Deciding the Word Problem in the Union of Equational Theories

Franz Baader[1]
Lehr- und Forschungsgebiet Theoretische Informatik
RWTH Aachen
Ahornstraße 55, 52074 Aachen, Germany
E-mail: baader@informatik.rwth-aachen.de

and

Cesare Tinelli[2]
Department of Computer Science
University of Iowa
14 MacLean Hall
Iowa City, IA 52242 – USA
E-mail: tinelli@cs.uiowa.edu

Version: 14 November 2001

The main contribution of this paper is a new method for combining decision procedures for the word problem in equational theories. In contrast to previous methods, this method is based on transformation rules. Furthermore, it is not limited to theories with disjoint signatures but it also applies to theories sharing *constructors.*

Key Words: equational reasoning, word problem, combination of decision procedures.

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 [16]. Nevertheless, there are decidability results for certain classes of equational theories (such as theories defined by a finite

[1]Partially supported by the ESPRIT Working Group CCL II, EP: 22457.
[2]Partially supported by grant no. 9972311 from the National Science Foundation.

set of ground equations [18]), and there are general approaches for tackling the word problem (such as Knuth-Bendix completion [14], which tries to generate a confluent and terminating term rewriting system for the theory).

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

The theories A and G of this example share a function symbol—the binary semigroup operation. What happens if we assume that there are no shared symbols, that is, the theories to be combined are built over disjoint signatures? In this case, decision procedures for the word problem can be combined (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 [21] using results from universal algebra. It was more recently rediscovered in the term rewriting and automated deduction community [24, 23, 19, 13]. Surprisingly, even these more recent presentations do 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 [24], it is not explicitly stated there; in [23, 13] it is stated as a corollary, but not mentioned in the abstract or the introduction; only [19] explicitly refers to the result in the abstract. The combination methods used in all these papers are essentially identical, the main differences lying in their proofs of correctness. They all directly transform the terms for which the word problem is to be decided, by applying collapse equations[3] and abstracting alien subterms. This transformation process must be carried on with a rather strict strategy (in principle, going from the leaves of the terms to their roots) and it is not easy to describe and comprehend.

In this paper, which combines the results first reported in [3], [6] and [7], we present a method for combining decision procedures for the word problem that works on a set of equations rather than terms. It is based on transformation rules, which 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 [15] and Robinson's original one [22]. 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

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

more precise later.

The only previous work that presents a combination method for the word problem in the union of non-disjoint theories is [9], where the problem of combining algorithms for the unification, matching, and word problem is investigated for theories sharing so-called "constructors." The combination method for the word problem described in [9] is not rule-based since it is an extension of the algorithms for the disjoint case, as described in [21, 23, 19, 13]. We will show that the notion of a constructor introduced in [9] is a strict subcase of our notion, and that the combination result for the word problem presented in [9] can also be obtained with our rule-based approach.

A recent work [10], inspired by our results in [6], presents an alternative combination approach for the word problem in the non-disjoint case. The combination method in [10] is based on rewriting techniques and is shown correct by means of category theoretic arguments. As we briefly discuss in Subsection 7.3, although the results in [10] generalize those presented in [6], they are equivalent to our own more general results, first introduced in [7] and now presented here in detail.

It is a common misconception that combining decision procedures for the word problem in the disjoint signature case is a special case of Nelson and Oppen's combination method [17]. 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 is incorrect and for two reasons. First, Nelson and Oppen require the single theories to be stably infinite, and equational theories need not satisfy this property.[4] Second, although we are only interested in the word problem for the combined theory, the Nelson-Oppen method generates validity problems in the single theories that are strictly more general than the word problem. Thus, just knowing that the word problem in each of the single theories is decidable is not sufficient. Nevertheless, our method for combining decision procedures for the word problem follows a similar approach to Nelson and Oppen's. Like them, we use a restricted form of constraint propagation between the decision procedures for the single theories to solve the validity problem in question in the combined theory. More details on the similarity between the two methods can be found in [3].

