Deciding the Word Problem in the Union of Equational Theories

Franz Baader[1]
baader@informatik.rwth-aachen.de

Cesare Tinelli
tinelli@cs.uiuc.edu

Department of Computer Science
University of Illinois at Urbana-Champaign

Report No. UIUCDCS-R-98-2073, UILU-ENG-98-1724

October 1998

[1]Address: Lehr- und Forschungsgebiet Theoretische Informatik, RWTH Aachen, Ahornstraße 55, 52074 Aachen, Germany.

Deciding the Word Problem in the Union of Equational Theories

Franz Baader
Lehr- und Forschungsgebiet Theoretische Informatik
RWTH Aachen
Ahornstraße 55, 52074 Aachen, Germany
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
tinelli@cs.uiuc.edu

October 1998

Abstract

The main contribution of this report is a new method for combining decision procedures for the word problem in equational theories. In contrast to previous methods, it is based on transformation rules, and also applies to theories sharing "constructors." In addition, we show that—contrary to a common belief—the Nelson-Oppen combination method cannot be used to combine decision procedures for the word problem, even in the case of equational theories with disjoint signatures.

Contents

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 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, 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 [Mat67]. Nevertheless, there are decidability results for certain classes of equational theories (such as theories defined by a finite set of ground equations [NO80]), and there are general approaches for tackling the word problem (such as Knuth-Bendix completion [KB70], which tries to generate a confluent and terminating term rewriting system for the theory).

The present report 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 [Toy87b], but unfortunately termination is not. In [Toy87a] 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 combination result was first proved in [Pig74] using results from universal algebra. It was more recently rediscovered in the term rewriting and automated deduction community [Tid86, SS89, Nip89, KR94]. Surprisingly, even these more recent presentations did not appear to be widely known in the computer science community, possibly because the result 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 [Tid86], it is not explicitly stated there; in [SS89, KR94] it is stated as a corollary, but not mentioned in the abstract or the introduction; only [Nip89] explicitly refers to the result in the abstract. The combination methods used in all these papers 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 report, which is a revised and significantly extended version of [BT97], 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 [MM82] and Robinson's original one [Rob65]. 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 function symbols* that are "constructors" (in a sense to be made more precise later).

The only other work that presents a combination method for the word problem in the union of non-disjoint theories is [DKR94], where the problem of combining algorithm for the unification, matching, and word problem was also investigated for theories sharing so-called "constructors." The combination method for the word problem described in [DKR94] is not rule-based since it is a straightforward extension of the algorithms for the disjoint case, as described in [Pig74, SS89, Nip89, KR94],

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

and thus shares the disadvantages of these algorithms. We will show that the notion of a constructor introduced in [DKR94] is a strict subcase of our notion, and that the combination result for the word problem presented in [DKR94] can also be obtained with the help of our rule-based approach.

There is a persistent rumor that combining decision procedures for the word problem in the disjoint signature case is a special case of Nelson and Oppen's combination method [NO79]. 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, this idea does not quite work, and 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. However, our method for combining decision procedures for the word problem follows an approach very similar to Nelson and Oppen's.

Outline of the report The next section introduces some necessary notation. Section 3 briefly describes the Nelson-Oppen combination procedure, and investigates whether it can be applied to equational theories. In Section 4, we introduce a first version of our combination procedure for the word problem, which works for equational theories over disjoint signatures. Before we can extend this procedure to the nondisjoint combination of equational theories, we must establish (in Section 5) some general model-theoretic results for combined equational theories (Subsection 5.1) and introduce our notion of a constructor (Subsection 5.2). Subsection 5.3 contains some results concerning the union of theories sharing constructors. In Section 6 we describe the extended combination procedure for theories sharing constructors, and prove its correctness. Section 7 investigates the connection between our notion of a constructor and the one introduced in [DKR94].

2 Formal Preliminaries

In the context of the Nelson-Oppen procedure, we will consider arbitrary first-order theories over a given 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

[2] It turns out, however, that they satisfy a somewhat weaker property, which in principle suffices to apply their method.

always present and thus needs not be included in the signature. The signature Σ is called *functional* iff $\Sigma^{\mathrm{P}} = \emptyset$. In this case, we will use Σ in place of Σ^{F}.

Throughout the report, we will only consider countable signatures, we will denote by V a fixed countably infinite set of variables and by $T(\Sigma^{\mathrm{F}}, V)$ the set of Σ^{F}-terms over V. We will use the symbols r, s, t to denote terms, and the symbols x, y, u, v, w, z to denote variables. With a common abuse of notation we will also use x, y, u, v, w as the actual variables in our examples. If t is a term, we will denote by $t(\epsilon)$ the top symbol of t and by $\mathcal{V}ar(t)$ the set of all variables occurring in t. Similarly, if φ is a formula, $\mathcal{V}ar(\varphi)$ will denote the set of free variables of φ.

Where $\bar{v}$ is a tuple of variables without repetition, we will write $t(\bar{v})$ to say that $\bar{v}$ lists *all* the variables of t. When convenient we will treat a tuple $\bar{t}$ of terms as the set of its elements.

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 $P \in \Sigma^{\mathrm{P}} \cup \{\equiv\}$ is an n-ary predicate symbol and $s_1, \ldots, s_n \in T(\Sigma^{\mathrm{F}}, V)$ are Σ^{F}-terms with variables from V. As usual, we say that a quantifier-free formula φ is *valid* in a theory 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 $\mathcal{A}, \alpha \models \neg\varphi$.

Given a function symbol $f \in \Sigma^{\mathrm{F}}$ and a Σ-structure $\mathcal{A}$, we denote by $f^{\mathcal{A}}$ the interpretation of f in $\mathcal{A}$. This notation can be extended to terms in the obvious way: if s is a Σ-term containing n distinct variables, then we denote by $s^{\mathcal{A}}$ the n-ary term function induced by the term s in $\mathcal{A}$. Given a Σ^{F}-term s, a Σ-structure $\mathcal{A}$, 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. Using the term function induced by s, this interpretation of s can also be written as $[\![s]\!]^{\mathcal{A}}_{\alpha} = s^{\mathcal{A}}(\bar{a})$, where $\bar{a}$ is the tuple of values which α assigns to the variables in s.

In the context of equational theories, the attention is restricted to functional signatures. An equational theory E over the functional signature Σ is a set of universally quantified equations between Σ-terms. As usual, we will omit the universal quantifiers; for example, we will denote the equational theory C axiomatizing the commutativity of the binary function symbol f by $C := \{f(x, y) \equiv f(y, x)\}$ instead of $C := \{\forall x, y. f(x, y) \equiv f(y, x)\}$. 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.

The equational theory E over the signature Σ defines a Σ-*variety*, i.e., the class of all model of E. When E is *non-trivial* i.e., has models of cardinality greater 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, the E-free algebra with generators X can be obtained as the quotient term algebra $\mathcal{T}(\Sigma, X)/{=_E}$. The following is a well-known characterization of free algebras (see, e.g., [Hod93]):

Proposition 1 *Let E be an equational theory over Σ and $\mathcal{A}$ a Σ-algebra. Then, $\mathcal{A}$ is E-free with generators X iff the following holds:*

1. *$\mathcal{A}$ is a model of E;*

2. *X generates $\mathcal{A}$;*

3. *for all $s, t \in T(\Sigma, V)$, if $\mathcal{A}, \alpha \models s \equiv t$ for some injection α of $\mathcal{V}ar(s \equiv t)$ into X, then $s =_E t$.*

In this report, 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) functional 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 $t(\epsilon) \in V \cup \Sigma_i$, i.e., if it is a variable or has the form $t = f(t_1, ..., t_{n-1})$ for some i-symbol f $(i = 1, 2)$. Notice 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. Notice that every Σ_i-term is a pure i-term and vice versa. An equation $s \equiv t$ is pure iff there is an i such that both s and t are pure i-terms.

3 The Nelson-Oppen Combination Method

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

3.1 The General Method

This method is concerned with combining decision procedures for the validity of quantifier-free formulae. 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?

When considering the satisfiability problem, as done in Nelson and Oppen's method, 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.

Given a conjunctive quantifier-free $(\Sigma_1 \cup \Sigma_2)$-formula φ to be tested for satisfiability, *Nelson and Oppen's method for combining decision procedures* 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 Consider the (equational) theories $E_1 := \{f(x, x) \equiv x\}$ and $E_2 := \{g(g(x)) \equiv g(x)\}$ over the signatures $\Sigma_1 := \{f\}$ and $\Sigma_2 := \{g\}$.[4] If we want to know whether the (mixed) quantifier-free formula

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

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

[4] Recall that the equations in E_i are implicitly assumed to be universally quantified.

is valid in $E_1 \cup E_2$, 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 E_1 (of cardinality 1). The formula φ_2 is, for example, satisfiable in the E_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 E_2 since φ_2 contains $v \equiv g(z) \wedge w \equiv g(g(z))$ and E_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 E_1, and its propagation yields

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

In *Step 2*, it turns out that φ_2'' is not satisfiable in E_2, and thus the answer is "unsatisfiable," which shows that $g(f(g(z), g(g(z)))) \equiv g(z)$ is valid in $E_1 \cup E_2$. In fact, $v \equiv g(z)$ and the equation $g(g(x)) \equiv g(x)$ of E_2 imply that $g(v) \equiv g(z)$, which contradicts $g(v) \not\equiv g(z)$.

Obviously, the procedure always 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 4 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 [NO79]; it was introduced in [Opp80]. Two new and simple proofs of soundness and completeness of the procedure are given in [Rin96, TH96]. In essence, both depend on the following proposition (see [TH96] for a proof). For a finite set of variables X, let Δ_X denote the conjunction of all disequations $x \not\equiv y$ for $x, y \in X, x \neq y$.

Proposition 3 *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 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, then 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 will use this proposition also to prove the correctness of our combination procedure in the disjoint case.

3.2 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 (implicitly) 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 4 Consider the theory

$$E_1 \;:=\; \{f(g(x), g(y)) \equiv x,\;\; f(g(x), h(y)) \equiv y\}.$$

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 in which $g(x) \equiv h(x)$ is satisfiable, 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

$$E_2 \ := \ \{k(x) \equiv k(x)\}$$

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 $E_i' := E_i \cup \{\exists x, y.\ x \not\equiv y\}$.

Lemma 5 *Let E be a non-trivial equational theory.*

1. *The theory $E' := E \cup \{\exists x, y.\ x \not\equiv y\}$ is stably infinite.*

2. *If the satisfiability in E of quantifier-free formulae is decidable, then the satisfiability in E' is also decidable.*

Proof. The *second statement* is immediate. In fact, let φ be a quantifier-free formula. Then φ is satisfiable in E' iff the quantifier-free formula $\varphi \wedge x \not\equiv y$ is satisfiable in E, where x, y are two distinct variables not occurring in φ.

The *first statement* 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 is satisfiable in E', i.e., φ is satisfiable in a non-trivial model $\mathcal{A}$ of E. Then the countably infinite direct product of $\mathcal{A}$ with itself is an infinite model of E (and of E'), and it is easy to see that it satisfies φ. $\square$

The lemma shows that the prerequisites for applying Nelson and Oppen's procedure are satisfied for the combined theory $E_1' \cup E_2'$, provided that the theories E_i are non-trivial and satisfiability of quantifier-free formulae are decidable in E_i ($i = 1, 2$). Thus, satisfiability of quantifier-free formulae in $E_1' \cup E_2'$ is decidable. Since a quantifier-free formula is satisfiable in $E_1 \cup E_2$ iff it is satisfiable in the trivial model or in a model of $E_1' \cup E_2'$, and since we have already seen that satisfiability in the trivial model is decidable, we obtain the following theorem.

Theorem 6 *Let E_1 and E_2 be two equational theories over disjoint signatures. If the satisfiability in E_i of quantifier-free formulae is decidable ($i = 1, 2$), then the satisfiability of quantifier-free formulae in $E_1 \cup E_2$ is also decidable.*

3.3 Its Application to the Word Problem

Recall that the word problem is concerned with the validity in E of quantifier-free formulae of the form $s \equiv t$ (equivalently: (un)satisfiability of $s \not\equiv t$ in E). Now, let E_1 and E_2 be two equational theories over disjoint signatures. Theorem 6 implies that the word problem is decidable for $E_1 \cup E_2$, provided that the validity (equivalently: satisfiability) in E_1 and E_2 of *arbitrary quantifier-free formulae* is decidable.

However, the assumption that the word problem (equivalently: the satisfiability of formulae of the form $s \not\equiv t$) is decidable for E_i ($i = 1, 2$) is too weak for a straightforward application of the Nelson-Oppen procedure. In fact, 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. The procedure for the word problem considers the satisfiability of a single disequation. In the second step of Nelson and Oppen's procedure, satisfiability of a conjunction consisting of at most one disequation and finitely many equations must be tested, and in the third step, satisfiability of a conjunction of finitely many disequations and finitely many equations must be tested.

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 $x \not\equiv x$.

[5]The rules of our combination approach are somewhat more complex for technical reasons.

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 A Combination Procedure for the Word Problem: The Disjoint Case

In the following, we will present a decision procedure for the word problem in an equational theory of the form $E_1 \cup E_2$ where each E_i is a non-trivial equational theory with decidable word problem. To simplify the exposition, we will start by first considering the case in which the signatures of E_1 and E_2 are disjoint. In Section 6 we will then extend the results given in this section to the case in which the two signatures are not disjoint. Of course, this requires some additional restrictions on the theories to be combined. These restrictions will be introduced in Section 5. Both in the disjoint and the nondisjoint case, we will assume (with no loss of generality) that all the signatures considered are countable.

To decide the word problem for $E := E_1 \cup E_2$, 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 terms in the signature of E. As in the Nelson-Oppen procedure, the first step of our procedure transforms a formula of this form into a conjunction of pure formulae by means of variable abstraction. To define the purification process in more detail, we need to introduce a little notation and some new concepts.

4.1 Abstraction Systems

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

We can define a procedure which, given a disequation $s_0 \not\equiv t_0$ with $s_0, t_0 \in T(\Sigma_1 \cup \Sigma_2, V)$, produces an *equisatisfiable* set $AS(s_0 \not\equiv t_0)$ consisting of pure equations and disequations.[6] The *purification 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, the procedure uses s_0 in place of x (t_0 in place of y), and omits 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

[6] Equisatisfiable means that, for all algebras $\mathcal{A}$, the disequation $s_0 \not\equiv t_0$ is satisfiable in $\mathcal{A}$ iff $AS(s_0 \not\equiv t_0)$ is satisfiable in $\mathcal{A}$.

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 the procedure stops and returns S_i.

It is easy to see that 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 7 *Let $x, y \in V$ and S be a set of equations of the form $v \equiv t$ where $v \in V$ and $t \in T(\Sigma_1 \cup \Sigma_2, V) \setminus V$. The relation $\prec$ is the smallest binary relation on $\{x \not\equiv y\} \cup S$ such that, for all $u \equiv s, v \equiv t \in S$,*

$$\begin{array}{lll} (x \not\equiv y) \prec (v \equiv t) & \textit{iff} & v \in \{x, y\}, \\ (u \equiv s) \prec (v \equiv t) & \textit{iff} & v \in \mathcal{V}ar(s). \end{array}$$

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 S such that $(v \equiv t) \prec^+ (v \equiv t)$.*

Notice that, when $\prec$ is acyclic, $\prec^*$ is a partial order, and $\prec^+$ is the corresponding strict partial order. Since $\prec$ is the smallest relation satisfying the above properties, the disequation $x \not\equiv y$ is not larger than any equation $u \equiv s$, even if x or y occurs in s.

Definition 8 (Abstraction System) *The set $\{x \not\equiv y\} \cup S$ is an* abstraction system with initial formula *$x \not\equiv y$ iff $x, y \in V$ and the following holds:*

1. *S 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 $(u \equiv s), (v \equiv t) \in S$,*

   (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 $t(\epsilon) \notin \Sigma_i$.

Condition (1) above 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 $u \notin \mathcal{V}ar(t)$; Condition (3a) implies that a variable cannot occur as the left-hand side of more than one equation of S; Condition (3b) implies, together with Condition (1), that the elements of every $\prec$-chain of S have *strictly* alternating signatures $(\ldots, \Sigma_1, \Sigma_2, \Sigma_1, \Sigma_2, \ldots)$.

The following proposition is an easy consequence of the definition of the purification procedure.

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

1. Let $S := AS(s_0 \not\equiv t_0)$.
2. Repeatedly apply (in any order) **Coll1**, **Coll2**, **Ident**, **Simpl** to S until none of them is applicable.
3. Succeed if S has the form $\{v \not\equiv v\} \cup T$, and fail otherwise.

Figure 1: The Combination Procedure.

Proposition 9 *Let $\mathcal{A}$ be a $(\Sigma_1 \cup \Sigma_2)$-algebra. The set $AS(s_0 \not\equiv t_0)$ obtained by applying the purification procedure to the disequation $s_0 \not\equiv t_0$ is an abstraction system which is satisfiable in $\mathcal{A}$ iff $s_0 \not\equiv t_0$ is satisfiable in $\mathcal{A}$.*

In particular, this proposition implies that the disequation $s_0 \not\equiv t_0$ is satisfiable in E iff $AS(s_0 \not\equiv t_0)$ is satisfiable in E.

4.2 The Combination Procedure

Let Σ_1 and Σ_2 be two disjoint (functional) signatures, and assume that E_i is a non-trivial equational theory over Σ_i with decidable word problem, for $i = 1, 2$. Fig. 1 describes a procedure that decides the word problem for the theory $E := E_1 \cup E_2$ by deciding, as we will show, 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. It repeatedly applies the transformation rules of Fig. 2 until no more rules are applicable.

The main idea of the procedure 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 of the equations between variables that are valid in one of 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 initial disequation $s_0 \not\equiv t_0$ is unsatisfiable in E (that is, iff $s_0 =_E t_0$).

During the execution of the 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 formulas explicitly listed in the premise of a rule are distinct.

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

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

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

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

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

if s, t are i-terms and $s =_{E_i} t$ for $i = 1$ or $i = 2$
and
$(y \equiv t) \not\prec^* (x \equiv s)$.

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

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

Figure 2: The Derivation Rules.

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. 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.[7]

Simpl eliminates those equations that have become unreachable along a $\prec$-path from the initial disequation because of the application of previous rules. As we will see, 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

[7]Notice that our use of the notation $[x/t]$ contrasts with the common practice in the literature (for instance, of programming languages theory). There, the expression $T[x/t]$ above would be written as $T[t/x]$ instead. We prefer our convention because we find it more intuitive, especially in the case of composed substitutions.

optimized, complete implementations of the combination procedure.

We prove in Section 4.3 that this combination procedure decides the word problem for E by showing that the procedure is partially correct (i.e., sound and complete) and terminates on all inputs.

4.3 The Correctness Proof

In the following, we will denote by S_0 the abstraction system $AS(s_0 \not\equiv t_0)$ obtained by applying the purification procedure to the input disequation, and by S_j $(j \geq 1)$ the set S of formulae generated by the combination procedure at the j^{th} iteration of Step 2. If Step 2 is iterated only n times, we will define $S_j := S_n$ for all $j > n$. Correspondingly, we will let $\prec_j$ denote the relation $\prec$ on S_j (cf. Def. 7).

