \equiv s_j\}$, would not satisfy condition (3b) of Definition 3.2. But if $s_j \notin \Sigma$ and $c =_{E_1} s_j$, then transformation (2c) applies to $S^{(n)}$.

In conclusion, we have shown that $\mathcal{A}_1, \alpha \models S_i \cup \{\Delta(X), \Delta(X, \Sigma), \Delta(\Sigma)\}$. To complete the proof, we must show that α also satisfies $x \not\equiv y$. We know that x, y are distinct. Thus, if both are elements of $\mathcal{V}(S_1)$, the above proof yields $\alpha(x) \neq \alpha(y)$. Otherwise, we simply need to extend α to $\mathcal{V}(S_1) \cup \{x, y\}$ so that $\alpha(x) \neq \alpha(y)$. □

Combining the results of this section, which show total correctness of our combination procedure, we obtain the following modularity result:

Theorem 5.9 *Let $E := E_1 \cup E_2$, where, for $i = 1, 2$, E_i is a non-trivial equational theory over Σ_i. Furthermore, assume that $\Sigma := \Sigma_1 \cap \Sigma_2$ is a finite set of constant symbols, and that E_1 and E_2 agree on Σ, that is, $c =_{E_1} d$ iff $c =_{E_2} d$ for all $c, d \in \Sigma$.*

If the word problem is decidable for E_1 and E_2, then it is also decidable for E.

6 Conclusion, related work and open questions

We have shown that decision procedures for the word problem can be combined with the help of a transformation-based procedure, if the signatures of the underlying theories share only finitely many constant symbols. Our main goal was not to prove a new combination result, but to develop a rule-based combination procedure, which is more transparent and more flexible than the known ones, and which uses deterministic rules that may be applied in arbitrary order. In addition, we wanted to clarify the connection to Nelson and Oppen's combination method.

An important open question is how far the combination result can be extended to theories sharing function symbols of larger arity. In [1], the problem of combining algorithm for the unification, matching, and word problem was investigated for theories "sharing constructors." For the word problem, this combination method presupposes the existence of a matching algorithm for certain restricted matching problems in the single theories. For shared constants, it is easy to see that these restricted matching problems reduce to word problems, and thus the result in [1] also yields the modularity result of Theorem 5.9, even though this is not explicitly mentioned in the paper. However, the algorithm described in [1] is not rule-based since it is a straightforward extension of the algorithms for the disjoint case, as described in [13, 9, 3], and thus shares the disadvantages of these algorithms, as mentioned in the introduction. A recent result of Domenjoud's [2] shows that for theories sharing constructors of arity > 0, the matching algorithms required by the approach of [1] cannot be dispensed with: a union of simple and regular theories sharing only constructors may have an undecidable word problem, even if each theory has a decidable word problem.

The approach presented in this paper suggests that we can tackle the problem of non-disjoint combination from another angle. The Nelson-Oppen combination method has been recently extended to the combination of theories with non-disjoint signatures [11] and more work in this direction is currently under way [16]. Since our approach is very similar in spirit to Nelson and Oppen's, it is conceivable that some of the results from [11, 16] may be used to extend our combination procedure to the case of equational theories sharing also non-constant functors. However, any such extension appears to be non-trivial and will probably impose more serious limitations on the type of equational theories that can be combined.

References

[1] E. Domenjoud, F. Klay, and C. Ringeissen. Combination techniques for non-disjoint equational theories. In A. Bundy, editor, *Proceedings 12th International Conference on Automated Deduction, Nancy (France)*, volume 814 of *Lecture Notes in Artificial Intelligence*, pages 267–281. Springer-Verlag, 1994.

[2] Eric Domenjoud. Undecidability of the word problem in the union of theories sharing constructors. In *Proceedings of the 10th International Workshop on Unification, UNIF'96*. CIS-Report 96-9, CIS, Universität München, 1996.

[3] Hélène Kirchner and Christophe Ringeissen. Combining symbolic constraint solvers on algebraic domains. *Journal of Symbolic Computation*, 18(2):113–155, 1994.

[4] Donald E. Knuth and P.B. Bendix. Simple word problems in universal algebra. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

[5] Alberto Martelli and Ugo Montanari. An efficient unification algorithm. *ACM Transactions on Programming Languages and Systems*, 4(2):258–282, 1982.

[6] J.V. Matijasevic. Simple examples of undecidable associative calculi. *Soviet Mathematics (Doklady)*, 8(2):555–557, 1967.

[7] Greg Nelson and Derek C. Oppen. Simplification by cooperating decision procedures. *ACM Trans. on Programming Languages and Systems*, 1(2):245–257, October 1979.

[8] Greg Nelson and Derek C. Oppen. Fast decision procedures based on congruence closure. *J. ACM*, 27(2):356–364, 1980.

[9] Tobias Nipkow. Combining matching algorithms: The regular case. In N. Dershowitz, editor, *Proceedings of the 3rd International Conference on Rewriting Techniques and Applications, Chapel Hill (N.C., USA)*, volume 335 of *Lecture Notes in Computer Science*, pages 343–358. Springer-Verlag, April 1989.

[10] Derek C. Oppen. Complexity, convexity and combinations of theories. *Theoretical Computer Science*, 12, 1980.

[11] Christophe Ringeissen. Cooperation of decision procedures for the satisfiability problem. In *Proceedings of the 1st International Workshop "Frontiers of Combining Systems", Munich (Germany)*, Applied Logic. Kluwer, 1996. (to appear).

[12] J.A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12:23–41, 1965.

[13] Manfred Schmidt-Schauß. Combination of unification algorithms. *Journal of Symbolic Computation*, 8(1–2):51–100, 1989.

[14] E. Tidén. *First-Order Unification in Combinations of Equational Theories.* Phd thesis, The Royal Institute of Technology, Stockholm, 1986.

[15] Cesare Tinelli and Mehdi Harandi. A new correctness proof of the Nelson–Oppen combination procedure. In *Proceedings of the 1st International Workshop "Frontiers of Combining Systems", Munich (Germany)*, Applied Logic. Kluwer, 1996. (to appear).

[16] Cesare Tinelli and Christophe Ringeissen. Unions of theories and combinations of satisfiability procedures. Technical report, Department of Computer Science, University of Illinois at Urbana-Champaign, 1996. (in preparation).

[17] Yoshihito Toyama. Counterexamples to termination for the direct sum of term rewriting systems. *Information Processing Letters*, 25:141–143, 1987.

[18] Yoshihito Toyama. On the Church-Rosser property for the direct sum of term rewriting systems. *Journal of the ACM*, 34(1):128–143, 1987.