Outline of the paper. We start in the next section by introducing some necessary notation. In Section 3, we present 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 non-disjoint combination of equational theories, we must establish (in Section 4) some general model-theoretic results for combined equational theories (Subsection 4.1) and introduce our notion of a constructor (Subsection 4.2) together with some properties enjoyed by unions of theories that share constructors (Subsection 4.3). In Section 5, we describe the extended combination procedure for theories sharing constructors, and prove its correctness. In Section 6, we show that our notion of constructors is modular in the sense that the union of two equational theories sharing a certain set Σ of constructors again has Σ as a set of constructors. This property is important since it entails that the application of our combination results can be iterated. We start

[4]It turns out, however, that they satisfy a somewhat weaker property, which in principle suffices to apply their method— see [3] for details.

Section 7 by relating this work to our previous work on the same topic. Next, we briefly compare our modularity results for the decidability of the word problem with some related modularity results from term rewriting. Then we illustrate in detail the connection between our notion of constructors and the one introduced in [9]. Finally, we compare our results with those presented in [10].

2. FORMAL PRELIMINARIES

Throughout the paper, we will consider only functional signatures, that is, signatures containing only function symbols—with constants being functions symbols of zero arity. Thus, the only predicate symbol available is the equality symbol, which we will denote by $\equiv$. All the signatures will be countable and will be usually denoted by the symbols Σ and Ω, possibly with subscripts.

We will denote by V a fixed countably infinite set of variables and by $T(\Sigma, V)$ the set of Σ-terms over V. We will use the symbols q, 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, z 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. Also, if $\bar{r}$ is a tuple of terms with the same length as $\bar{v}$, we will denote by $t(\bar{r})$ the term obtained from $t(\bar{v})$ by replacing each variable of $\bar{v}$ with the corresponding element of $\bar{r}$. When convenient, we will treat a tuple $\bar{r}$ of terms as the set of its elements.

As usual, for all functional signatures Σ, we say that a Σ-formula φ is *valid* in a Σ-theory Γ and write $\Gamma \models \varphi$ iff it holds in all models of Γ, i.e., iff for all Σ-algebras $\mathcal{A}$ that satisfy Γ and all valuations α of the free variables of 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 $f \in \Sigma$ and a Σ-algebra $\mathcal{A}$, 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, we denote by $s^{\mathcal{A}}$ the n-ary term function induced by the term s in $\mathcal{A}$. Given a Σ-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, the interpretation $[\![s]\!]^{\mathcal{A}}_{\alpha}$ may also be written as $s^{\mathcal{A}}(\bar{a})$, where $\bar{a}$ is the tuple of values that α assigns to the variables in s.

An equational theory E over the 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 customary, we write $s =_E t$ to express that the formula $s \equiv t$ is valid in E. We say that a term

t is *collapsing in E* iff $v =_E t$ for some variable v. We say that E is *collapse-free* iff no non-variable term is collapsing in E.

An equational theory E over the signature Σ defines a Σ-*variety*, the class of all models 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.* More precisely, if $\mathcal{A}$ is a free algebra in E's Σ-variety with a set X of free generators we will say that $\mathcal{A}$ is *free in E over X*, or also, that $\mathcal{A}$ is a *free model of E over X*. 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., [11]):

PROPOSITION 2.1. *Let E be an equational theory over Σ and $\mathcal{A}$ a Σ-algebra. Then, $\mathcal{A}$ is free in E over some set 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 paper, we are interested in *combined* equational theories, that is, equational theories E of the form $E := E_1 \cup E_2$, where E_1 and E_2 are equational theories over two (not necessarily disjoint) functional signatures Σ_1 and Σ_2. The elements of $\Sigma := \Sigma_1 \cap \Sigma_2$ are called *shared* symbols.

We call (strict) 1-*symbols* the elements of Σ_1 ($\Sigma_1 \setminus \Sigma$) and (strict) 2-*symbols* the elements of Σ_2 ($\Sigma_2 \setminus \Sigma$). Note that shared symbols are both 1- and 2-symbols, and that they are strict for neither signature.

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 $t = f(t_1, ..., t_n)$ 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. 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.