We first show that all sets S_j obtained in correspondence of one run of the combination procedure are in fact abstraction systems. The proof of acyclicity (Condition 2 in Definition 8) will be facilitated by the following lemma, whose simple proof is omitted.

Lemma 10 *Let $<$ be a binary relation on a finite set A, and $a, b \in A$ be such that $b \not<^* a$. We denote the restriction of $<$ to $A \setminus \{a\}$ by $<_a$,*[8] *and consider the relations*

$$<_1 \;:=\; <_a \cup \{\langle d,\, e\rangle \mid d < a, b < e\}$$
$$<_2 \;:=\; <_a \cup \{\langle d,\, b\rangle \mid d < a\}.$$

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

Lemma 11 *Given an execution of the combination procedure, S_j is an abstraction system for all $j \geq 0$.*

Proof. We prove the claim by induction on j. The induction base $(j = 0)$ is immediate by construction of S_0 and Proposition 9. Thus, assuming that $j > 0$ and that S_{j-1} is an abstraction system, consider the following cases, labeled by the derivation rule applied to S_{j-1} to obtain S_j.

Coll1. By the rule's definition, S_{j-1} and S_j must have the following form:

$$\begin{array}{lclclclcl} S_{j-1} & = & \{u \not\equiv v\} & \cup & \{x \equiv t[y]\} & \cup & \{y \equiv r\} & \cup & T \\ S_j & = & \{u \not\equiv v\}[x/y] & \cup & & & \{y \equiv r\} & \cup & T[x/r] \end{array}$$

Let $u' \not\equiv v' := (u \not\equiv v)[x/y]$. We show that S_j is an abstraction system with initial formula $u' \not\equiv v'$.

Let $S'_{j-1} := S_{j-1} \setminus \{u \not\equiv v\}$ and $S'_j := S_j \setminus \{u' \not\equiv v'\}$, and let $\prec'_{j-1}$ and $\prec'_j$ respectively be the restrictions of $\prec_{j-1}$ and $\prec_j$ to these sets of equations. If we take

[8] That is, $<_a := < \cap (A \setminus \{a\})^2$.

$\prec'_{j-1}$ to be the relation $<$ of Lemma 10, $x \equiv t$ to be a, and $y \equiv r$ to be b, it is easy to see that $\prec'_j$ coincides with $<_1$ (as defined in the lemma). Because $\prec_{j-1}$ is acyclic by induction, its subrelation $< = \prec'_{j-1}$ is acyclic as well. Since we know that $a < b$, this also implies that $b \not<^* a$, and thus the preconditions of Lemma 10 are satisfied. It follows that $\prec'_j$ is acyclic. By definition of the relation $\prec$, the initial disequation cannot be involved in a cycle, and thus $\prec_j$ is acyclic as well. This shows that condition (2) of Definition 8 holds.

Since applying the substitution $[x/r]$ does not change the left-hand sides of equations in T, it is immediate that condition (3a) of Definition 8 holds as well.

Finally, observe that x can appear in T only in an equation of the form $z \equiv s[x]$ and that $(z \equiv s) \prec_{j-1} (x \equiv t) \prec_{j-1} (y \equiv r)$. By induction, we know that there is an $i \in \{1, 2\}$ such that s and r are both non-variable Σ_i-terms; therefore, the replacement of x by r in T does not generate non-pure terms and it does not change the signature of the equations in T. It follows that S_j satisfies both conditions (1) and (3b) of Definition 8.

Coll2. The proof is essentially a special case of the one above, with r replaced by y. The proof of condition (2) of Definition 8 is, however, easier in this case. If we take $x \equiv t$ to be a and $\prec_{j-1}$ to be the relation $<$, then $\prec_j$ coincides with $<_a$ as defined in Lemma 10. If $<$ is acyclic, then its subrelation $<_a$ is acyclic as well.

Ident. By the rule's definition, S_{j-1} and S_j must have the following form:

$$
\begin{array}{lllllll}
S_{j-1} & = & T \cup \{u \not\equiv v\} & \cup & \{x \equiv s\} & \cup & \{y \equiv t\} \\
S_j & = & (T \cup \{u \not\equiv v\})[x/y] & \cup & & & \{y \equiv t\},
\end{array}
$$

where 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 10, where $x \equiv s$ is a and $y \equiv t$ is b. Again, the preconditions of the lemma are satisfied, and it follows that $\prec_j$ satisfies condition (2) of Definition 8. By induction, we know that x appears as the left-hand side of no equations in T, and so it is immediate that S_j satisfies condition (3a). It is also immediate that S_j satisfies condition (1). Finally, since s and t are both non-variable i-terms, S_j also satisfies condition (3b). It follows that S_j is an abstraction system with initial formula $(u \not\equiv v)[x/y]$.

Simpl. Immediate consequence of the easily provable fact that, if $\{u \not\equiv v\} \cup T'$ is an abstraction system, then $\{u \not\equiv v\} \cup T$ is also an abstraction system for every $T \subseteq T'$. □

The next lemma shows that the derivation rules preserve satisfiability.

Lemma 12 *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. First assume that S_j has been produced by an application of **Coll1**. We know that S_{j-1} and S_j have the form

$$\begin{array}{lllllllll} S_{j-1} & = & \{u \not\equiv v\} & \cup & \{x \equiv t[y]\} & \cup & \{y \equiv r\} & \cup & T \\ S_j & = & \{u \not\equiv v\}[x/y] & \cup & & & \{y \equiv r\} & \cup & T[x/r] \end{array}$$

and that $y =_{E_i} t$ for $i = 1$ or $i = 2$. Assume that the valuation α satisfies S_{j-1} in the model $\mathcal{A}$ of E. Since $y \equiv t$ is valid in E, for being valid in E_i, α must assign both x and y with $[\![t]\!]_\alpha^\mathcal{A}$, i.e., the interpretation of the term t in $\mathcal{A}$ under the valuation α. In addition, since α satisfies S_{j-1}, we know that $\alpha(y) = [\![r]\!]_\alpha^\mathcal{A}$. It follows immediately that α satisfies S_j in $\mathcal{A}$.

Now, assume that the valuation α satisfies S_j in the model $\mathcal{A}$ of E. Observe that, since S_{j-1} is an abstraction system, x does not occur in $y \equiv r$, and as a consequence it does not occur in S_j at all. Let α' be the valuation defined by $\alpha'(z) := \alpha(z)$ for all $z \neq x$ and $\alpha'(x) := \alpha(y)$. It is immediate that α' satisfies the set $T_1 := T \cup \{x \equiv r\} \cup \{u \not\equiv v\} \cup \{x \equiv y\} \cup \{y \equiv r\}$ in $\mathcal{A}$. Since $\mathcal{A}$ is a model of E and the equation $y \equiv t$ is valid in E, it is also immediate that α' satisfies the set $T_2 := \{x \equiv t\}$ in $\mathcal{A}$. It follows that α' satisfies S_{j-1}, which is a subset of $T_1 \cup T_2$.

The proof for **Coll2** can be derived as a special case of the one for **Coll2** with r replaced by y. **Ident** can be treated similarly.

Simpl. In this case S_{j-1} and S_j have the form

$$\begin{array}{lllll} S_{j-1} & = & T & \cup & \{x \equiv t\} \\ S_j & = & T & & \end{array}$$

with $x \notin \mathcal{V}ar(T)$. It immediate that, if S_{j-1} is satisfiable in $\mathcal{A}$, so is S_j. Assume then that S_j is satisfied in $\mathcal{A}$ by a valuation α of T's variables. We can first choose an arbitrary extension α' of α to the variables in $\mathcal{V}ar(t) \setminus \mathcal{V}ar(T)$. From the assumptions and the fact that S_{j-1} is an abstraction system we know that x does not occur in $\mathcal{V}ar(t) \cup \mathcal{V}ar(T)$. Therefore, we can further extend α' so that it assigns to x the individual denoted by t, i.e., $\alpha'(x) := [\![t]\!]_{\alpha'}^\mathcal{A}$. It follows that α' satisfies $T \cup \{x \equiv t\}$ in $\mathcal{A}$. □

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

Proposition 13 *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 $S_n = \{v \not\equiv v\} \cup T$. Since S_n is clearly unsatisfiable in E, we can conclude by a repeated application of Lemma 12 that $S_0 = AS(s_0 \not\equiv t_0)$ is also unsatisfiable in E. By Proposition 9, it follows that $s_0 \not\equiv t_0$ is unsatisfiable in E, which means that $s_0 =_E t_0$. □

Next, we show that the combination procedure always terminates.

Lemma 14 *The combination procedure halts on all inputs.*

Proof. As mentioned above, the purification procedure used in Step 1 of the combination procedure terminates. In addition, since every equivalence test in the derivation rules can be performed in finite time because of the decidability of the word problems in E_1 and in E_2, every run of Step 2 is also executable in finite time. All we need to show then is that the procedure performs Step 2 only finitely many times. For $j \geq 0$, let N_j be the number of variables occurring on the left-hand side of an equation in S_j. Looking at each derivation rule, it is easy to see that $N_0 > N_1 > N_2 \ldots$, which means that the total number of repetitions of Step 2 is bounded by N_0. □

Finally, we show that the combination procedure is also complete.

Proposition 15 *The combination procedure succeeds on input* (s_0, t_0) *if* $s_0 =_E t_0$.

Proof. By Lemma 14, 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 12 and the construction of S_0, it is sufficient to show that S_n is satisfiable in E.

From Lemma 11 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

$$T_1 \ := \ \{x_j \equiv s_j(\bar{u}_j)\}_{j \in I} \quad \text{and} \quad T_2 \ := \ \{y_j \equiv t_j(\bar{v}_j)\}_{j \in J},$$

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 $\bar{u}_j$ (resp. $\bar{v}_j$) is the sequence of variables occurring in s_j (resp. t_j). It is an easy consequence of Definition 8 that each x_j occurs only once in T_1,[9] each y_j occurs only once in T_2, and $\{x_i\}_{i \in I}$ and $\{y_j\}_{j \in J}$ are disjoint.

Let X be the set $\mathcal{V}ar(T_1) \cap \mathcal{V}ar(T_2)$ of all variables occurring in both T_1 and T_2. For $i = 1, 2$, let $\mathcal{A}_i$ be an E_i-free algebra over a countably infinite set of generators Y_i. Since E_i is non-trivial by assumption, this free algebra exists and has a cardinality greater than 1. Consequently, $\mathcal{A}_i$ is also a model of $E'_i := E_i \cup \{\exists u, v.\ u \not\equiv v\}$. By Lemma 5, the theory E'_i is stably infinite. We show below that, for $i = 1, 2$,

$$T_i \cup \{x \not\equiv y\} \cup \Delta_X$$

[9] Note that condition (2) of Definition 8 entails that x_j cannot occur in $\bar{u}_j$, condition (3b) entails that x_j cannot occur in $\bar{u}_{j'}$ for $j \neq j'$, and condition (3a) entails that $x_j \neq x_{j'}$ for $j \neq j'$.

is satisfiable in $\mathcal{A}_i$, where Δ_X is the set of all the disequations between two distinct elements of X. It will then follow from an application of Proposition 3 that S_n is satisfiable in $E'_1 \cup E'_2$, and thus also in $E = E_1 \cup E_2$.

We restrict our attention to the case in which $i = 1$ ($i = 2$ can be treated analogously). Let U be the set of all variables occurring on the right-hand sides of equations in T_1 (that is, the variables in the sequences $\bar{u}_j$). We consider a valuation α of $\mathcal{V}ar(T_1)$ into $\mathcal{A}_1$ assigning each $u \in U$ with a distinct element of Y_1 and each x_j with $[\![s_j]\!]_\alpha^{\mathcal{A}_1}$ (the interpretation of the term s_j in $\mathcal{A}_1$ under the variable assignment α). 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 T_1.

Next, we show that $\alpha(u) \neq \alpha(v)$ for all distinct variables $u, v \in \mathcal{V}ar(T_1)$, which will imply that α satisfies Δ_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 $\mathcal{A}_1$ is E_1-free, it follows by Proposition 1 that $s_j =_{E_1} s_\ell$; but then, given that either $(x_\ell \equiv s_\ell) \not\prec^* (x_j \equiv s_j)$ or $(x_j \equiv s_j) \not\prec^* (x_\ell \equiv s_\ell)$ by the acyclicity of abstraction systems, the derivation rule **Ident** applies to S_n, against the assumption that S_n was 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 **Coll1** or **Coll2** applies to S_n, again against the assumption that S_n was the last set.

In conclusion, we have shown that

$$\mathcal{A}_1, \alpha \models T_1 \cup \Delta_X.$$

To complete the proof, we must show that α also satisfies $x \not\equiv y$. Recall that x, y are distinct. Thus, if they both occur in T_1, we already know by the above that $\alpha(x) \neq \alpha(y)$. Otherwise, we simply need to extend α to $\mathcal{V}ar(T_1) \cup \{x, y\}$ so that $\alpha(x) \neq \alpha(y)$. □

As an aside, we would like to point out that nowhere in the proof of Proposition 15 did we use the fact that **Simpl** can no longer be applied. Thus, the proof also shows that the modified procedure obtained by removing the rule **Simpl** is complete. Obviously, this modified procedure is sound and terminating as well.

Combining the results of this section, which show total correctness of the procedure, we also obtain the known modularity result for the word problem in case of component theories with disjoint signatures.

Theorem 16 *For $i = 1, 2$, let E_i be a non-trivial equational theory of signature Σ_i such that $\Sigma_1 \cap \Sigma_2 = \emptyset$. If the word problem is decidable for E_1 and for E_2, then it is also decidable for $E_1 \cup E_2$.*

A closer look at the termination proof and the definition of the purification procedure reveals that, modulo the complexity of the decision procedures for the word problem in the single theories, our combination procedure is polynomial.

Corollary 17 *Let E_1 and E_2 be non-trivial equational theories over disjoint signatures whose word problems are decidable in polynomial time. Then the word problem for $E_1 \cup E_2$ is also decidable in polynomial time.*

5 Combining Non-Disjoint Equational Theories

The rest of this report is concerned with the question of how the combination result stated in Theorem 16 can be lifted to the combination of equational theories whose signatures are not disjoint. As shown in the introduction, the union of equational theories with decidable word problem need not have a decidable word problem. Thus, one needs appropriate restrictions on the theories to be combined. The purpose of this section is to introduce such restrictions, and to establish some useful properties of theories satisfying these conditions. In particular, we will show a result that will play the rôle of Proposition 3 in the proof of completeness of the combination procedure.

Several of the results in Subsections 5.1 and 5.2 below[10] are special cases of more general results first described in [TR98]. That work considers the problem of combining decision procedures for the satisfiability of first-order formulae with respect to arbitrary first-order theories. The combination method described there is a proper extension of the Nelson-Oppen method and as such cannot be applied to the word problem, exactly for the same reasons given in Section 3. However, the proof of correctness for the combination procedure in [TR98] uses some general results about the combination of models of first-order sentences which are useful for our purposes as well. We have adapted some of the concepts and results introduced in [TR98] to the more specific context of equational theories. Although the proofs of the original results carry over to this special case, we provide direct proofs here both for completeness' sake and because they are simpler.

5.1 Fusions of Algebras and Unions of Equational Theories

In the following, given an Ω-algebra $\mathcal{A}$ and a subset Σ of Ω, we will denote by $\mathcal{A}^\Sigma$ the reduct of $\mathcal{A}$ to the subsignature Σ. Furthermore, we will use the symbol A to denote the carrier of $\mathcal{A}$.

[10]Specifically, Proposition 19, Lemma 20, Proposition 21, and Proposition 31. Also, what we give as a *characterization* of constructors in Theorem 24 is a special case of the *definition* of constructors in [TR98].

Definition 18 *The* $(\Sigma_1 \cup \Sigma_2)$*-algebra* $\mathcal{F}$ *is a* fusion *of the* Σ_1*-algebra* $\mathcal{A}_1$ *and the* Σ_2*-algebra* $\mathcal{A}_2$ *iff* $\mathcal{F}^{\Sigma_1}$ *is* Σ_1*-isomorphic to* $\mathcal{A}_1$ *and* $\mathcal{F}^{\Sigma_2}$ *is* Σ_2*-isomorphic to* $\mathcal{A}_2$*.*

We will denote by $Fus(\mathcal{A}_1, \mathcal{A}_2)$ the set of all the fusions of $\mathcal{A}_1$ and $\mathcal{A}_2$. By the above definition, it is immediate that $Fus(\mathcal{A}_1, \mathcal{A}_2) = Fus(\mathcal{A}_2, \mathcal{A}_1)$ and that $Fus(\mathcal{A}_1, \mathcal{A}_2)$ is closed under $(\Sigma_1 \cup \Sigma_2)$-isomorphism.[11]

In essence, a fusion of $\mathcal{A}_1$ and $\mathcal{A}_2$, if it exists, is an algebra that is identical to $\mathcal{A}_1$ when seen as a Σ_1-algebra, and identical to $\mathcal{A}_2$ when seen as a Σ_2-algebra. We can show that two algebras admit a fusion exactly when they have the same cardinality and interpret in the same way the symbols shared by their signatures.

Proposition 19 *Let* $\mathcal{A}$ *be a* Σ_1*-algebra,* $\mathcal{B}$ *a* Σ_2*-algebra, and* $\Sigma := \Sigma_1 \cap \Sigma_2$*. Then,* $Fus(\mathcal{A}, \mathcal{B}) \neq \emptyset$ *iff* $\mathcal{A}^\Sigma$ *is* Σ*-isomorphic to* $\mathcal{B}^\Sigma$*.*

Proof. ($\Rightarrow$) Let $\mathcal{F} \in Fus(\mathcal{A}, \mathcal{B})$. By definition we have that $\mathcal{A} \cong \mathcal{F}^{\Sigma_1}$ and $\mathcal{B} \cong \mathcal{F}^{\Sigma_2}$. From the fact that $\Sigma \subseteq \Sigma_1$ and $\Sigma \subseteq \Sigma_2$ it follows immediately that $\mathcal{A}^\Sigma \cong \mathcal{F}^\Sigma$ and $\mathcal{B}^\Sigma \cong \mathcal{F}^\Sigma$, which implies that $\mathcal{A}^\Sigma \cong \mathcal{B}^\Sigma$.

($\Leftarrow$) Let h be an arbitrary Σ-isomorphism of $\mathcal{A}^\Sigma$ onto $\mathcal{B}^\Sigma$. Consider a $(\Sigma_1 \cup \Sigma_2)$-structure $\mathcal{F}$ whose carrier is the carrier B of $\mathcal{B}$, and which interprets the function symbols of $\Sigma_1 \cup \Sigma_2$ as follows: for all $g \in \Sigma_1 \cup \Sigma_2$ of arity $n \geq 0$ and all $b_1, \ldots, b_n \in B$,

$$g^{\mathcal{F}}(b_1, \ldots, b_n) := \begin{cases} h(g^{\mathcal{A}}(h^{-1}(b_1), \ldots, h^{-1}(b_n))) & \text{if } \ g \in (\Sigma_1 \setminus \Sigma_2) \\ g^{\mathcal{B}}(b_1, \ldots, b_n) & \text{if } \ g \in \Sigma_2 \end{cases}$$

Intuitively, $\mathcal{F}$ interprets Σ_2-symbols as $\mathcal{B}$ does. For Σ_1-symbols that are not also Σ_2-symbols, the isomorphism h is used to transfer their interpretation from $\mathcal{A}$ to $\mathcal{B}$.