Most combination procedures produce pure terms and equations by abstracting "alien" subterms (i.e., replacing them by new variables and adding appropriate new equations). Intuitively, an alien subterm of an i-term t is a maximal subterm of t such that its top symbol does not belong to Σ_i. For the case of *disjoint signatures*, this intuition can be straightforwardly transformed into the following formal definition: a subterm s of an i-term t is an *alien subterm* of t iff it is not an i-term and every proper superterm of s in t is an i-term.

If the signatures Σ_1 and Σ_2 are not disjoint, however, this definition is ambiguous since a term t starting with a shared symbol is both a 1- and a 2-term. Then, what counts as an alien subterm of t depends on whether t is considered to be a 1-term or a 2-term. For example, assume that f is a strict 1-symbol, g a strict 2-symbol, and h a shared one. If $t := h(f(x), g(x))$ is considered to be a 1-term, then $g(x)$ is its (only) alien subterm; if t is considered to be a 2-term, then $f(x)$ is its (only) alien subterm. One might think that, to avoid such non-determinism, one could just fix (arbitrarily) that terms starting with a shared symbol are considered to be 1-terms in the definition of alien subterms. However, this would lead to unnecessary abstractions, as exemplified by the term $h(g(x), g(x))$, which would then have the subterms $g(x)$ as alien subterms although it is a pure term. Also, in the (non-pure) term $h(g(f(x)), g(x))$, we would like to have $f(x)$ as alien subterm rather than the two terms $g(f(x))$ and $g(x)$.

The definition of alien subterms given below takes care of all the problems mentioned above.

DEFINITION 2.2 (Alien subterms). Let $t \in T(\Sigma_1 \cup \Sigma_2, V)$. If the top symbol of t is a strict i-symbol, then a subterm s of t is an *alien subterm* of t iff it is not an i-term and it is maximal with this property, i.e., every proper superterm of s in t is an i-term.

If the top symbol of t is a shared symbol, then we consider the set S of all proper maximal subterms of t starting with a non-shared symbol. Let $S = S_1 \cup S_2$ be the partition of S into the terms starting with a strict 1-symbol (S_1) and the terms starting with a strict 2-symbol (S_2).

- If $S_1 \neq \emptyset$, then t is considered to be a 1-term, i.e., a subterm s of t is an *alien subterm* of t iff it is not a 1-term and it is maximal with this property.
- If $S_1 = \emptyset$ and $S_2 \neq \emptyset$, then t is considered to be a 2-term, i.e., a subterm s of t is an *alien subterm* of t iff it is not a 2-term and it is maximal with this property
- If $S_1 \cup S_2 = \emptyset$, then t is pure and so it has no aliens subterms.

3. 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 of signature Σ_i with decidable word problem. To simplify the exposition, we will start by first considering in this section the case in which the signatures of E_1 and E_2 are disjoint. Here our results coincide with the known ones in [21, 24, 23, 19, 13]. What is new is that our combination procedure is based on a number of transformation rules. We will be able to extend the procedure to the non-disjoint signatures case in Section 5 by simply introducing additional rules. As a consequence, almost all the proofs we give in this section will carry over unchanged to Section 5. There, we will only need to take care of changes introduced by the new rules. Of course, to allow non-disjoint signatures will require some additional constraints on the theories to be combined. These constraints will be introduced in Section 4.

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, $\Sigma_1 \cup \Sigma_2$. As in the Nelson-Oppen procedure [17], 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 in more detail the purification process and the result it produces, we need to introduce a little more notation and some new concepts. Since the same notation and concepts will also be employed in the case of non-disjoint signatures, the following subsection does *not* assume Σ_1 and Σ_2 to be disjoint.

3.1. Abstraction Systems

We will often use finite sets of formulae in place of conjunctions of such formulae, that is, we will treat a finite set S of formulae as the formula $\bigwedge_{\varphi \in S} \varphi$. We will then say that S 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 a set $AS(s_0 \not\equiv t_0)$ consisting of pure equations and disequations such that $s_0 \not\equiv t_0$ and $AS(s_0 \not\equiv t_0)$ are "equivalent" in a sense to be made more precise below.