By construction of $\mathcal{F}$, it is immediate that $\mathcal{B}$ and $\mathcal{F}^{\Sigma_2}$ are isomorphic (with the identity mapping as isomorphism). We prove below that h is a Σ_1-isomorphism of $\mathcal{A}$ onto $\mathcal{F}^{\Sigma_1}$. It will then follow from Definition 18 that $\mathcal{F}$ is a fusion of $\mathcal{A}$ and $\mathcal{B}$.

Since we already know that h is a bijection, it remains to be shown that it is a Σ_1-homomorphism. If g is an n-ary function symbol of $\Sigma_1 \setminus \Sigma_2$ and $a_1, \ldots, a_n \in A$, then

$$\begin{aligned} h(g^{\mathcal{A}}(a_1, \ldots, a_n)) &= h(g^{\mathcal{A}}(h^{-1}(h(a_1), \ldots, h^{-1}(h(a_1)a_n)))) && \text{(by def. of inverse)} \\ &= g^{\mathcal{F}}(h(a_1), \ldots, h(a_n)) && \text{(by def. of } g^{\mathcal{F}}\text{).} \end{aligned}$$

If g is an n-ary function symbol of $\Sigma = \Sigma_1 \cap \Sigma_2$ and $a_1, \ldots, a_n \in A$, then

$$\begin{aligned} h(g^{\mathcal{A}}(a_1, \ldots, a_n)) &= g^{\mathcal{B}}(h(a_1), \ldots, h(a_n)) && \text{(since } h \text{ is a hom.)} \\ &= g^{\mathcal{F}}(h(a_1), \ldots, h(a_n)) && \text{(by def. of } g^{\mathcal{F}}\text{).} \end{aligned}$$

□

[11] We must point out, however, that $Fus(\mathcal{A}_1, \mathcal{A}_2)$ may contain non-isomorphic structures.

If $\mathcal{A}_1$ and $\mathcal{A}_2$ are two algebras and h is a isomorphism between their Σ-reducts (with Σ being the intersection of their signatures), we will call *canonical fusion* of $\mathcal{A}_1$ and $\mathcal{A}_2$ with respect to h the fusion of $\mathcal{A}_1$ and $\mathcal{A}_2$ constructed exactly as in the proof above.

Fusions of algebras have a close link with unions of theories, which we will exploit later.

Lemma 20 *If E_1, E_2 are two equational theories of signature Σ_1, Σ_2, respectively, and $\mathcal{F}$ is a fusion of a model of E_1 and a model of E_2, then $\mathcal{F}$ is a model of $E_1 \cup E_2$.*

Proof. By the definition of fusion it is immediate that $\mathcal{F}^{\Sigma_1}$ models every sentence in E_1 while $\mathcal{F}^{\Sigma_2}$ models every sentence in E_2; therefore, $\mathcal{F}$ models every sentence of $E_1 \cup E_2$. □

Conversely, it is also easy to see that every model of $E_1 \cup E_2$ is the fusion of a model of E_1 and a model of E_2 (see [TR98] for a proof).

In the presence of certain conditions, the test for satisfiability in a fusion of two algebras can be reduced to a "local" satisfiability test in each of the algebras. For $i = 1, 2$, consider a Σ_i-algebra $\mathcal{A}_i$ and an arbitrary Σ_i-formula φ_i, and let $\Sigma := \Sigma_1 \cap \Sigma_2$. Obviously, if we know that the joint formula $\varphi_1 \wedge \varphi_2$ is satisfiable in a fusion of $\mathcal{A}_1$ and $\mathcal{A}_2$, we also know that φ_1 is satisfiable in $\mathcal{A}_1$ and φ_2 is satisfiable in $\mathcal{A}_2$.

What about the converse? Assume we know that φ_1 is satisfiable in $\mathcal{A}_1$ and φ_2 is satisfiable in $\mathcal{A}_2$. Under which additional conditions can we conclude that $\varphi_1 \wedge \varphi_2$ is satisfiable in a fusion of $\mathcal{A}_1$ and $\mathcal{A}_2$? Intuitively, and speaking modulo isomorphism, we know from the definition of fusion that, for $i = 1, 2$, a valuation α_i of $\mathcal{V}ar(\varphi_i)$ that makes φ_i true in $\mathcal{A}_i$ will also make φ_i true in every element of $Fus(\mathcal{A}_1, \mathcal{A}_2)$. For $\varphi_1 \wedge \varphi_2$ to be satisfiable in an element of $Fus(\mathcal{A}_1, \mathcal{A}_2)$, however, it is necessary that α_1 and α_2 *agree* on the values they assign to the variables shared by φ_1 and φ_2. The problem is that, in general, we cannot tell whether there exist valuations α_1 and α_2 that agree on the shared variables. One case in which we can is described by the proposition below.

Proposition 21 *Let $\mathcal{A}_1$ be a Σ_1-algebra and and $\mathcal{A}_2$ be a Σ_2-algebra, and $\Sigma := \Sigma_1 \cap \Sigma_2$. Assume that their 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 fusion of $\mathcal{A}_1$ and $\mathcal{A}_2$ in which $\varphi_1 \wedge \varphi_2$ is satisfiable.*

Proof. Let $U := \mathcal{V}ar(\varphi_1) \cap \mathcal{V}ar(\varphi_2)$. Then for $i = 1, 2$, consider a valuation $\alpha_i : \mathcal{V}ar(\varphi_i) \longrightarrow A_i$ such that $\mathcal{A}_i, \alpha_i \models \varphi_i$, and whose restriction to U is an injection of U into Y_i. Where $\alpha_1(U)$ denotes the image of U under α_1, consider the map $f\colon \alpha_1(U) \longrightarrow Y_2$ such that

$$f(\alpha_1(v)) = \alpha_2(v) \text{ for all } v \in U.$$

Since f is injective by construction and Y_1 and Y_2 have the same cardinality, f can be extended to a bijection f' of Y_1 onto Y_2. Now, by assumption the algebras $\mathcal{A}_1{}^\Sigma$ and $\mathcal{A}_2{}^\Sigma$ are free in the same variety. By well-known results from Universal Algebra (see, e.g., [BN98], Theorem 3.3.3) then, f', which is a bijection between their sets of generators, can be extended to a Σ-isomorphism h between the two algebras. It follows by Proposition 19 that $\mathit{Fus}(\mathcal{A}_1, \mathcal{A}_2)$ is nonempty.

Let $\mathcal{F}$ be the canonical fusion of $\mathcal{A}_1$ and $\mathcal{A}_2$ with respect to h.[12] Then let $\alpha\colon \mathcal{V}ar(\varphi_1 \wedge \varphi_2) \longrightarrow F$ be a valuation such that

$$\alpha(v) := \begin{cases} h(\alpha_1(v)) & \text{if } v \in \mathcal{V}ar(\varphi_1) \\ \alpha_2(v) & \text{if } v \in \mathcal{V}ar(\varphi_2) \end{cases}$$

Notice that α is well-defined because by construction $h(\alpha_1(v)) = \alpha_2(v)$ for all $v \in \mathcal{V}ar(\varphi_1) \cap \mathcal{V}ar(\varphi_2)$. We show that $\mathcal{F}, \alpha \models \varphi_1 \wedge \varphi_2$, which will prove the claim.

We know from the proof of Proposition 19 that h is actually a Σ_1-isomorphism of $\mathcal{A}_1$ onto $\mathcal{F}^{\Sigma_1}$. Similarly, the identity map between the carriers of $\mathcal{A}_2$ and $\mathcal{F}$ is a Σ_2-isomorphism of $\mathcal{A}_2$ onto $\mathcal{F}^{\Sigma_2}$. From this it is easy to see that

$$\mathcal{F}^{\Sigma_1}, h \circ \alpha_1 \models \varphi_1 \quad \text{and} \quad \mathcal{F}^{\Sigma_2}, \alpha_2 \models \varphi_2.$$

From the definition of α (and of reduct) it follows immediately that $\mathcal{F}, \alpha \models \varphi_1$ and $\mathcal{F}, \alpha \models \varphi_2$. Therefore, $\mathcal{F}, \alpha \models \varphi_1 \wedge \varphi_2$. □

Notice that the proposition does not require that the whole algebras be free but just their *reducts* to the common signature. In the following, however, we will be interested in countably generated E_i-free Σ_i-algebras ($i = 1, 2$) whose reducts to the common signature $\Sigma := \Sigma_1 \cap \Sigma_2$ are free for the same variety, and over a countably infinite set of generators. In the next subsection then, we will first develop criteria that make sure that the reduct of a free algebra is again free.

5.2 Theories Admitting Constructors

In general, 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

$$E := \{x \equiv \mathsf{p}(\mathsf{s}(x)),\ x \equiv \mathsf{s}(\mathsf{p}(x))\}. \tag{1}$$

[12] Recall that, by construction, $\mathcal{F}$ and $\mathcal{A}_2$ have the same carrier.

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. In fact, any singleton set of integers is a set of free generators for $\mathcal{Z}$. The number zero, for instance, generates all the positive integers with the successor function and the negative ones with predecessor function. 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,[13] which is a necessary condition for an algebra to be free.

Nonetheless, there are free algebras some of whose reducts, although requiring a possibly larger set of generators, are still free. These algebras are models of equational theories that admit *constructors* in the sense explained below.

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 22 (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.*

Definition 22 is a rather abstract formulation of our requirements on the theory E. In the following, we develop a more concrete characterization of theories admitting constructors. In particular, this characterization 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. More precisely, every member of $T(\Sigma, G)$ is obtained from a term $s \in T(\Sigma, V)$ by replacing the variables of s with terms from G. To express this construction we will denote any such term by $s(\bar{r})$ where $\bar{r}$ is the tuple made, without repetitions, 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.

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

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

[13]A set of generators for an algebra $\mathcal{A}$ is *redundant* if one of its proper subsets is also a set of generators for $\mathcal{A}$.

In essence, $G_E(\Sigma, V)$ is made, modulo E equivalence, of Ω-terms whose top symbol is not in Σ. 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 prove the following properties of $G_E(\Sigma, V)$.

Lemma 23 *Let E be an equational theory over Ω and $\Sigma \subseteq \Omega$.*

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

Proof. Let $G := G_E(\Sigma, V)$.

(1) Let $r \in G$. Then, any $t \in T(\Omega, V)$ such that $t =_E r$ is an element of G. Otherwise, there would be a term $t' \in T(\Omega, V)$ such that $t' =_E t$ and $t'(\epsilon) \in \Sigma$. But then we would also have that $t' =_E r$, against the assumption that $r \in G$.

(2) Since V is assumed to be countably infinite, $V \subseteq G$ obviously implies that G is nonempty. We prove the other direction by proving its contrapositive. Assume that there exists a variable $v \in V \setminus G$. By definition of G then, there exists an $f \in \Sigma$ and a tuple $\bar{t}$ of Ω-terms such that $v =_E f(\bar{t})$. Now consider any $r \in T(\Omega, V)$. By applying the substitution $\{v \mapsto r\}$ to the equation $v \equiv f(\bar{t})$, we obtain a tuple of Ω-terms $\bar{t}'$ such that $r =_E f(\bar{t}')$, which means that $r \notin G$. From the generality of r it follows that G is empty.

(3) Again, we prove the contrapositive. Assume that E^Σ is not collapse-free. Then there exists a *non-variable* Σ-term s and a variable $v \in V$ such that $v =_{E^\Sigma} s$. By definition of G this implies that $v \notin G$, and thus $V \not\subseteq G$. □

Theorem 24 (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 $s_1(\bar{r}_1), s_2(\bar{r}_2) \in T(\Sigma, G)$,*
$$s_1(\bar{r}_1) =_E s_2(\bar{r}_2) \quad \textit{iff} \quad s_1(\bar{v}_1) =_E s_2(\bar{v}_2),$$
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.

Proof. Let $\mathcal{A}$ be an E-free Ω-algebra with the countably infinite set of generators X. Where α is any bijective valuation of V onto X,[14] let

$$Y \;:=\; \{[\![r]\!]^{\mathcal{A}}_{\alpha} \mid r \in G\}.$$

($\Leftarrow$) Assume that the three conditions in the formulation of the theorem are satisfied. We show that E^{Σ} is collapse-free and $\mathcal{A}^{\Sigma}$ is E^{Σ}-free with generators Y.

By Lemma 23(3), the assumption that $V \subseteq G$ implies that E^{Σ} is collapse-free.

To show that $\mathcal{A}^{\Sigma}$ is E^{Σ}-free we start by observing that, since $\mathcal{A}$ is a model of E, its reduct $\mathcal{A}^{\Sigma}$ is a model of E^{Σ}. Next, we show that $\mathcal{A}^{\Sigma}$ is generated by Y. In fact, let a be an element of A—which is also the carrier of $\mathcal{A}^{\Sigma}$. We know that as an Ω-algebra $\mathcal{A}$ is generated by X; thus there exists a term $t \in T(\Omega, V)$ such that $a = [\![t]\!]^{\mathcal{A}}_{\alpha}$. By condition (2), the term $t \in T(\Omega, V)$ is equivalent modulo E to a term $s(\bar{r}) \in T(\Sigma, G)$. Since $\mathcal{A}$ is a model of E, this implies that $a = [\![t]\!]^{\mathcal{A}}_{\alpha} = [\![s(\bar{r})]\!]^{\mathcal{A}}_{\alpha}$, from which it easily follows by definition of Y that a is Σ-generated by Y.

The above entails that $\mathcal{A}^{\Sigma}$ satisfies the first two conditions of Proposition 1. To show that it is E^{Σ}-free then it is enough to show that it also satisfies the third condition of the same proposition.

Thus, let $s_1(\bar{v}_1), s_2(\bar{v}_2) \in T(\Sigma, V)$ and assume that $\mathcal{A}^{\Sigma}, \alpha' \models s_1(\bar{v}_1) \equiv s_2(\bar{v}_2)$ for some injection α' of $V_0 := \mathcal{V}ar(s_1(\bar{v}_1) \equiv s_2(\bar{v}_2))$ into Y. By definition of Y we know that, for all $v \in V_0$, there is a term $r_v \in G$ such that $\alpha'(v) = [\![r_v]\!]^{\mathcal{A}}_{\alpha}$. Using these terms we can construct two tuples $\bar{r}_1$ and $\bar{r}_2$ of terms in G such that, for $i = 1, 2$, the term $s_i(\bar{r}_i)$ is obtained from $s_i(\bar{v}_i)$ by replacing each variable v in $\mathcal{V}ar(s_i(\bar{v}_i))$ by the term r_v, and $\mathcal{A}, \alpha \models s_1(\bar{r}_1) \equiv s_2(\bar{r}_2)$. Since $\mathcal{A}$ is E-free with generators X and α is injective as well we can conclude by Proposition 1(3) that $s_1(\bar{r}_1) =_E s_2(\bar{r}_2)$.

Since α' is injective, we know that $r_u \neq_E r_v$ for distinct variables $u, v \in V_0$. Thus, considered the other way round, the equation $s_1(\bar{v}_1) \equiv s_2(\bar{v}_2)$ can be obtained from $s_1(\bar{r}_1) \equiv s_2(\bar{r}_2)$ by abstracting the terms $\bar{r}_1, \bar{r}_2$ such that two terms are abstracted by the same variable iff they are equivalent modulo E. Thus, we can apply condition (3) to obtain $s_1(\bar{v}_1) =_E s_2(\bar{v}_2)$. Since the terms $s_1(\bar{v}_1), s_2(\bar{v}_2)$ are Σ-terms, this is the same as saying that $s_1(\bar{v}_1) =_{E^{\Sigma}} s_2(\bar{v}_2)$.

($\Rightarrow$) Now assume that Σ is a set of constructors for E, which implies that $\mathcal{A}^{\Sigma}$ is E^{Σ}-free for some set Z of generators. First, notice that Z cannot be the empty set. Otherwise, the Ω-algebra $\mathcal{A}$ would also be generated by the empty set, contradicting our assumption that the theory E is non-trivial. In fact, take an arbitrary element x from the (countably infinite) set of generators X of $\mathcal{A}$. If $\mathcal{A}$ is also generated by the empty set, then there exists a ground term s (i.e., a term without variables) such that $x = [\![s]\!]^{\mathcal{A}}_{\alpha}$.[15] Where $v \in V$ is such that $\alpha(v) = x$, the identity $x = [\![s]\!]^{\mathcal{A}}_{\alpha}$

[14] Such a valuation α exist since V is assumed to be countably infinite.

[15] Since s is a ground term, the value $[\![s]\!]^{\mathcal{A}}_{\alpha}$ does not depend on α.

entails that $\mathcal{A}, \alpha \models v \equiv s$, and so, by Proposition 1(3), that $v =_E s$. Since s does not contain v, this implies that any term is equivalent in E to s, i.e., E is trivial.

Next, we prove that $Z = Y$. Ad absurdum, assume that $Y \setminus Z$ is nonempty and let $y \in Y \setminus Z$. Since $\mathcal{A}$ is Ω-generated by X and $\mathcal{A}^\Sigma$ is Σ-generated by Z, we know that there exist a *non-variable* Σ-term s and a tuple $\bar{t}$ of Ω-terms such that $[\![t_i]\!]^{\mathcal{A}}_\alpha \in Z$ for all elements t_i of $\bar{t}$ and $y = [\![s(\bar{t})]\!]^{\mathcal{A}}_\alpha$. By definition of Y we know that there is a term $r \in G$ such that $y = [\![r]\!]^{\mathcal{A}}_\alpha$. As $\mathcal{A}$ is E-free and α is injective, we can then conclude by Proposition 1(3) that $r =_E s(\bar{t})$, but then r cannot be in G. It follows that $Y \subseteq Z$.

To show the other inclusion, consider a generator $z \in Z$. We prove below that $z \in Y$ and so $Z \subseteq Y$. Since $\mathcal{A}$ is Ω-generated by X, there exists an Ω-term r such that $z = [\![r]\!]^{\mathcal{A}}_\alpha$. If $r \notin G$, there exists a function symbol $f \in \Sigma$ and a tuple of Ω-terms $\bar{t}$ such that $r =_E f(\bar{t})$. Since the elements of the tuple $\bar{t}$ are all Σ-generated by Z, there is a variable v, a *non-variable* Σ-term s, and an injective mapping α' of $\mathcal{V}ar(s) \cup \{v\}$ into Z such that $\alpha'(v) = z = [\![s]\!]^{\mathcal{A}^\Sigma}_{\alpha'}$.[16] As $\mathcal{A}^\Sigma$ is E^Σ-free with generators Z, we obtain that $v =_{E^\Sigma} s$. But this contradicts the fact that E^Σ is collapse-free. It follows that $r \in G$, which implies that $z \in Y$ by definition of Y.

In conclusion, we have shown that Z is nonempty and coincides with $Y = \{[\![r]\!]^{\mathcal{A}}_\alpha \mid r \in G\}$. In particular, this means that G is nonempty either. The first condition in the formulation of the theorem then follows directly from Lemma 23(2). The second condition follows from the fact that $\mathcal{A}^\Sigma$ is Σ-generated by Z. Similarly, the third condition follows from Proposition 1(3). □

The proof of the theorem provides a little more information than stated in the formulation of the theorem.

Corollary 25 *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 $X \subseteq Y$.*

Notice that $X \subseteq Y$ is an immediate consequence of $V \subseteq G_E(\Sigma, V)$.

Condition 2 of Theorem 24 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, we will say that it is *E-irreducible*.

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

[16] Note that v may be an element of $\mathcal{V}ar(s)$.

Definition 26 (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*

$$NF_{\Sigma}^{E}\colon T(\Omega, V) \longrightarrow T(\Sigma, G)$$

such that $NF_{\Sigma}^{E}(t)$ is a normal form of t, i.e., $NF_{\Sigma}^{E}(t) =_{E} t$.

Notice that Definition 26 does not entail that the variables of $NF_{\Sigma}^{E}(t)$ are included in the variables of t. However, if $V_0 := \mathcal{V}ar(NF_{\Sigma}^{E}(t)) \setminus \mathcal{V}ar(t)$ is nonempty, then $\pi(NF_{\Sigma}^{E}(t))$ 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 $\mathcal{V}ar(NF_{\Sigma}^{E}(t)) \setminus \mathcal{V}ar(t)$ and V_1 are disjoint.[17] As a rule then we will *always* assume that the variables occurring in a normal form $NF_{\Sigma}^{E}(t)$ but not in t, if any, are *fresh* variables.

An important consequence of Definition 26—to which we will appeal in proving the termination of the extended combination procedure—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 27 *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.*

Proof. Observe that any $t \in T(\Omega, V)$ can be seen as having the form $s(\bar{r})$ where s is a Σ-term and $\bar{r}$ are terms whose top symbols are not in Σ. From the definition of normal form it is immediate that $s(\bar{r})$ is in normal form exactly when every components of $\bar{r}$ is in G. But being a member of G is a decidable property of Ω-terms: to test whether any $r \in T(\Omega, V)$ is in G, it is enough to compute $NF_{\Sigma}^{E}(r)$ and look at its top symbol. In fact,

$$r \in G \quad \text{iff} \quad NF_{\Sigma}^{E}(r)(\epsilon) \notin \Sigma.$$

To see that first notice that, by the definition of G, if $NF_{\Sigma}^{E}(r)$ starts with a Σ-symbol then $r \notin G$. Now, if $NF_{\Sigma}^{E}(r)$ does not start with a Σ-symbol, since it is a term in $T(\Sigma, G)$ it must be an element of G, r' say. But then, by definition of NF_{Σ}^{E}, r and r' are equivalent in E, which entails that $r \in G$ by Lemma 23(1). □

We provide below two examples of equational theories admitting constructors in the sense of Definition 22. But first, let us consider some immediate counter-examples:

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

- The signature $\Sigma := \Omega := \{f\}$ is not a set of constructors for the theory E axiomatized by $\{x \equiv f(x)\}$ because Definition 22(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 24(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 24 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 28 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:

$$\begin{aligned} x + (y + z) &\equiv (x + y) + z, \\ x + y &\equiv y + x, \\ x + \mathsf{s}(y) &\equiv \mathsf{s}(x + y), \\ x + 0 &\equiv x. \end{aligned} \tag{2}$$

The signature $\Sigma := \{0, \mathsf{s}\}$ is a set of constructors for E_1 in the sense of Definition 22. Instead of showing this directly, we prove that the three conditions of Theorem 24 are satisfied.

First observe that the first two equations of E_1 define associativity and commutativity of $+$. Let us call the theory axiomatized by these two equations AC. It is possible to show[18] that orienting the other equations in E_1 from left to right, one obtains a canonical term rewrite system R modulo AC. Here "modulo AC" means that, instead of syntactic matching, AC-matching is used when determining whether a rule is applicable. We denote the rewrite relation induced by R modulo AC by $\rightarrow_{R,AC}$. The normal form of a term t w.r.t. $\rightarrow_{R,AC}$ (i.e., the irreducible term reached by applying $\rightarrow_{R,AC}$ as long as possible starting with t), is unique only modulo AC. For any $t \in T(\Omega, V)$, we use $t{\downarrow}_{R,AC}$ to denote a normal form of t with respect $\rightarrow_{R,AC}$. Because R is canonical modulo AC, any term has a normal form (termination), and the normal forms of two E_1-equivalent terms are equivalent modulo AC, that is, $s =_{E_1} t$ iff $s{\downarrow}_{R,AC} =_{AC} t{\downarrow}_{R,AC}$ (Church-Rosser modulo AC).

We claim that $G_{E_1}(\Sigma, V)$ is the set of Ω-terms whose normal form w.r.t. $\rightarrow_{R,AC}$ not 0 and does not start with s.[19] In other words, $G := G_{E_1}(\Sigma, V)$ coincides with

[18]For example by employing a term rewriting laboratory like REVEAL.

[19]Note that this property is invariant under AC, i.e., whether it is satisfied or not does not depend on which representative of the AC-class of the normal form is taken.

the set

$$G' \;:=\; \{t \in T(\Omega, V) \mid t{\downarrow}_{R,AC}(\epsilon) \notin \Sigma\}.$$

From the fact that $t =_E t{\downarrow}_{R,AC}$, it follows immediately that $G \subseteq G'$. To prove that $G' \subseteq G$ we show that no term not in G is in G'.

If $t \notin G$, then t is equivalent in E_1 to a term t' where t' is either $\mathsf{0}$ or starts with s. Since t and t' have the same normal form (modulo AC) it is enough to show that $t'{\downarrow}_{R,AC}(\epsilon) \in \Sigma$. This is immediate if t' is $\mathsf{0}$ because $\mathsf{0}{\downarrow}_{R,AC}$ is obviously $\mathsf{0}$. If t' starts with s, notice that, since the left-hand sides of the rules in R do not start with s, there cannot be a rewrite at the top of t' (or any of its descendants). In addition, the equations from AC do not contain s at all. It follows that $t'{\downarrow}_{R,AC}$ starts with s as well. This completes the proof of $G' = G_{E_1}(\Sigma, V)$.

Now, it is immediate that $v \in G'$ for all variables $v \in V$ as $v{\downarrow}_{R,AC} = v$. Hence, Theorem 24(1) is satisfied by E_1 and Σ.

To see that Theorem 24(2) is satisfied it is enough to show that $t{\downarrow}_{R,AC} \in T(\Sigma, G')$ for all $t \in T(\Omega, V)$. This is immediate if $t{\downarrow}_{R,AC} \in T(\Sigma, \emptyset) \cup G'$. Otherwise is is easy to show that $t{\downarrow}_A C = \mathsf{s}^n(r)$ where $n \geq 1$ and r is not $\mathsf{0}$ and does not start with s. Since any subterm of an irreducible term is irreducible as well, we know that $r{\downarrow}_{R,AC} = r$. Thus, $r \in G'$ by definition of G', and so $t{\downarrow}_{R,AC} \in T(\Sigma, G')$.

To see that Theorem 24(3) is satisfied, first observe that (again a consequence of the fact that s does not occur at the top in the left-hand sides of the rewrite rules) $\mathsf{s}^n(r){\downarrow}_{R,AC} = \mathsf{s}^n(r{\downarrow}_{R,AC})$ for all $n \geq 0$ and terms r.

Now let $t_1, t_2 \in T(\Sigma, G')$ be such that $t_1 =_{E_1} t_2$. We know that each t_i has the form $\mathsf{s}^{n_i}(r_i)$ where $n_i \geq 0$ and r_i does not start with the symbol s. Since R is canonical modulo AC, $t_1 =_{E_1} t_2$ implies that $\mathsf{s}^{n_1}(r_2){\downarrow}_{R,AC} =_{AC} \mathsf{s}^{n_2}(r_2){\downarrow}_{R,AC}$. As seen before $\mathsf{s}^{n_i}(r_i){\downarrow}_{R,AC} = \mathsf{s}^{n_1}(r_i{\downarrow}_{R,AC})$ for $i = 1, 2$, and $r_i{\downarrow}_{R,AC}$ does not start with s. It follows that $n_1 = n_2$, and thus $r_1{\downarrow}_{R,AC} =_{AC} r_2{\downarrow}_{R,AC}$, which entails that $r_1 =_{E_1} r_2$. Abstracting then r_1 and r_2 by the same variable v in the equation $\mathsf{s}^{n_1}(r_1) \equiv \mathsf{s}^{n_1}(r_2)$ we obtain the equation $\mathsf{s}^{n_1}(v) \equiv \mathsf{s}^{n_1}(v)$, which is trivially valid in E_1.

Note that the restriction of E_1 to Σ (i.e., the theory ${E_1}^{\Sigma}$) is the syntactic equality of Σ-terms. This is an immediate consequence of the fact that the rules in R and the equations in AC cannot be applied to terms that do not contain $+$.

Example 29 Consider the signature $\Sigma_2 := \{\mathsf{0}, \mathsf{s}, \mathsf{mod2}\}$ and the equational theory E_2 axiomatized by the equations below:

$$\begin{aligned} \mathsf{mod2}(\mathsf{0}) &\equiv \mathsf{0}, \\ \mathsf{mod2}(\mathsf{s}(\mathsf{0})) &\equiv \mathsf{s}(\mathsf{0}), \\ \mathsf{mod2}(\mathsf{s}(\mathsf{s}(x))) &\equiv \mathsf{mod2}(x), \\ \mathsf{mod2}(\mathsf{mod2}(x)) &\equiv \mathsf{mod2}(x). \end{aligned} \tag{3}$$

The signature $\Sigma := \{\mathsf{0}, \mathsf{s}\}$ is a set of constructors for E_2 in the sense of Definition 22. As in the previous example we can show that the three conditions of Theorem 24. Here we can use the fact that orienting the equations from left to right yields a canonical term rewriting system.[20] As in the previous example, the restriction of E_2 to Σ (i.e., the theory ${E_2}^{\Sigma}$) is the syntactic equality of Σ-terms.

The next example differs from the previous ones in that the restriction of the theory to the constructor signature is no longer syntactic equality.

Example 30 Consider the signature $\Sigma_3 := \{\mathsf{0}, \mathsf{1}, \mathsf{rev}, \cdot\}$ and the equational theory E_3 axiomatized by the equations below:

$$\begin{array}{rcl} x \cdot (y \cdot z) & \equiv & (x \cdot y) \cdot z, \\ \mathsf{rev}(\mathsf{0}) & \equiv & \mathsf{0}, \\ \mathsf{rev}(\mathsf{1}) & \equiv & \mathsf{1}, \\ \mathsf{rev}(x \cdot y) & \equiv & \mathsf{rev}(y) \cdot \mathsf{rev}(x), \\ \mathsf{rev}(\mathsf{rev}(x)) & \equiv & x. \end{array} \tag{4}$$

Note that orienting the equations from left to right yields a canonical term rewriting system R_3. We denote the normal form of a term t w.r.t. this rewrite system by $t{\downarrow_{R_3}}$.

We claim that the signature $\Sigma' := \{\mathsf{0}, \mathsf{1}, \cdot\}$ is a set of constructors for E_3 in the sense of Definition 22. Again, we prove this by showing that the three conditions of Theorem 24 are satisfied.

First, we show that $G := G_E(\Sigma', V)$ is equal to

$$G' \ := \ \{\mathsf{rev}^k(v) \mid v \in V \text{ and } k \geq 0\}.$$

Assume that $s \notin G$. Then s is E_3-equivalent to $\mathsf{0}$ or $\mathsf{1}$, or it is of the form $s_1 \cdot s_2$. If we analyze the rules in R_3, and take into account that the R_3-normal forms of E_3-equivalent terms are equal, then we see that the R_3-normal form of s is $\mathsf{0}$ or $\mathsf{1}$, or has top symbol $\cdot$. Since the R_3-normal form of any term in G' is either v or $\mathsf{rev}(v)$, this shows that $s \notin G'$, and thus $G' \subseteq G$.

Conversely, assume that $s \in G \setminus G'$, and let s be minimal with this property. Since $s(\epsilon) \in \Sigma'$ would contradict our assumption that $s \in G$, we know that $s = \mathsf{rev}(s')$ for a term s'. Obviously, $s' \notin G'$ since otherwise $s \in G'$ as well. In addition, since s was assumed to be a minimal term in $G \setminus G'$, we know that $s' \in G$. However, this means that $s' =_{E_1} t$ for a term t with $t(\epsilon) \in \Sigma'$. But then the rules of R_3 can be applied to $\mathsf{rev}(t)$ such that the resulting term has its top symbol in Σ' as well. Since

[20] Termination should be clear and confluence can easily be checked by computing all critical pairs.

$s =_{E_3} \mathsf{rev}(t)$, this contradicts our assumption that $s \in G$. Thus, we have shown that $G \subseteq G'$.

Now it is immediate from the definition of G' that $V \subseteq G'$, and thus Theorem 24(1) is satisfied by E_3 and Σ'.

To see that Theorem 24(2) is satisfied, it is sufficient to show that the R_3-normal form of any term $t \in T(\Sigma_3, V)$ is of the form

$$t{\downarrow_{R_3}} = (\cdots((r_1 \cdot r_2) \cdot r_3) \cdot \;\cdots\; \cdot r_k)$$

where $r_i \in \{0,1\} \cup V \cup \{\mathsf{rev}(v) \mid v \in V\}$. This can easily be proved by showing that, to any term not in this form, one of the rules of R_3 applies.

To see that Theorem 24(3) holds, we consider a term $s(\bar{r}) \in T(\Sigma', G)$, that is $s(\bar{v})$ is a Σ'-term and any r in the tuple $\bar{r}$ belongs to G. It is easy to see that the R_3-normal form of $s(\bar{r})$ can be obtained by computing the normal form of $s(\bar{v})$ w.r.t. the rule $x \cdot (y \cdot z) \to (x \cdot y) \cdot z$, and then inserting into this term the normal forms of the terms in $\bar{r}$ w.r.t. the rule $\mathsf{rev}(\mathsf{rev}(x)) \to x$. Now, Theorem 24(3) is an easy consequence of this fact.

5.3 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 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 theories E_1, E_2 introduced in the above examples satisfy these conditions. In fact, we have already seen that Σ is a set of constructors for E_1 and for E_2, and ${E_1}^{\Sigma} = {E_2}^{\Sigma}$ is syntactic equality of Σ-terms.

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,[21] 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.

Proposition 31 *Let φ_1, φ_2 be two arbitrary first-order formulas 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 $E_1 \cup E_2$.*

[21] $\mathcal{A}_i$ exists because E_i is non-trivial.

Proof. Let $E_0 := {E_1}^\Sigma$ $(= {E_2}^\Sigma)$. By Corollary 25, ${\mathcal{A}_i}^\Sigma$ is E_0-free with generators Y_i for $i = 1, 2$. Moreover, Y_1 and Y_2 have the same cardinality because, for $i = 1, 2$, $X_i \subseteq Y_i \subseteq A_i$ by construction of Y_i and X_i and A_i are countably infinite by assumption. By Proposition 21 then $\varphi_1 \wedge \varphi_2$ is satisfiable in a fusion of $\mathcal{A}_1$ and $\mathcal{A}_2$, which is a model of $E_1 \cup E_2$ by Lemma 20. □

Again, note that φ_1 and φ_2 in the proposition above are arbitrary formulae. Therefore, if we take both of them to be the disequation $x \not\equiv y$ we immediately obtain the following corollary.

Corollary 32 *If E_1 and E_2 satisfy the above assumptions, then $E_1 \cup E_2$ is non-trivial.*

In the following we will show that, under the assumptions on E_1, E_2 stated at the beginning of this subsection, the signature $\Sigma := \Sigma_1 \cap \Sigma_2$ is also a set of constructors for $E := E_1 \cup E_2$, and $E^\Sigma = {E_1}^\Sigma = {E_2}^\Sigma$. In addition, if the word problem for E_i is decidable and normal forms are computable for Σ and E_i $(i = 1, 2)$, then normal forms are also computable for Σ and E.

We start by showing that E is a conservative extension of both E_1 and E_2.

Proposition 33 *For all $j \in \{1, 2\}$ and $t_1, t_2 \in T(\Sigma_j, V)$*

$$t_1 =_{E_j} t_2 \quad \textit{iff} \quad t_1 =_E t_2.$$

Proof. The implication from left to right is immediate since $E_j \subseteq E$. For the converse, assume that $j = 2$ (the proof for $j = 1$ is symmetrical), and let $t_1, t_2 \in T(\Sigma_2, V)$ such that $t_1 =_E t_2$.

Then, for $i = 1, 2$, let $\mathcal{A}_i$ be the E_i-free algebra with the countably infinite set of generators X_i. In the proof of Proposition 31 we have already seen that (under the given assumptions) ${\mathcal{A}_1}^\Sigma$ and ${\mathcal{A}_2}^\Sigma$ are isomorphic. Consider the canonical fusion $\mathcal{F}$ of $\mathcal{A}_1$ and $\mathcal{A}_2$ w.r.t. some isomorphism h of ${\mathcal{A}_1}^\Sigma$ onto ${\mathcal{A}_2}^\Sigma$, and recall that $\mathcal{F}^{\Sigma_2} = \mathcal{A}_2$.

Now, since $t_1 =_E t_2$ and $\mathcal{F}$ is a model of E, we have that $\mathcal{F}, \alpha \models t_1 \equiv t_2$ for any valuation α of $\mathcal{V}ar(t_1 \equiv t_2)$ into F $(= A_2)$. In particular, we can choose α to be an injection into X_2. Observing that t_1, t_2 are Σ_2-terms we then have that $\mathcal{A}_2, \alpha \models t_1 \equiv t_2$. It follows by Proposition 1 that $t_1 =_{E_2} t_2$. □

From the above result it is almost trivial to show the following.

Corollary 34 $E^\Sigma = {E_1}^\Sigma = {E_2}^\Sigma$.

To show that Σ is a set of constructors for $E_1 \cup E_2$, we will show that the three conditions in Theorem 24 are satisfied. Before we can do that, we need an appropriate characterization of $G_E(\Sigma, V)$. We will show that, modulo E, this set is identical to the set G' defined below.

Definition 35 *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 an element of G' cannot have a shared symbol (i.e., a symbol in Σ) as top symbol since it is a variable or it "starts" with an element of G_i.

Lemma 36 *For all $t \in T(\Sigma_1 \cup \Sigma_2, V)$, there exists a term $s(\bar{r}) \in T(\Sigma, G')$ such that $t =_E s(\bar{r})$, and this term can be effectively computed from t.*

Proof. In order to show that $s(\bar{r})$ is computable, we will need to know that the word problem for E is decidable. In the next section we will show that, under the assumptions on E_i made above, this is in fact the case (see Theorem 51). We prove the claim by term induction.

(Base case) If $t \in V$ then $t \in G'$, and thus the claim is trivially true.

(Induction step) Let $t \in T(\Sigma_1 \cup \Sigma_2, V) \setminus V$. If $t \in T(\Sigma_i, V)$ for $i = 1$ or $i = 2$, then we can simply compute the normal for Σ and E_i, which does the job since $G_i \subseteq G'$.

Otherwise, t has the form $t_1(\bar{t})$, where $t_1 \in T(\Sigma_i, V) \setminus V$ for $i = 1$ or $i = 2$, and $\bar{t}$ is a tuple with at least one nonvariable term and such that the top symbol of no term in $\bar{t}$ is in Σ_i. For simplicity, let us assume that $\bar{t}$ has length 1 and $i = 1$. (The proof for the general case is an easy generalization of what follows.)

Therefore, let $\bar{t} = t_2$ and so $t = t_1(t_2)$. We know that t_2 is a nonvariable term with top symbol in $\Sigma_2 \setminus \Sigma_1$. By the induction hypothesis there is a term $t_3 \in T(\Sigma, G')$ effectively computable from t_2 and such that $t_2 =_E t_3$. This means that $t =_E t_1(t_3)$. If t_3 is a variable then $t_1(t_3) \in T(\Sigma_1, V)$ and so we can prove the claim as before.

Therefore, assume that t_3 is not a variable (and is not equivalent to one in E).[22] Since $t_3 \in T(\Sigma, G')$, there exists a Σ-term $s_3(\bar{u}_3)$ and a tuple $\bar{r}_3$ of elements of G' such that $t_3 = s_3(\bar{r}_3)$.

For components r of $\bar{r}_3$ we distinguish *three cases.* If r is a variable, then $r \in G'$ and $r(\epsilon) \notin \Sigma_1$. If $r(\epsilon) \in \Sigma_2$, then $r \in G'$ also implies $r(\epsilon) \notin \Sigma_1$ by definition of G'. If $r(\epsilon) \in \Sigma_1$, then $r \in G'$ and the definition of G' imply that r is of the form $\widehat{r}(\bar{r}_4)$ where $\widehat{r}(\bar{u}_4)$ is a Σ_1-term and the components of $\bar{r}_4$ are elements of G' whose top symbol does not belong to Σ_1. From this it is clear that the decomposition of r into the form $\widehat{r}(\bar{r}_4)$ can be effectively computed.

As an easy consequence of the above case distinction we can represent t_3 in the form $t_3 =_E t_4(\bar{r})$ where $t_4(\bar{u})$ is a Σ_1-term and the components of $\bar{r}$ are elements of G' whose top symbol does not belong to Σ_1. In addition, since the word problem for E is decidable and E is nontrivial, we can assume without loss of generality that different positions in the tuple $\bar{r}$ are occupied by terms that are not equivalent in E, and that a nonvariable component of $\bar{r}$ is not equivalent in E to a variable.[23] To sum up, we have that

$$t \;=_E\; t_1(t_4(\bar{u}))[\bar{u}/\bar{r}],$$

where $t_1(t_4(\bar{u}))$ is a Σ_1-term, and each component of $\bar{r}$ is an element of G' whose top symbol does not belong to Σ_1.

By our assumption on E_1, since $t_1(t_4(\bar{u}))$ is a Σ_1-term, it is possible to compute a normal form of it for Σ and E_1. This normal form is a term $s_1(\bar{r}_1) \in T(\Sigma, G_1)$ such that $t_1(t_4(\bar{u})) =_{E_1} s_1(\bar{r}_1)$. Furthermore, we can assume without loss of generality that: (a) all variables occurring in $\bar{r}_1$ but not in $\bar{u}$ are fresh; and (b) if $r =_E v$ for a variable v and a component r of $\bar{r}_1$, then $r = v$.[24]

From the fact that $E_1 \subseteq E$ it follows that

$$t \;=_E\; t_1(t_4(\bar{u}))[\bar{u}/\bar{r}] \;=_E\; s_1(\bar{r}_1)[\bar{u}/\bar{r}].$$

Because of the way $s_1(\bar{r}_1)[\bar{u}/\bar{r}]$ was constructed, it is immediate that this term is computable from t. To complete the proof of the lemma, it remains to show that $s_1(\bar{r}_1)[\bar{u}/\bar{r}] \in T(\Sigma, G')$. To do that it is enough to show that $r[\bar{u}/\bar{r}] \in G'$ for each component r of $\bar{r}_1$.

If r is a variable not occurring in $\bar{u}$, the claim is obvious because $r[\bar{u}/\bar{r}] = r \in V$ and $V \subseteq G'$. If r is a variable in $\bar{u}$, then $r[\bar{u}/\bar{r}]$ is a component of $\bar{r}$ and so an element of G' by the above.

[22]The second assumption is without loss of generality since E is nontrivial and the word problem for E is decidable.

[23]Because of this assumption we need not have syntactic equality between t_3 and $t_4(\bar{r})$.

[24]Recall that $V \subseteq G_1$, and that the word problem for E is decidable.

Otherwise, r has the form $r(\bar{z})$ where $\bar{z}$ are the variables of r with no repetitions. Observe that assumption (b) above entails that $r(\bar{z}) \neq_E v$ for all $v \in V$, and thus Definition 35(2a) is satisfied for $r(\bar{z})$. Now let $\bar{r}' := \bar{z}[\bar{u}/\bar{r}]$. It is easy to see that each component of $\bar{r}'$ is an element of G' satisfying Definition 35(2b) (for $i = 1$). In fact, each component of $\bar{r}'$ is either a component of $\bar{r}$ or a variable. To see that Definition 35(2e) is satisfied, recall that, first, the tuple $\bar{r}$ satisfies this property, and the possible additional variables in $\bar{r}'$ (i.e., the variables in $\bar{z}$ that are not contained in $\bar{u}$) were assumed to be fresh. It is easy to see that the other conditions of Definition 35(2) are satisfied as well. It follows that $r[\bar{u}/\bar{r}] = r(\bar{z})[\bar{z}/\bar{r}'] \in G'$. □

Lemma 37 *For all $t \in G_E(\Sigma, V)$ there exists $r \in G'$ such that $t =_E r$.*

Proof. By the previous lemma, there exists $s(\bar{r}) \in T(\Sigma, G')$ such that $t =_E s(\bar{r})$. The definition of $G_E(\Sigma, V)$ implies that s cannot be a nonvariable term, and thus $s(\bar{r}) = r$ for some $r \in G'$. □

Lemma 38 $G' \subseteq G_E(\Sigma, V)$.

Proof. 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

$$Y_i \;:=\; \{[\![r]\!]^{\mathcal{A}_i}_{\alpha_i} \mid r \in G_{E_i}(\Sigma, V)\}$$

where α_i is any bijective valuation of V onto X_i. By Corollary 25, ${\mathcal{A}_i}^{\Sigma}$ is ${E_i}^{\Sigma}$-free with generators Y_i and $X_i \subseteq Y_i$.

Now let $Z_i := Y_i \setminus X_i$ and let $\{X_{1,1}, X_{1,2}\}$ be a partition of X_1 such that $X_{1,1}$ is countably infinite and $Card(X_{1,2}) = Card(Z_2)$.[25] Similarly, let $\{X_{2,1}, X_{2,2}\}$ be a partition of X_2 such that $Card(X_{2,2}) = Card(Z_1)$ and $X_{2,1}$ is countably infinite. Then consider 3 arbitrary bijections

$$h_1 \colon Z_1 \longrightarrow X_{2,2}, \quad h_2 \colon X_{1,1} \longrightarrow X_{2,1}, \quad h_3 \colon X_{1,2} \longrightarrow Z_2.$$

Observing that $\{Z_i, X_{i,1}, X_{i,2}\}$ is a partition of Y_i for each i, it is immediate that $h_1 \cup h_2 \cup h_3$ is a (well-defined) bijection of Y_1 onto Y_2.

Since ${E_1}^{\Sigma} = {E_2}^{\Sigma}$ by Corollary 34, ${\mathcal{A}_1}^{\Sigma}$ and ${\mathcal{A}_2}^{\Sigma}$ are free in the same variety with sets of generators of the same cardinality. As we have seen in the proof of Proposition 21, the bijection $h_1 \cup h_2 \cup h_3$ can be extended to a Σ-isomorphism h of ${\mathcal{A}_1}^{\Sigma}$ onto ${\mathcal{A}_2}^{\Sigma}$.

Let $\mathcal{F}$ be the canonical fusion of $\mathcal{A}_1, \mathcal{A}_2$ w.r.t. h as constructed in the proof of Proposition 19. Recall that $\mathcal{F}$ is a model of E and $\mathcal{A}^{\Sigma_2} = \mathcal{A}_2$, and let α be

[25] This is possible because Z_2 is countable (possibly finite).

an arbitrary bijective valuation of V onto $X_{2,1}$. We will prove later that for all $r \in G' \setminus V$,

$$[\![r]\!]_\alpha^{\mathcal{F}} \in Z_2 \text{ if } r(\epsilon) \in \Sigma_2 \quad \text{and} \quad [\![r]\!]_\alpha^{\mathcal{F}} \in X_{2,2} \text{ if } r(\epsilon) \in \Sigma_1 \tag{5}$$

which entails that $[\![r]\!]_\alpha^{\mathcal{F}} \in Y_2$ for all $r \in G'$.

To prove the lemma's claim now let $r \in G'$ and assume by contradiction that $r \notin G_E(\Sigma, V)$. Then, by the definition of $G_E(\Sigma, V)$ and Lemma 36, there is a term $s(\bar{r}) \in T(\Sigma, G')$ with s nonvariable such that $r =_E s(\bar{r})$. In fact, since $r \notin G_E(\Sigma, V)$ we know by definition of $G_E(\Sigma, V)$ that there is a term $f(\bar{t})$ with $f \in \Sigma$ such that $r =_E f(\bar{t})$. By Lemma 36 we can assume that each term in $\bar{t}$ is in $T(\Sigma, G')$. It follows that the term $f(\bar{t})$ is in $T(\Sigma, G')$ as well. Obviously, this term has the form $s(\bar{r})$ with *s nonvariable*.

Since $\mathcal{F}$ is a model of E, we then have that $\mathcal{F}, \alpha \models r \equiv s(\bar{r})$. Let, $v, \bar{v}$ be fresh variables abstracting $r, \bar{r}$ in $r \equiv s(\bar{r})$ so that terms equivalent in E are replaced by the same variable. Since we know that $[\![r]\!]_\alpha^{\mathcal{F}} \in Y_2$ for all $r \in G'$, it is clear that there exists an injective valuation β of $v, \bar{v}$ into Y_2 such that

$$\mathcal{F}, \beta \models v \equiv s(\bar{v}).$$

Since $v \equiv s(\bar{v})$ is a Σ-equation and $\mathcal{F}^\Sigma$ is ${E_2}^\Sigma$-free with generators Y_2, this entails by Proposition 1 that $v =_{{E_2}^\Sigma} s(\bar{v})$. However, this is impossible because ${E_2}^\Sigma$ is collapse-free by assumption. From the generality of r it follows that $G' \subseteq G_E(\Sigma, V)$.

We are left with proving that (5) above holds. We will do this by term induction.

(Base case) Assume that $r \in G_{E_2}(\Sigma, V) \setminus V$. First, we show that $[\![r]\!]_\alpha^{\mathcal{F}} \in Y_2$. Since α is a bijective valuation of V onto $X_{2,1}$, α_2 is a bijective valuation of V onto X_2, and $X_{2,1} \subseteq X_2$, there is a term r' obtained by a bijective renaming of the variables in r such that $[\![r]\!]_\alpha^{\mathcal{F}} = [\![r']\!]_{\alpha_2}^{\mathcal{A}_2}$. It is easy to see that $r \in G_{E_2}(\Sigma, V)$ implies $r' \in G_{E_2}(\Sigma, V)$, and thus $[\![r']\!]_{\alpha_2}^{\mathcal{A}_2} \in Y_2$ by definition of Y_2. We prove by contradiction that $[\![r]\!]_\alpha^{\mathcal{F}} \notin X_2$. In fact, if $[\![r]\!]_\alpha^{\mathcal{F}} \in X_2$, it is easy to show that there is a $v \in V$ and an injective valuation β of $\mathcal{V}ar(v \equiv r)$ into X_2 such that $\mathcal{F}, \beta \models v \equiv r$. Recalling that $\mathcal{F}^{\Sigma_2}$ is E_2-free with generators X_2 we then obtain by Proposition 1 that $v =_{E_2} r$, which contradicts the fact that $v \neq_E r$ by construction of G' (see Definition 35(2a)). It follows that $[\![r]\!]_\alpha^{\mathcal{F}} \in Z_2 = Y_2 \setminus X_2$.

If $r(\bar{v}) \in G_{E_1}(\Sigma, V) \setminus V$, let $\bar{b}$ be the tuple of values that α assigns, in order, to the variables in $\bar{v}$. By construction of $\mathcal{F}$, we know that[26]

$$[\![r]\!]_\alpha^{\mathcal{F}} = r^{\mathcal{F}}(\bar{b}) = h(r^{\mathcal{A}_1}(h^{-1}(\bar{b}))).$$

Since $\bar{b}$ contains no repetitions and is included in $X_{2,1}$, we have by construction of h that $h^{-1}(\bar{b})$ contains no repetitions and is included in $X_{1,1}$. As we did in the previous

[26]The expression $h^{-1}(\bar{b})$ below denotes the tuple obtained from $\bar{b}$ by replacing each element b of $\bar{b}$ by $h^{-1}(b)$.

case then, we can prove that $r^{\mathcal{A}_1}(h^{-1}(\bar{b})) \in Z_1$. By construction of h again this then implies that $[\![r]\!]^{\mathcal{F}}_{\alpha} = h(r^{\mathcal{A}_1}(h^{-1}(\bar{b}))) \in X_{2,2}$.

(Induction step) If $t \in G' \setminus (G_{E_1}(\Sigma, V) \cup G_{E_2}(\Sigma, V))$, then t has the form

$$r(\bar{v}, \bar{r})$$

where $r \in G_{E_i}(\Sigma, V) \setminus V$ with $i \in \{1, 2\}$, $\bar{v} \subseteq V$, $\bar{r} \subseteq G' \setminus V$, $\bar{r}$ is nonempty and $r'(\epsilon) \notin \Sigma_i$ for all $r' \in \bar{r}$. Let $\bar{b}$ be the tuple of values that α assigns, in order, to the variables in $\bar{v}$ and $\bar{c}$ the tuple made, in order, of all the elements $[\![r']\!]^{\mathcal{F}}_{\alpha}$ with $r' \in \bar{r}$.

If $i = 2$, then $\bar{b} \subseteq X_{2,1}$ by definition of α and $\bar{c} \subseteq X_{2,2}$ by induction hypothesis. It is immediate that $\bar{b}$ contains no repetitions and has no elements in common with $\bar{c}$. We claim that $\bar{c}$ contains no repetitions either. In fact, assume that $[\![r_1]\!]^{\mathcal{F}}_{\alpha} = [\![r_2]\!]^{\mathcal{F}}_{\alpha}$ for two distinct $r_1, r_2 \in \bar{r}$. Then, $\mathcal{F}, \alpha \models r_1 \equiv r_2$, which implies, again by Proposition 1, that $r_1 =_{E_2} r_2$. This contradicts the fact that the tuple $\bar{r}$ must satisfy Definition 35(2e). Given these facts, it is easy to show (as in the base case) that $[\![r(\bar{v}, \bar{r})]\!]^{\mathcal{F}}_{\alpha} \in Z_2$.

If $i = 1$, by construction of $\mathcal{F}$ we know that

$$r^{\mathcal{F}}(\bar{b}, \bar{c}) = h(r^{\mathcal{A}_1}(h^{-1}(\bar{b}), h^{-1}(\bar{c}))).$$

Now, observe that $\bar{b} \subseteq X_{2,1}$ by definition of α and $\bar{c} \subseteq Z_2$ by induction hypothesis. It follows by construction of h that $h^{-1}(\bar{b}) \subseteq X_{1,1}$ and $h^{-1}(\bar{c}) \subseteq X_{1,2}$. Observing that $\bar{b}$ and $\bar{c}$ do not contain repetitions (and have no common elements) we can prove (as in the case $i = 2$ before) that $r^{\mathcal{A}_1}(h^{-1}(\bar{b}), h^{-1}(\bar{c})) \in Z_1$. By construction of h again, we then finally have that $r^{\mathcal{F}}(\bar{b}, \bar{c}) \in X_{2,2}$ which means that $[\![r]\!]^{\mathcal{F}}_{\alpha} \in X_2$. □

Theorem 39 *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 $E_1 \cup E_2$.*
2. *$E^{\Sigma} = {E_1}^{\Sigma} = {E_2}^{\Sigma}$.*
3. *Normal forms are computable for Σ and E.*

Proof. Point 2 is immediate by Corollary 34. Point 3 is an easy consequence of Lemma 36 and 38. We prove point 1 by showing that E and Σ satisfy Theorem 24.

Now, Theorem 24(1) and (2) are an immediate consequence of the definition of G' and Lemma 36 and 38. To prove Theorem 24(3) we will use the algebra $\mathcal{F}$ and the valuation α defined in the proof of Lemma 38.

Let $s_1(\bar{r}_1), s_2(\bar{r}_2)$ be terms in $T(\Sigma, G_E(\Sigma, V))$ and $s_1(\bar{v}_1), s_2(\bar{v}_2)$ the terms obtained from them by abstracting E-equivalent elements in $\bar{r}_1, \bar{r}_2$ with the same variable. It is immediate that $s_1(\bar{v}_1) =_E s_2(\bar{v}_2)$ implies $s_1(\bar{r}_1) =_E s_2(\bar{r}_2)$.

Therefore, assume that $s_1(\bar{r}_1) =_E s_2(\bar{r}_2)$. By Lemmas 37 we can assume with no loss of generality that $s_1(\bar{r}_1), s_2(\bar{r}_2) \in T(\Sigma, G')$. Now, since $\mathcal{F}$ is a model of E, $s_1(\bar{r}_1) =_E s_2(\bar{r}_2)$ entails that

$$\mathcal{F}, \alpha \models s_1(\bar{r}_1) \equiv s_2(\bar{r}_2).$$

Recall that $\mathcal{F}^\Sigma$ is E^Σ-free with generators Y_2 and $[\![r]\!]_\alpha^\mathcal{F} \in Y_2$ for all elements of G'. From this it is easy to see that there is an injective valuation β of $\bar{v}_1 \cup \bar{v}_2$ into the generators of $\mathcal{F}^\Sigma$ such that $\mathcal{F}^\Sigma, \beta \models s_1(\bar{v}_1) \equiv s_2(\bar{v}_2)$. It follows by Proposition 1 that $s_1(\bar{v}_1) =_{E^\Sigma} s_2(\bar{v}_2)$, which implies immediately that $s_1(\bar{v}_1) =_E s_2(\bar{v}_2)$. □

6 An Extended Combination Procedure

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 (countable) signature Σ_i;
- $\Sigma := \Sigma_1 \cap \Sigma_2$ is a set of constructors for E_i;
- the word problem for E_i is decidable;
- normal forms are computable for Σ and E_i.

For now, we do not assume that ${E_1}^\Sigma = {E_2}^\Sigma$, as we did in the previous subsection. As we will see, such a restriction is not required to show the termination and soundness properties of the extended combination procedure. It will be used only to prove the procedure's completeness.

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 $G_{E_i}(\Sigma, V) \setminus V$.

In Section 4, we have introduced the notion of an abstraction system to prove some properties of the combination procedure. Now we will view the elements of the abstraction system (i.e., the equations and the initial disequation) as nodes of a graph whose edges are induced by the relation $\prec$.

6.1 Abstraction Systems as Directed Acyclic Graphs

Consider an abstraction system A as defined in Section 4. Such a system induces a graph $G_A := (A, \prec)$ whose set of *nodes* is A and whose set of *edges* consists of all pairs $(a_1, a_2) \in A \times A$ such that $a_1 \prec a_2$. According to Definition 8, G_A is in fact a directed acyclic graph (or *dag*).[27]

Assuming the standard definition of path between two nodes and of length of a path in a dag, we define below a notion of *height* of a node, which measures the longest possible path from a "root" of the graph to the node. This notion will be used in the definition of our combination procedure, and it will be important for the termination proof.

Definition 40 (Node Height) *Let $G := (\mathsf{N}, \mathsf{E})$ be a dag with finite sets of nodes and edges. A node $a \in \mathsf{N}$ is a* root *of G iff there is no $a' \in \mathsf{N}$ such that $(a', a) \in \mathsf{E}$.*[28] *The function $\mathsf{h} : \mathsf{N} \longrightarrow \mathbb{N}$ is defined as follows. For all $a \in \mathsf{N}$,*

- $\mathsf{h}(a) = 0$, *if a is a root of G;*
- $\mathsf{h}(a)$ *equals the maximum of the lengths of all the paths from the roots of G to a, otherwise.*[29]

Later, we will appeal to the following easily provable facts about the height function introduced above.

Lemma 41 *The following holds for every finite dag G and associated height function h.*

1. *For all nodes a, b of G, if there is a non-empty path from a to b then $\mathsf{h}(a) < \mathsf{h}(b)$.*
2. *Adding an edge from a node of G to another of greater height does not change the height of any node of G.*
3. *Removing an edge in G does not increase the height of any node of G (although it may decrease the height of some).*
4. *Removing a node and relative edges from G does not increase the height of the remaining nodes (although it may decrease the height of some).*

[27]Observe that G_A need not be a tree or even be connected.
[28]Because of the acyclicity condition, any finite dag has at least one root.
[29]This maximum exists because G is finite and acyclic.

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

1. Let $S := AS(t_1 \not\equiv t_2)$.
2. Repeatedly apply (in any order) **Coll1**, **Coll2**, **NIdent**, **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.

Figure 3: The Extended Combination Procedure.

We say that an equation of an abstraction system A is *reducible* iff one of its sides is E_i-reducible for $i = 1$ or $i = 2$. The disequation in A is always irreducible.

Definition 42 (Node Reducibility) *Let $(A, \prec)$ be the dag induced by the abstraction system A and let $a \in A$. We say that the* reducibility *of a is 1, and write $\mathsf{r}_A(a) = 1$, if a is reducible; we say that it is 0, and write $\mathsf{r}_A(a) = 0$, otherwise.*

6.2 The Extended Combination Procedure

The combination procedure described in Fig. 3 is an extension of the combination procedure introduced in Section 4. We have added two new derivation rules, **Shar1** and **Shar2**, and have modified the rule **Ident** (see Fig. 4). Notice that neither **Shar1** nor **Shar2** applies if Σ_1 and Σ_2 do not share function symbols.

The only difference between the rules **Ident** and **NIdent** is that the condition "$(y \equiv t) \not\prec^* (x \equiv s)$" is replaced by "$x \neq y$ and $\mathsf{h}(x \equiv s) \leq \mathsf{h}(y \equiv t)$." Since $(y \equiv t) \prec^+ (x \equiv s)$ implies $\mathsf{h}(x \equiv s) > \mathsf{h}(y \equiv t)$ (by Lemma 41(1)), the condition in **NIdent** implies the one in **Ident**, although the converse need not be true. However, the strengthening of the precondition will not have any impact on the completeness of the new combination procedure. In fact, we still have the property that for any two equations $x \equiv s, y \equiv t$ in an abstraction system, either $\mathsf{h}(x \equiv s) \leq \mathsf{h}(y \equiv t)$ or $\mathsf{h}(x \equiv s) \geq \mathsf{h}(y \equiv t)$. Thus, if the abstraction system contains two distinct equations $x \equiv s, y \equiv t$ satisfying the condition that s, t are i-terms and $s =_{E_i} t$ for $i = 1$ or $i = 2$, then **NIdent** is applicable. As a consequence, all the previous proofs in which we assumed that **Ident** had been applied carry over unchanged to this section where we assume that **NIdent** has been applied.

The main idea of the rules **Shar1** and **Shar2** is to push shared function 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

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

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

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

if (a) t is an E_i-reducible i-term for $i = 1$ or $i = 2$,
(b) $NF^{E_i}_{\Sigma}(t) = s(\bar{y}, \bar{r}) \notin V$,
(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 $(y \equiv r) \in T$.

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

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

Figure 4: The New Derivation Rules.

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 8 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 expressions containing tuples are to be interpreted accordingly; e.g., $[\bar{z}/\bar{r}]$ replaces the variable z_j by r_j for each

$j \in \{1, \dots, n\}$.

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 equivalent 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 8(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 $x \equiv t$.[30] **Shar2** restores this property by merging into $x \equiv s$ all of its immediate successors—which are collected, if any, in the set $\bar{y} \equiv \bar{r}$. Condition (d) in **Shar2** makes sure that the tuple $\bar{y} \equiv \bar{r}$ 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 $\bar{r}$ (resp. $\bar{r}_1$) be in normal form is necessary to ensure termination. We will see that it can be imposed without loss of generality.

We prove below that the new combination procedure decides the word problem for $E = E_1 \cup E_2$ again by showing that the procedure is sound, terminates on all inputs, and, whenever ${E_1}^{\Sigma} = {E_2}^{\Sigma}$, is also complete.

6.3 The Correctness Proof

In the following, we assume that S_i and $\prec_i$ are defined as in Subsection 4.3. Again, we will first show that all sets S_j obtained in correspondence of one run of the combination procedure are in fact abstraction systems.

Lemma 43 *Given an execution of the combination procedure, S_j is an abstraction system for all $j \geq 0$.*

Proof. We prove the claim by induction on j. The induction base ($j = 0$) is again immediate by construction of S_0 and Proposition 9.[31] The induction step is also proved as in Lemma 11 for the cases in which S_j is derived from S_{j-1} by an application of **Coll1**, **Coll2**, or **NIdent**.[32] We show below that S_j is an abstraction system even when it is derived by **Shar1** or by **Shar2**.

[30]Recall that we may 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 26). Thus, the equations in $\bar{y} \equiv \bar{r}$ are in fact successors of $x \equiv t$.

[31]Note that this proposition also holds if the signatures Σ_1 and Σ_2 are not disjoint.

[32]To reuse the proof in Lemma 11 for the **NIdent** case, we appeal to the fact that $x \neq y$ and $\mathsf{h}(x \equiv s) \leq \mathsf{h}(y \equiv t)$ imply $(y \equiv t) \not\prec^* (x \equiv s)$.

Shar1. We know that S_{j-1} and S_j have the following form:

$$\begin{array}{lllllllll} S_{j-1} & = & T & & & \cup & \{u \not\equiv v\} & \cup & \{x \equiv t\} & \cup & \{\bar{y}_1 \equiv \bar{r}_1\} \\ S_j & = & T[x/s(\bar{y},\bar{z})[\bar{y}_1/\bar{r}_1]] & \cup & \bar{z} \equiv \bar{r} & \cup & \{u \not\equiv v\} & \cup & \{x \equiv s(\bar{y},\bar{r})\} & \cup & \{\bar{y}_1 \equiv \bar{r}_1\} \end{array}$$

To see that S_j satisfies Condition (1) of Definition 8, first notice that $s(\bar{y},\bar{r})$ is not a variable by precondition (b) of the rule, and that the terms in $\bar{r}$ are also non-variable terms. Because S_{j-1} is assumed to be an abstraction system, it satisfies the alternating signature assumption, and thus the terms in $\bar{r}_1$ are Σ_ι-terms for $\iota \in \{1,2\} \setminus \{i\}$. Since $s(\bar{y},\bar{z})$ is a Σ-term, we know that $s(\bar{y},\bar{z})[\bar{y}_1/\bar{r}_1]$ is also a Σ_ι-term. The alternating signature assumption for S_{j-1} also implies that any term in T containing x is a Σ_ι-term, and so the replacement of x by $s(\bar{y},\bar{z})[\bar{y}_1/\bar{r}_1]$ does not generate mixed terms.

Condition (3a) is satisfied because $\bar{z}$ consists of fresh variables with no repetitions. Condition (3b) is satisfied because

- the replacement of x by $s(\bar{y},\bar{z})[\bar{y}_1/\bar{r}_1]$ in T does not change the signature of any equations there, nor does it change the top symbol of any term;
- the elements of $\bar{r}$ are members of $G_{E_i}(\Sigma, V) \setminus V$, and thus do no start with a Σ-symbol;
- $\bar{r}$ has the same signature as t, and every immediate $\prec$-predecessor of an equation in $\bar{z} \equiv \bar{r}$ has the signature of the immediate predecessors of $x \equiv t$ in S_{j-1};
- $x \equiv s(\bar{y},\bar{r})$, which possibly starts with a shared symbol, has no $\prec$-predecessors in $S_j \setminus \{u \not\equiv v\}$ since x has been replaced;
- all the immediate successors of $x \equiv s(\bar{y},\bar{r})$ are inherited from $x \equiv t$ because, by our assumptions on the variables of normal forms, the variables in $\mathcal{V}ar(s(\bar{y},\bar{r})) \setminus \mathcal{V}ar(t)$ do not occur in S_{j-1};
- $s(\bar{y},\bar{r})$ is not a shared term because the tuple $\bar{r}$ is assumed to be non-empty;
- if an equation $x' \equiv t'[x]$ in T is replaced by $x' \equiv t'[s(\bar{y},\bar{z})[\bar{y}_1/\bar{r}_1]]$, then any new successor of such an equation is an equation in $\bar{z} \equiv \bar{r}$ or a successor of an equation in $\bar{y}_1 \equiv \bar{r}_1$.

To show that Condition (2) is satisfied, we first prove that $T_j := S_j \setminus \{\bar{z} \equiv \bar{r}\}$ gives rise to an acyclic graph. This graph has essentially the same nodes (i.e., equations) as S_{j-1}, although the right-hand sides of the equations may have changed. Even if there are possibly new edges, it is easy to see that there are no new connections between nodes, since any connection achieved by such a new edge in T_j can be achieved by a path in S_{j-1}. Since S_{j-1} gives rise to an acyclic graph by assumption, this implies that the graph corresponding to T_j is acyclic as well. The additional nodes in S_j (i.e., the equations in $\bar{z} \equiv \bar{r}$) cannot cause a cycle either since any path through such a node comes from a predecessor of $x \equiv t[\bar{y}]$ in S_{j-1} and goes to a successor of $x \equiv t[\bar{y}]$ in S_{j-1}. Thus, the cycle would have already been present in S_{j-1}.

Shar2. We know that S_{j-1} and S_j have the following form:

$$\begin{array}{lclcccccc} S_{j-1} & = & T & \cup & \{u \not\equiv v\} & \cup & \{x \equiv t[\bar{y}]\} & \cup & \bar{y} \equiv \bar{r} \\ S_j & = & T[x/s[\bar{y}/\bar{r}]] & \cup & \{u \not\equiv v\} & \cup & \{x \equiv s[\bar{y}/\bar{r}]\} & \cup & \bar{y} \equiv \bar{r} \end{array}$$

We can show that S_j satisfies Conditions (1), (2), (3a), and (3b) of Definition 8 essentially in the same way as in the **Shar1** case. For Condition (3b), additionally observe that we cannot use $x \equiv s$ in S_j since s is a shared term. In fact, $x \equiv s$ together with an equation in $\bar{y} \equiv \bar{r}$ would violate the alternating signature assumption. By using $x \equiv s[\bar{y}/\bar{r}]$ instead, we make sure that any successors of this equation is a successor of an equation in $\bar{y} \equiv \bar{r}$. Since every equation in $\bar{y} \equiv \bar{r}$ is a successor of $x \equiv t$ in S_{j-1},[33] and S_{j-1} satisfies Condition (3b) by induction, all the equations in $\bar{y} \equiv \bar{r}$ have the same signature, which is also the signature of $x \equiv s[\bar{y}/\bar{r}]$. Thus, Condition (3b) for $x \equiv s[\bar{y}/\bar{r}]$ and its successors in S_j is satisfied since it is satisfied for the equations in $\bar{y} \equiv \bar{r}$ and their successors in S_{j-1}. □

In the lemma below we show that the combination procedure halts on all inputs. For that we will make use of a well-founded ordering[34] on abstraction systems, defined in the following.

Let $>_l$ denote the lexicographic ordering over the set $P := \mathbb{N} \times \{0, 1\}$ obtained from the standard strict ordering over $\mathbb{N}$ and its restriction to $\{0, 1\}$. Where $\mathcal{M}(P)$ denotes the set of all finite multisets of elements of P, we will denote by $\sqsupset$ the *multiset ordering induced by* $>_l$, that is, the relation on $\mathcal{M}(P)$ defined as follows—where $\in, \subseteq, =, \setminus, \cup$ are to be interpreted as multiset operators (see [DM79] for more details).

Definition 44 ($\sqsupset$) *For all $M, N \in \mathcal{M}(P)$, $M \sqsupset N$ iff there exist $X, Y \in \mathcal{M}(S)$ such that*

- *$\emptyset \neq X \subseteq M$,*
- *$N = (M \setminus X) \cup Y$, and*
- *for all $y \in Y$ there is an $x \in X$ such that $x >_l y$.*

[33] Recall again that the variables in $\mathcal{V}ar(s) \setminus \mathcal{V}ar(t)$ do not occur in S_{j-1}.

[34] A strict ordering $>$ is well-founded if there are no infinitely decreasing chains $a_1 > a_2 > a_3 > \cdots$.

It is possible to show that $\sqsupset$ is a well-founded total ordering on $\mathcal{M}(P)$ [DM79]. Intuitively, this ordering says that a multiset M is reduced by removing one or more elements from M, and replacing them by a finite number of $>_l$-smaller elements. As customary, we will denote by $\sqsupseteq$ the reflexive closure of $\sqsupset$.

Now, given a run of the combination procedure, let h_j and r_j be the height and reducibility functions on the nodes of the dag induced by S_j, for $j \geq 0$. These functions can be used to associate a finite multiset to the abstraction system S_j: the multiset M_j consisting of the pairs $(\mathsf{h}_j(a), \mathsf{r}_j(a))$ for every (dis)equation a in S_j. Notice that M_j is indeed a multiset: if S_j contains m irreducible nodes with height n, M_j contains m occurrences of the pair $(n, 0)$. Similarly, if S_j contains m reducible nodes with height n, M_j contains m occurrences of the pair $(n, 0)$.

The next lemma shows that each application of a derivation rule decreases the multiset associated to the current abstraction system with respect to the ordering $\sqsupset$.

Lemma 45 *For all $j \geq 0$, $M_j \sqsupset M_{j+1}$ whenever S_{j+1} is generated from S_j by an application of* **Coll1** *or* **Coll2** *or* **Simpl** *or* **NIdent** *or* **Shar**.

Proof. We consider only the application of **Coll1**, **NIdent**, **Shar1**, and **Shar2**. The proof for **Coll2** is very similar to that for **Coll1**, and the proof for **Simpl** is trivial.

Coll1. We can think of S_{j+1} as being derived from S_j by applying the intermediate steps below.

$$\begin{array}{lllllllll}
S_j & = & T & \cup & \{u \not\equiv v\} & \cup & \{v_1 \equiv s_1[v_2]\} & \cup & \{v_2 \equiv s_2\} \\
S & = & T & \cup & \{u \not\equiv v\}[v_1/v_2] & \cup & \{v_1 \equiv s_1[v_2]\} & \cup & \{v_2 \equiv s_2\} \\
S' & = & T[v_1/s_2] & \cup & \{u \not\equiv v\}[v_1/v_2] & \cup & \{v_1 \equiv s_1[v_2]\} & \cup & \{v_2 \equiv s_2\} \\
S_{j+1} & = & T[v_1/s_2] & \cup & \{u \not\equiv v\}[v_1/v_2] & \cup & & & \{v_2 \equiv s_2\}
\end{array}$$

As in the proof of Lemma 11 we can easily show that S and S' are abstraction systems as well. Then, where M and M' are the multisets associated to S and S', respectively, we show that $M_j \sqsupseteq M \sqsupseteq M' \sqsupset M_{j+1}$.

($M_j \sqsupseteq M$) If v_1 does not occur in $u \not\equiv v$ then $M_j = M$, as S_j and S coincide. If v_1 occurs in $u \not\equiv v$, then we know that, since the height of $u \not\equiv v$ in S_j is 0, the height of $v_2 \equiv s_2$ is at least 2. Now, the replacement of v_1 by v_2 turns the dag induced by S_j into the dag induced by S essentially by adding an edge from $u \not\equiv v$ to $v_2 \equiv s_2$ and removing the edge from $u \not\equiv v$ to the equation $v_1 \equiv s_1$. By points 2 and 3 of Lemma 41, some nodes in S may have a smaller height than they had in S_j, but no node in S has a greater height. It is obvious that all the nodes have in S the same reducibility they had in S_j. Thus, when going from M_j to M, the first component of some pairs may decrease, but no pair increases. By definition of $\sqsupset$ (Definition 44), we can then conclude that $M_j \sqsupseteq M$.

$(M \sqsupseteq M')$ If v_1 does not occur in T then $M = M'$, as S and S' coincide. If v_1 occurs in T, since S is an abstraction system, it will necessarily occur in the right-hand side of some equations of T. Let $v_0 \equiv s_0$ be any such equation. Since

$$(v_0 \equiv s_0[v_1]) \prec (v_1 \equiv s_1[v_2]) \prec (v_2 \equiv s_2) \tag{6}$$

we know from Lemma 41(1) that every $v \equiv t$ in S such that $(v_2 \equiv s_2) \prec (v \equiv t)$ has a higher height in S than $v_0 \equiv s_0$. The replacement of v_1 by s_2 adds an edge from $v_0 \equiv s_0$ only to nodes $v \equiv t$ like the one above. This means that, going from S to S', the only new edges are from a node of S to one that is already higher. By Lemma 41(2) then no node in S moves to a greater height in S' because of such edge additions. Now, $v_0 \equiv s_0[v_1]$ above becomes $v_0 \equiv s_0[v_1/s_2]$ in S', hence it may become reducible even if it was irreducible before. If n is the height of $v_0 \equiv s_0$ in S, then a pair of the form $(n, 0)$ may be replaced by the larger pair $(n, 1)$ when going from M to M'. This, however, is not a problem because at least one greater pair, $(n+1, \mathsf{r}(v_1 \equiv s_1))$, is replaced by a smaller one as well. To see this observe that, since v_1 does not occur in $S' \setminus \{v_1 \equiv s_1\}$, the height of $v_1 \equiv s_1$ in S' is 0. However, because of $(v_0 \equiv s_0) \prec (v_1 \equiv s_1)$ it was greater than 0 in S. By definition of $\sqsupset$, we can conclude that $M \sqsupset M'$.

$(M' \sqsupset M_{j+1})$ As S_{j+1} is obtained from S' by removing the node $v_1 \equiv s_1$, we can use Lemma 41(4) to show that the pairs corresponding to the remaining nodes do not increase. Since one pair (the one corresponding to $v_1 \equiv s_1$) is removed, this implies $M' \sqsupset M_{j+1}$.

NIdent. We have $S_j = T \cup \{x \equiv s, y \equiv t\}$ and $S_{j+1} = T[x/y] \cup \{y \equiv t\}$, where $\mathsf{h}(x \equiv s) \leq \mathsf{h}(y \equiv t)$ in S_j.

The graph induced by S_{j+1} can be obtained from the one induced by S_j as follows. First, add edges from the immediate predecessors in S_j of $x \equiv s$ to $y \equiv t$. Since the height of $y \equiv t$ is at least the height of $x \equiv s$, and thus larger than the height of these predecessors, Lemma 41(2) shows that this does not change the height of any node. Second, remove the edges that go from the immediate predecessors in S_j of $x \equiv s$ to $x \equiv s$. By Lemma 41(3), this does not increase the height of any node. Third, remove the node $x \equiv s$. By Lemma 41(4), this does not increase the height of any of the remaining nodes.

By applying the substitution $[x/y]$ to the equations in T, the reducibility of a node containing x may change from 0 to 1. However, these nodes have a height that is smaller than the height of $x \equiv s$. Thus, an increase in the pair associated to such a node in the multiset is compensated by the fact that the pair associated to $x \equiv s$ is removed. This shows that $M_j \sqsupset M_{j+1}$.

Shar1. We know that S_j and S_{j+1} have the following form:

$$\begin{array}{lllllllll} S_j & = & T & & & \cup & \{u \not\equiv v\} & \cup & \{x \equiv t\} & \cup & \{\bar{y}_1 \equiv \bar{r}_1\} \\ S_{j+1} & = & T[x/s(\bar{y}, \bar{z})[\bar{y}_1/\bar{r}_1]] & \cup & \bar{z} \equiv \bar{r} & \cup & \{u \not\equiv v\} & \cup & \{x \equiv s(\bar{y}, \bar{r})\} & \cup & \{\bar{y}_1 \equiv \bar{r}_1\} \end{array}$$

Observe that there may be more nodes in S_{j+1} than S_j: those corresponding to equations in $\bar{z} \equiv \bar{r}$. Let n be the height of $x \equiv t$ in S_j. We start by showing that the height of the new nodes in S_{j+1} cannot be greater than n.

Going from S_j to S_{j+1}, the new equations $\bar{z} \equiv \bar{r}$ are introduced while each occurrence of x in the right-hand side of an equation is replaced by $s(\bar{y}, \bar{z})[\bar{y}_1/\bar{r}_1]$. Consider any equation $z \equiv r$ in $\bar{z} \equiv \bar{r}$. Observing that z occurs in the tuple $\bar{z}$ we then obtain

$$\varphi[x/s(\bar{y}, \bar{z})[\bar{y}_1/\bar{r}_1]] \prec_{j+1} (z \equiv r)$$

for all equations φ (and only those) such that

$$\varphi \prec_j (x \equiv t).$$

Using the fact that $\prec_j$ is acyclic, it is easy to see that no such equation φ changes its height when going from S_j to S_{j+1}. As a consequence, $z \equiv r$ has in S_{j+1} the height that $x \equiv t$ had in S_j, namely, n.

The new node $z \equiv r$ may also have outgoing edges. Since the variables in $\mathcal{V}ar(s(\bar{y}, \bar{r})) \setminus \mathcal{V}ar(t)$ do not occur in S_j, however, these edges will go only into old nodes ψ such that $x \equiv t \prec_j \psi$. In other words, all the edges out of $z \equiv r$ will end in nodes whose height was already $> n$ in S_j.

Similarly, the replacement of x by $s(\bar{y}, \bar{r})[\bar{y}_1/\bar{r}_1]$ in T may introduce new edges in S_{j+1} between old nodes,[35] but it is again easy to see that each of these edges will go from a node to one with already greater height. Finally, and again because the variables in $\mathcal{V}ar(s(\bar{y}, \bar{r})) \setminus \mathcal{V}ar(t)$ do not occur in S_j, the replacement of t by $s(\bar{y}, \bar{r})$ in the node $x \equiv t$ will possibly remove some edges from S_{j+1}, but will not introduce new ones.

By Points 1 and 3 of Lemma 41 then some old nodes may move to a lower height in S_{j+1} but none will move to a higher height because of the mentioned replacements. In conclusion, we can say that the number of nodes at heights $> n$ will not increase from S_j to S_{j+1}. In addition, the reducibility value of these nodes will not change (since their right-hand sides are not modified).

Now, if some node with height $> n$ in S_j moves to a smaller height in S_{j+1}, we can already conclude that $M_j \sqsupset M_{j+1}$. If, on the contrary, all the nodes at height $> n$ keep the same height, to prove that $M_j \sqsupset M_{j+1}$ we argue that some of the nodes at height n change their reducibility from 1 to 0. To see that, it is enough to make the following three observations. First, it is possible that the replacement of x by $s(\bar{y}, \bar{z})$ alters the reducibility of some nodes to 1, but as shown above this will happen only at heights $< n$. Second, when no old node at height $> n$ moves

[35]Specifically, between a node of the form $x_0 \equiv t_0[x]$ and a successor node of one of the equations in $\bar{y}_1 \equiv \bar{r}_1$.

to a smaller height, the number of nodes at height n increases only because of the presence of the new nodes in $\bar{z} \equiv \bar{r}$, whose reducibility is 0, as each $r \in \bar{r}$ is already in normal form. Third, the node $x \equiv t$ in S_j, which by assumption had height n and was reducible, may or may not move to a lower height in S_{j+1}, but it certainly becomes irreducible for being changed to $x \equiv s(\bar{y}, \bar{r})$ where $s(\bar{y}, \bar{r})$ is in normal form.

Shar2. We know that S_j and S_{j+1} have the following form:

$$\begin{array}{lclcccccc} S_j & = & T & \cup & \{u \not\equiv v\} & \cup & \{x \equiv t[\bar{y}]\} & \cup & \bar{y} \equiv \bar{r} \\ S_{j+1} & = & T[x/s[\bar{y}/\bar{r}]] & \cup & \{u \not\equiv v\} & \cup & \{x \equiv s[\bar{y}/\bar{r}]\} & \cup & \bar{y} \equiv \bar{r} \end{array}$$

Let n be the height of $x \equiv t$ in S_j. As in the **Shar1** case we can show that the number of nodes at height $> n$ does not increase going from S_j to S_{j+1}, and that the reducibility value of these nodes does not change. It is enough to show then that the number of reducible nodes at height n decreases by one. Now, the node $x \equiv t$ in S_j changes to $x \equiv s[\bar{y}/\bar{r}]$ in S_{j+1}. Because S_j is an abstraction system, we know that the elements of $\bar{r}$ are all Σ_i-terms for $i = 1$ or $i = 2$. Moreover, each of them is irreducible by assumption and so has the form $s'(\bar{r}')$ where s' is a Σ-term and all the terms in $\bar{r}'$ are in $G_{E_i}(\Sigma, V)$. It is easy to see that $s[\bar{y}/\bar{r}]$ too is a Σ_i-term in normal form, which means that $x \equiv s[\bar{y}/\bar{r}]$ is irreducible. □

Proposition 46 (Termination) *The combination procedure halts on all inputs.*

Proof. Consider any run of the combination procedure. Since, for $i = 1, 2$, $NF_{\Sigma}^{E_i}$ is computable by assumption and the E_i-irreducibility of Σ_i-terms is decidable by Proposition 27, it is immediate that **Shar1**, **Shar2** are applicable in finite time. We already know that the other derivation rules are applicable in finite time as well. As in the proof of Proposition 14 then all we need to show is that the procedure applies the various rules only finitely many times. But this is immediate by Lemma 45 and the well-foundedness of $\sqsupset$. □

The next lemma shows that the derivation rules preserve satisfiability.

Lemma 47 *For all $j > 0$ and all models $\mathcal{A}$ of $E = E_1 \cup E_2$, the abstraction system S_j is satisfiable in $\mathcal{A}$ iff S_{j-1} is satisfiable in $\mathcal{A}$.*

Proof. As before, we can index all the possible cases by the derivation rule applied to S_{j-1} to obtain S_j. The cases **Coll1**, **Coll2**, **NIdent**, **Simpl** are proved exactly as in Lemma 12. Below we give a proof only of the **Shar1** case, as the proof for **Shar2** is almost identical.

When S_j is generated by an application of **Shar1**, S_{j-1} and S_j have the form

$$\begin{array}{lclcccccccc} S_{j-1} & = & T & & & \cup & \{u \not\equiv v\} & \cup & \{x \equiv t\} & \cup & \{\bar{y}_1 \equiv \bar{r}_1\} \\ S_j & = & T[x/s(\bar{y}, \bar{z})[\bar{y}_1/\bar{r}_1]] & \cup & \bar{z} \equiv \bar{r} & \cup & \{u \not\equiv v\} & \cup & \{x \equiv s(\bar{y}, \bar{r})\} & \cup & \{\bar{y}_1 \equiv \bar{r}_1\} \end{array}$$

Since no variable in $\bar{z}$ also occurs in $T \cup \{u \not\equiv v\}$, it is easy to see that S_j is equisatisfiable with the set

$$T[x/s(\bar{y},\bar{r})] \quad \cup \quad \{u \not\equiv v\} \quad \cup \quad \{x \equiv s(\bar{y},\bar{r})\} \quad \cup \quad \{\bar{y}_1 \equiv \bar{r}_1\}$$

The claim then follows from the fact that $\mathcal{A}$ is a model of E and $t =_E s(\bar{y},\bar{r})$ (because $t =_{E_i} s(\bar{y},\bar{r})$ for $i = 1$ or $i = 2$ and $E_i \subseteq E$). □

Exactly as we did in Section 4.3 we can now prove that the extended combination procedure is sound.

Proposition 48 (Soundness) *If the combination procedure succeeds on an input (s_0, t_0), then $s_0 =_E t_0$.*

The completeness proof will be simplified by appealing to the following lemma.

Lemma 49 *The final abstraction system S_n generated by a failed execution of the combination procedure can be partitioned into the sets*

$$D := \{x \not\equiv y\} \qquad T_1 := \{u_j \equiv r_j\}_{j \in J} \qquad T_2 := \{v_k \equiv t_k\}_{k \in K}$$

where

1. *x an y are distinct and J and K are finite;*
2. *each $r_j \in T(\Sigma_1, V) \setminus V$ and each $t_k \in T(\Sigma_2, V) \setminus V$;*
3. *each u_j occurs only once in T_1 and each v_k occurs only once in T_2;*
4. *for all $v \in \mathcal{V}ar(T_1) \cap \mathcal{V}ar(T_2)$,*
   (a) *if $v = u_j$ for some $j \in J$ then $v \in \mathcal{V}ar(t_k)$ for some $k \in K$,*
   if $v = v_k$ for some $k \in K$ then $v \in \mathcal{V}ar(r_j)$ for some $j \in J$,
   (b) *if $v = u_j$ for some $j \in J$ then $r_j \in G_{E_1}(\Sigma, V)$,*
   if $v = v_k$ for some $k \in K$ then $t_k \in G_{E_2}(\Sigma, V)$.

Proof. Since the procedure has failed, we know that $x \neq y$, and thus point 1 is trivial. Points 2, 3, 4a are an immediate consequence of the fact that S_n is an abstraction system.

To prove (4b), let $v = u_j$ for some $j \in J$ (if $v = v_k$ the argument is analogous). We claim that r_j is in normal form (i.e., irreducible). In fact, if we assume otherwise we can also assume with no loss of generality, since $\prec$ is acyclic and S_n is finite, that there are no equations $v_k \equiv t_k$ in S_n such that t_k is reducible and $(u_j \equiv r_j) \prec (v_k \equiv t_k)$.[36]

[36]Otherwise we can consider the case in which $v = v_k$ since v_k is also a shared variable.

But then, one of **Coll1**, **Coll2**, **Shar1**, **Shar2** applies to $u_j \equiv r_j$, against the assumption that S_n is the final abstraction system.

Now, from (4a) above we know that there is an equation $v_k \equiv t_k$ in T_2 such that $(v_k \equiv t_k) \prec (u_j \equiv r_j)$. By Definition 8(3b), the top symbol of r_j cannot be a Σ_2-symbol and so, in particular, cannot be a Σ-symbol. But the only Σ_1-terms in normal form that do not start with a Σ-symbol are the terms of $G_{E_1}(\Sigma, V)$. □

To prove that the procedure is complete for the word problem in $E := E_1 \cup E_2$ we make the additional assumption that

$$E_1{}^{\Sigma} = E_2{}^{\Sigma}.$$

In this case, we have the following.

Proposition 50 (Completeness) *The combination procedure succeeds on input* (t_1, t_2) *if* $t_1 =_E t_2$.

Proof. As before, we can prove the claim by proving that, if the procedure fails on input (t_1, t_2), then $t_1 \neq_E t_2$. Suppose then, that the procedure fails and S_n is the final abstraction system. Given Lemma 47 and the construction of S_0, it is enough to show that S_n is satisfiable in E.

From Lemma 49 we know that S_n is an abstraction system with an initial formula of the form $x \not\equiv y$, where x and y are distinct. Furthermore, $S_n \setminus \{x \not\equiv y\}$ can be partitioned into the sets

$$T_1 \;:=\; \{u_j \equiv r_j\}_{j \in J} \quad \text{and} \quad T_2 \;:=\; \{v_k \equiv t_k\}_{k \in K},$$

where T_1 and T_2 satisfy Lemma 49(1–4b). 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

$$Y_i \;:=\; \{[\![r]\!]^{\mathcal{A}_i}_{\alpha_i} \mid r \in G_{E_i}(\Sigma, V)\}$$

where α_i is any bijective valuation of V onto X_i as in the proof of Theorem 24. By Corollary 25, $\mathcal{A}_i{}^{\Sigma}$ is $E_i{}^{\Sigma}$-free with generators Y_i and $X_i \subseteq Y_i$.

Now, for $i = 1, 2$, we will construct a valuation β_i of $\mathcal{V}ar(T_i)$ into $\mathcal{A}_i$ that assigns with a distinct element of Y_i each variable shared by $\{x \not\equiv y\} \cup T_1$ and T_2. Furthermore, β_1 and β_2 will be such that

$$\mathcal{A}_1, \beta_1 \models \{x \not\equiv y\} \cup T_1 \quad \text{and} \quad \mathcal{A}_2, \beta_2 \models T_2.$$

By Proposition 31 then, this will entail that $\{x \not\equiv y\} \cup T_1 \cup T_2$ (that is, S_n) is satisfiable in E. Again, we can restrict our attention to the case in which $i = 1$, as the other case (which is even simpler) can be treated analogously.

Let β_1 be the valuation of $\mathcal{V}ar(T_1)$ defined as follows:

$$\beta_1(v) := \begin{cases} \alpha_1(v) & \text{for all } v \in \bigcup_j \mathcal{V}ar(r_j) \\ [\![r_j]\!]_{\alpha_1}^{\mathcal{A}_1} & \text{for all } v \in \bigcup_j \{u_j\} \end{cases}$$

Such a valuation is well-defined because all the variables u_j are distinct and none of them belongs to $V_1 := \bigcup_j \mathcal{V}ar(r_j)$, as shown in Lemma 49. By construction, β_1 satisfies T_1 in $\mathcal{A}_1$. We prove below that β_1 is injective.

Let $u, v \in \mathcal{V}ar(T_1)$, $u \neq v$. If both u and v are in V_1, then $\beta_1(u) \neq \beta_1(v)$ by construction of α_1. Hence, let $u = u_j$ for some $j \in J$ and assume by contradiction that $\beta_1(u_j) = \beta_1(v)$.

If $v = u_\ell$ for some $\ell \in J$, then $\mathcal{A}_1, \beta_1 \models r_j \equiv r_\ell$ by construction of β_1. As β_1 evaluates the variables in the equation $r_i \equiv r_j$ by distinct generators of $\mathcal{A}_1$, and $\mathcal{A}_1$ is E_1-free, we obtain that $r_j =_{E_1} r_\ell$ by Proposition 1; but then, since either $\mathsf{h}(u_\ell \equiv r_\ell) \leq \mathsf{h}(u_j \equiv r_j)$ or $\mathsf{h}(u_j \equiv r_j) \leq \mathsf{h}(u_\ell \equiv r_\ell)$, **NIdent** applies to S_n against the assumption that S_n is the final abstraction system.

If $v \in V_1$, similarly to the previous case, we can show that $v =_{E_1} r_j$ and (since E_1 is non-trivial) that v occurs in r_j. Therefore, either **Coll1** or **Coll2** applies, again against the assumption that S_n is the final abstraction system. In conclusion, β_1 is injective.

We now show that $\beta_1(v) \in Y_1$ for every variable v that T_1 shares with T_2. Let $v \in \mathcal{V}ar(T_1) \cap \mathcal{V}ar(T_2)$. If $v \in V_1$, then $\beta_1(v) = \alpha_1(v) \in X_1 \subseteq Y_1$ by construction. If $v = u_j$ for some $j \in J$, we know from Lemma 49(4b) that $r_j \in G_{E_1}(\Sigma, V)$. Observing that β_1 assigns the variables of r_j with elements of X_1 and recalling the definition of Y_1, we can conclude that $\beta_1(v)$, which is the same as $[\![r_j]\!]_{\alpha_1}^{\mathcal{A}_1}$, is an element of Y_1.

To complete the proof we finally need to make sure that β_1 is properly defined for x and y as well. If both x and y occur in T_1, we know by the above that β_1 is already defined for them and that $\beta_1(x) \neq \beta_1(y)$, as x and y are distinct. If x occurs in T_2 as well, we also know that $\beta_1(x) \in Y_1$ (similarly for y). If x or y (or both) does not occur in T_1, let $Z := \{x, y\} \setminus \mathcal{V}ar(T_1)$. Since Y_1 is infinite, we can extend β_1 arbitrarily to $\mathcal{V}ar(T_1) \cup Z$ so that, for all $z \in Z$, $\beta_1(z) \in Y_1$ and $\beta_1(z) \neq \beta_1(v)$ for all $v \in \mathcal{V}ar(T_1) \cup Z \setminus \{z\}$.

In conclusion, we have constructed a valuation β_1 of $\mathcal{V}ar(T_1) \cup \{x, y\}$ which satisfies $\{x \not\equiv y\} \cup T_1$ in $\mathcal{A}_1$ and maps the variables shared by $\{x \equiv y\} \cup T_1$ and T_2 to distinct elements of Y_1. □

The results of this section, which show the total correctness of the extended procedure, are indeed a lifting of the correctness results in Section 4.3. In fact, whenever the set Σ of symbols shared by E_1 and E_2 is empty, it is a set of constructors for both E_1 and E_2, provided that each of them is non-trivial. Furthermore, ${E_1}^\Sigma$ and ${E_2}^\Sigma$ are the same because they both coincide with the set $\{v \equiv v \mid v \in V\}$.

Combining the results of this section then we obtain the following modularity result for the decidability of the word problem, which properly extends Theorem 16.

Theorem 51 *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.

In contrast to the termination proof in the disjoint case, the termination argument employed in Lemma 45 does not provide us with an upper-bound on the complexity of the combination procedure. The actual complexity of the procedure will crucially depend on the normal forms computed by the functions $NF_{\Sigma}^{E_i}$.

From Theorem 39 it follows that, given the right conditions, the combination procedure applies immediately by recursion to more than two component theories. For instance, to obtain a decision procedure for the word problem in $E_1 \cup E_2 \cup E_3$ one first applies the combination procedure for E_1 and E_2, and then for $E_1 \cup E_2$ and E_3. The next corollary states what the "right conditions" are.

Corollary 52 *Let Σ be a functional 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 also decidable.

Alternatively, one could prove this corollary by directly extending the combination procedure to handle the union of $n > 2$ theories pairwise sharing the same constructors.

7 Related work

In this section, we investigate the connection between our notion of a constructor and the one introduced in [DKR94]. We will show that their notion is a special case of ours, and that their combination result for the word problem in theories sharing constructors (Theorem 14 in [DKR94]) can be obtained as a corollary of Theorem 51.

Before we can define the notion of constructors according to [DKR94], called DKR-constructors in the following, we need to introduce some notation. 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 $f \in \Omega$. Notice that it is always possible to construct a (total,) well-founded, monotonic ordering on $T(\Omega, V)$ for any functional signature Ω.[37]

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 53 *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 $t = t{\downarrow}_E^>$. The following are some easy consequences of Definition 53.

Lemma 54 *Let Σ be set of* DKR*-constructors for E w.r.t. $>$.*

1. *For all $s, t \in T(\Omega, V)$, $s =_E t$ iff $s{\downarrow}_E^> = t{\downarrow}_E^>$.*
2. *For all $s, t \in T(\Sigma, V)$, $s =_E t$ iff $s = t$,*
   i.e., E^Σ is the theory of syntactic equality on Σ-terms.
3. *If t is in* DKR*-normal form, then all its subterms are also in* DKR*-normal form.*
4. *If $f(s_1, \ldots, s_m) =_E g(t_1, \ldots, t_n)$ for some constructors $f, g \in \Sigma$ and terms $s_1, \ldots, s_m, t_1, \ldots, t_n \in T(\Omega, V)$ then $f = g$ (and thus $n = m$) and $s_i =_E t_i$ for all $i \in \{1, \ldots, m\}$.*

For the theories E_1 and E_2 in Examples 28 and 29, the signature Σ is set of DKR-constructors for E_i $(i = 1, 2)$ w.r.t. an appropriate well-founded and monotonic ordering $>_i$:

[37] For instance, one can take the lexicographic path ordering induced by a total well-founded precedence on $\Omega \cup V$ (see [BN98]), where the variables are treated as constants—which is admissible since the ordering is not required to be closed under substitutions.

- In Example 29 we have seen that orienting the equations of E_2 from left to right yields a canonical term rewriting system R_2 for E_2. Consequently, the transitive closure $\xrightarrow{+}_{R_2}$ of the rewrite relation induced by R_2 is monotonic and well-founded, and every E_2-equivalence class contains a unique R_2-irreducible element. The second point shows that the ordering $>_2 := \xrightarrow{+}_{R_2}$ satisfies Definition 53(1). That (2) of Definition 53 is satisfied is an easy consequence of the fact that no element of Σ occurs on the top of a left-hand side in R_2.

- In Example 28 we cannot simply take the transitive closure of the rewrite relation $\rightarrow_{R,AC}$ as monotonic and well-founded ordering $>_1$. The problem is that normal forms are unique only modulo AC, i.e., an E_1-equivalence class may contain different normal forms, although they can be transformed into each other using equations from AC. We can, however, take an arbitrary total, monotonic, and well-founded ordering $>$ on Σ_1-terms, and define $>_1$ to be the lexicographic product of $\xrightarrow{+}_{R,AC}$ with $>$. The effect of this that the ordering $>$ "picks" a least representative out of the AC-equivalent $\rightarrow_{R,AC}$-normal forms in each E_1-equivalence class. Therefore, Definition 53(1) is satisfied. That Definition 53(2) is also satisfied is again an easy consequence of the fact that no element of Σ occurs on the top of a left-hand side in R, and that the same is true both for left- and right-hand sides of equations in AC.

In contrast, the signature Σ' is *not* a set of DKR-constructors for the theory E_3 of Example 30 since the restriction ${E_3}^{\Sigma'}$ of E_3 to Σ' is not the theory of syntactic equality on Σ'-terms. Hence, a set of constructors in our sense need not be a set of DKR-constructors.

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

Proof. If $r{\downarrow}_E^{>}(\epsilon) \in \Sigma$ then $r \notin G_E(\Sigma, V)$ since $r =_E r{\downarrow}_E^{>}$ by definition of DKR-normal forms. Conversely, assume that $r \notin G_E(\Sigma, V)$, i.e., $r =_E f(\bar{t})$ for some function symbol $f \in \Sigma$ and tuple $\bar{t}$ of Ω-terms. By definition of DKR-constructors, the DKR-normal form $f(\bar{t}){\downarrow}_E^{>}$ of $f(\bar{t})$ has top symbol f, and by Lemma 54(1) it is also the DKR-normal form of r. It follows that $r{\downarrow}_E^{>} \notin \Sigma$. □

Proposition 56 *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 22.*

Proof. We show that the three conditions of Theorem 24 are satisfied.

(1) It is sufficient to show that $v{\downarrow}_E^> = v$ for all variables $v \in V$. Thus, assume that $v{\downarrow}_E^> = t \neq v$. Since E is consistent, the term t must contain v. However, then $v > v{\downarrow}_E^> = t$ contradicts our assumption that $>$ is well-founded and monotonic.

(2) Let t be an arbitrary Ω-term. Then its DKR-normal form $t{\downarrow}_E^>$ can be represented as $s(\bar{r})$, where $s(\bar{v})$ is a Σ-term and all terms r in the tuple $\bar{r}$ have top symbols not in Σ. Since these terms r are subterms of a term in DKR-normal form, they are also in DKR-normal form, and thus belong to $G_E(\Sigma, V)$ by Lemma 55.

(3) Let $s_1(\bar{r}_1), s_2(\bar{r}_2) \in T(\Sigma, G_E(\Sigma, V))$, and assume that $s_1(\bar{v}_1)$, $s_2(\bar{v}_2)$ are obtained from $s_1(\bar{r}_1)$, $s_2(\bar{r}_2)$ by 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. We must show that $s_1(\bar{r}_1) =_E s_2(\bar{r}_2)$ implies $s_1(\bar{v}_1) =_E s_2(\bar{v}_2)$ (since the converse is trivial).

If $s_1(\bar{v}_1)$ is a variable, then $s_1(\bar{r}_1) = r$ for an element r of $G_E(\Sigma, V)$. By definition of $G_E(\Sigma, V)$, this implies that $s_2(\bar{v}_2)$ is also a variable. In addition, since E-equivalent terms are abstracted by the same variable, these two variables coincide, and thus $s_1(\bar{v}_1) = s_2(\bar{v}_2)$. The same argument applies if $s_2(\bar{v}_2)$ is a variable.

Therefore, assume that $s_1(\bar{v}_1)$ and $s_2(\bar{v}_2)$ are both nonvariable Σ-terms. By Lemma 54(4), they have the same top symbol and their respective subterms are E-equivalent. Thus, we can easily show the claim by structural induction. □

Point (2) of the above proof may seem to entail that normal forms for E and Σ are computable in the sense of Definition 26. This is not the case, however, because the argument in (2) relies on DKR-normal forms, whereas the computability of such normal forms is not assured by the sole assumption that Σ is a set of DKR-constructors for E w.r.t $>$. In [DKR94], DKR-normal forms are shown to be computable by also assuming that the so-called symbol matching problem is decidable.

Definition 57 *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.*

For the theories E_1 and E_2 of Examples 28 and 29, it is easy to see that the symbol matching problem on Σ is decidable. In fact, given a Σ_i-term t, one simply computes the normal form $\widehat{t}$ of t w.r.t. the corresponding rewrite relation (i.e., $\rightarrow_{R,AC}$ if $i = 1$, and $\rightarrow_{R_2}$ if $i = 2$). If $\widehat{t}$ starts with a symbol $f \in \Sigma$, then $\widehat{t} = f(\bar{t})$ for some tuple of Ω-terms $\bar{t}$, and thus t matches onto Σ modulo E. Otherwise, it is easy to see that t does not match onto Σ modulo E. This is again a consequence of the fact that no symbol from Σ appears at the top of a left-hand side of a rewrite rule.

As pointed out in [DKR94], 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,[38] 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 58 *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. *$NF^E_\Sigma(t) := f(NF^E_\Sigma(t_1), \ldots, NF^E_\Sigma(t_n))$ 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. *$NF^E_\Sigma(t) := t$, otherwise.*

Lemma 59 *Under the assumptions of Definition 58 the function NF^E_Σ is well-defined and satisfies the requirements of Definition 26.*

Proof. To show that NF^E_Σ is well-defined, it is sufficient to find a well-founded ordering on terms such that, in the first case of the definition, the terms $t_1, \ldots, t_n$ are smaller than t w.r.t. this ordering.

We define this ordering using a mapping α from $T(\Omega, V)$ into the nonnegative integers. For any Ω-term s, its DKR-normal form can be uniquely represented in the form $s{\downarrow}^>_E = s_0(\bar{r})$, where $s_0(\bar{v})$ is a Σ-term and all terms r in the tuple $\bar{r}$ have top symbols that do not belong to Σ. Let $\alpha(s)$ be the size of the term $s_0(\bar{v})$. If we define $s_1 \succ s_2$ iff $\alpha(s_1) > \alpha(s_2)$, then $\succ$ is a well-founded ordering on Ω-terms. It remains to be shown that, if $t =_E f(t_1, \ldots, t_n)$ for some $f \in \Sigma$, then $\alpha(t) > \alpha(t_i)$ for all $i \in \{1, \ldots, n\}$. But this is an easy consequence of the fact that $t{\downarrow}^>_E = f(t_1, \ldots, t_n){\downarrow}^>_E = f(t_1{\downarrow}^>_E, \ldots, t_n{\downarrow}^>_E)$. In conclusion, we have shown that NF^E_Σ is well-defined.

By our assumptions, the case distinction in the definition is effective and a symbol matching algorithm on Σ modulo E exists. Therefore, the function NF^E_Σ is computable as well.

Now we prove by well-founded induction on $\succ$ that $NF^E_\Sigma(t)$ is a normal form of t. When the second case of Definition 58 applies, t belongs to $G_E(\Sigma, V)$ by definition, which entails immediately that $NF^E_\Sigma(t) := t$ is in normal form. When the first case

[38] Recall that our signatures are assumed to be countable, and thus the sets of terms are countable as well.

applies, we know that $NF^E_\Sigma(t) = f(NF^E_\Sigma(t_1), \ldots, NF^E_\Sigma(t_n))$ for some Σ-symbol f and tuple $(t_1, \ldots, t_n)$ such that $t =_E f(t_1, \ldots, t_n)$. As we have seen above, $t \succ t_i$ for all $i \in \{1, \ldots, n\}$, which entails by induction that $NF^E_\Sigma(t_i)$ is a normal form of t_i for each $i \in \{1, \ldots, n\}$. Since $f \in \Sigma$, it is immediate that $f(NF^E_\Sigma(t_1), \ldots, NF^E_\Sigma(t_n))$ is in normal form as well. To see that $NF^E_\Sigma(t)$ is indeed a normal form of t, it is now enough to observe that $t =_E f(t_1, \ldots, t_n) =_E f(NF^E_\Sigma(t_1), \ldots, NF^E_\Sigma(t_n))$, where the last equivalence is a consequence of the induction assumption that $t_i =_E NF^E_\Sigma(t_i)$ for each $i \in \{1, \ldots, n\}$. □

We are now ready to show that Theorem 14 in [DKR94] can be obtained as a corollary of our Theorem 51.

Corollary 60 *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.

Proof. We show that the prerequisites of Theorem 51 are satisfied. By Proposition 56, Σ is a set of constructors according to Definition 22 for both E_1 and E_2. By Lemma 54(2), ${E_1}^\Sigma = {E_2}^\Sigma$ since both coincide with the syntactic equality on Σ-terms. Finally, normal forms are computable for Σ and E_i $(i = 1, 2)$ by Lemma 59. □

We believe that our definition of constructor has several advantages over the one introduced in [DKR94]. First, it is more general since we only require E^Σ to be collapse-free whereas [DKR94] requires E^Σ to be equal to the theory of syntactic equality on Σ-terms. Second, the definition of DKR-constructors is rather technical and depends strongly on the chosen ordering $>$. In contrast, our definition uses only abstract algebraic properties. Finally, the combination algorithm described in [DKR94] is not rule-based, since it is a straightforward extension of the algorithms for the disjoint case described in [SS89, Nip89, KR94], and thus shares the disadvantages of these algorithms, as mentioned in the introduction.

8 Conclusion and Open Questions

In this report, we have introduced a new, rule-based procedure that combines in a modular fashion decision procedures for the word problem. The procedure's main idea, propagation of equality constraints between the component decision procedures, is similar in spirit to the Nelson-Oppen combination method, a general method for combining decision procedures for the validity of quantifier-free formulae in theories over disjoint signatures. Its specifics, however, are essentially different because the word problem is a rather restricted kind of validity problem. As a matter of fact, and contrary to common belief, the Nelson-Oppen method cannot be used for the purpose of combining decision procedures for the world problem, as we have shown in Section 3.

We have first presented (in Section 4) a procedure that can deal with equational theories over disjoint signatures, and then extended this procedure (in Section 6) so that it can also treat theories sharing symbols that we called constructors. Essentially, this extension was achieved by adding two more rules that handle the shared constructors. The reasons for choosing this two-step approach were mainly of a didactic nature. The proof of correctness of the procedure for the disjoint case is simpler than the one for the extended procedure, but has a very similar structure. Thus, it prepares the reader for the more complex proof in the general case.

As mentioned in the introduction, the modularity result for *the disjoint case* has been known for quite some time [Pig74, Tid86, SS89, Nip89, KR94]. Our main goal in Section 4 was to develop a rule-based combination procedure, which is more transparent and more flexible than the known ones, and uses deterministic rules that can be applied in arbitrary order. Another distinguishing feature of our approach is that the proof of completeness of the procedure is based solely on algebraic arguments. This not only provides for a simpler proof, as we think we have demonstrated, but it also leads to a rather general extension of the procedure to the *non-disjoint case*.

The only combination procedure we are aware of for the case of component decision procedures whose theories have symbols in common is described in [DKR94]. We have shown that our approach applies to a more general class of theories than the one considered in [DKR94]. In addition, we believe that our algebraic method yields a less technical, and thus more transparent, definition of this class. It should be noted, however, that [DKR94] also contains combination results for unification and matching, whereas the present report is concerned only with the word problem. Thus, one direction for future research is 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.

A further generalization would be to extend our results to the case of many-sorted equational logic. This should not be very hard, but from a practical point of view it would considerably increase the class of theories to which our approach applies. For instance, many examples from algebraic specification (such as lists of natural numbers, etc.) make sense only in a sorted environment.

References

[BN98] Franz Baader and Tobias Nipkow. *Term Rewriting and All That.* Cambridge University Press, United Kingdom, 1998.

[BT97] Franz Baader and Cesare Tinelli. A new approach for combining decision procedures for the word problem, and its connection to the Nelson-Oppen combination method. In W. McCune, editor, *Proceedings of the 14th International Conference on Automated Deduction (Townsville, Australia)*, volume 1249 of *Lecture Notes in Artificial Intelligence*, pages 19–33. Springer-Verlag, 1997.

[DKR94] 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.

[DM79] Nachum Dershowitz and Zohar Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, August 1979.

[Hod93] Wilfrid Hodges. *Model Theory*, volume 42 of *Enclyclopedia of mathematics and its applications.* Cambridge University Press, 1993.

[KB70] 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.

[KR94] Hélène Kirchner and Christophe Ringeissen. Combining symbolic constraint solvers on algebraic domains. *Journal of Symbolic Computation*, 18(2):113–155, 1994.

[Mat67] J.V. Matijasevic. Simple examples of undecidable associative calculi. *Soviet Mathematics (Doklady)*, 8(2):555–557, 1967.

[MM82] Alberto Martelli and Ugo Montanari. An efficient unification algorithm. *ACM Transactions on Programming Languages and Systems*, 4(2):258–282, 1982.

[Nip89] 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.

[NO79] Greg Nelson and Derek C. Oppen. Simplification by cooperating decision procedures. *ACM Trans. on Programming Languages and Systems*, 1(2):245–257, October 1979.

[NO80] Greg Nelson and Derek C. Oppen. Fast decision procedures based on congruence closure. *J. ACM*, 27(2):356–364, 1980.

[Opp80] Derek C. Oppen. Complexity, convexity and combinations of theories. *Theoretical Computer Science*, 12, 1980.

[Pig74] Don Pigozzi. The join of equational theories. *Colloquium Mathematicum*, 30(1):15–25, 1974.

[Rin96] Christophe Ringeissen. Cooperation of decision procedures for the satisfiability problem. In F. Baader and K.U. Schulz, editors, *Frontiers of Combining Systems: Proceedings of the 1st International Workshop, Munich (Germany)*, Applied Logic, pages 121–140. Kluwer, March 1996.

[Rob65] J.A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12:23–41, 1965.

[SS89] Manfred Schmidt-Schauß. Combination of unification algorithms. *Journal of Symbolic Computation*, 8(1–2):51–100, 1989. Special issue on unification. Part II.

[TH96] Cesare Tinelli and Mehdi T. Harandi. A new correctness proof of the Nelson–Oppen combination procedure. In F. Baader and K.U. Schulz, editors, *Frontiers of Combining Systems: Proceedings of the 1st International Workshop, Munich (Germany)*, Applied Logic, pages 103–120. Kluwer, March 1996.

[Tid86] E. Tidén. *First-Order Unification in Combinations of Equational Theories.* Phd thesis, The Royal Institute of Technology, Stockholm, 1986.

[Toy87a] Yoshihito Toyama. Counterexamples to termination for the direct sum of term rewriting systems. *Information Processing Letters*, 25:141–143, 1987.

[Toy87b] Yoshihito Toyama. On the Church-Rosser property for the direct sum of term rewriting systems. *Journal of the ACM*, 34(1):128–143, 1987.

[TR98] Cesare Tinelli and Christophe 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).