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 occurring in S_i, and s' is obtained from s by replacing t at position p by v. Otherwise, if none of the equations in S_i contains an alien subterm, all terms occurring in S_i are pure, and 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 (see Proposition 3.3 below), whose importance will become clear later on.

Definition 3.1. Let T be a set of equations of the form $v \equiv t$ where $v \in V$ and $t \in T(\Sigma_1 \cup \Sigma_2, V) \setminus V$. The relation $\prec$ on T is defined as follows for all $u \equiv s, v \equiv t \in T$:

$$(u \equiv s) \prec (v \equiv t) \quad \text{iff} \quad v \in \mathcal{V}ar(s).$$

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

Notice that, when $\prec$ is acyclic, $\prec^*$ is a partial order, and $\prec^+$ is the corresponding strict partial order.

Definition 3.2 (Abstraction System). The set $\{x \not\equiv y\} \cup T$ is an *abstraction system with disequation* $x \not\equiv y$ iff $x, y \in V$ and the following holds:

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

Condition 1 above states that T consists of equations between variables and pure non-variable terms; Condition 2 implies that for all $(u \equiv s), (v \equiv t) \in T$, if $(u \equiv s) \prec^* (v \equiv t)$ then $u \notin \mathcal{V}ar(t)$; Condition 3a implies that a variable cannot occur as the left-hand side of more than one equation of T; Condition 3b implies, together with Condition 1, that the elements of every $\prec$-chain of T have *strictly* alternating signatures $(\ldots, \Sigma_1, \Sigma_2, \Sigma_1, \Sigma_2, \ldots)$. In particular, when Σ_1 and Σ_2 have a non-empty intersection Σ, Condition 3b entails that if $(u \equiv s) \prec (v \equiv t)$ neither s nor t can be a Σ-term: one of the two must contain symbols from $\Sigma_1 \setminus \Sigma$ and the other must contain symbols from $\Sigma_2 \setminus \Sigma$.

We will call the variables occurring in an abstraction system S as the left-hand side of an equation the *left-hand side variables* of S. Similarly, we will call the terms occurring in an abstraction system S as the right-hand side of an equation the *right-hand side terms* of S.

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

PROPOSITION 3.3. *The set $S := AS(s_0 \not\equiv t_0)$ obtained by applying the purification procedure to the disequation $s_0 \not\equiv t_0$ is an abstraction system. Furthermore, $\exists \bar{v}.S \leftrightarrow (s_0 \not\equiv t_0)$ is logically valid, where $\bar{v}$ are all the left-hand side variables of S.*

In particular, the second part of the proposition implies that a disequation $s_0 \not\equiv t_0$ is satisfiable in E iff $AS(s_0 \not\equiv t_0)$ is satisfiable in E. However, the statement in the proposition is considerably stronger: if $\mathcal{A}$ is a $(\Sigma_1 \cup \Sigma_2)$-algebra and α a valuation that satisfies $s_0 \not\equiv t_0$ in $\mathcal{A}$, then there exists a valuation α' that coincides with α on $\mathcal{V}ar(s_0 \not\equiv t_0)$ and satisfies $AS(s_0 \not\equiv t_0)$, and vice versa. In fact, the left-hand side variables in $AS(s_0 \not\equiv t_0)$ are fresh variables that do not occur in $s_0 \not\equiv t_0$, and all the newly introduced variables are left-hand side variables. Thus, the variables in $\mathcal{V}ar(s_0 \not\equiv t_0)$ are the free variables of both $s_0 \not\equiv t_0$ and $\exists \bar{v}.S$, which means that they are (implicitly) universally quantified on the outside in the equivalence $\exists \bar{v}.S \leftrightarrow (s_0 \not\equiv t_0)$. We will appeal to this stronger statement in Section 6.

Abstraction Systems as Directed Acyclic Graphs

Every abstraction system $\{x \neq y\} \cup T$ induces a graph $\mathcal{G}$ whose set of *nodes* is T and whose set of *edges* consists of all the pairs $(a_1, a_2) \in T \times T$ such that $a_1 \prec a_2$. According to Definition 3.2, $\mathcal{G}$ is in fact a directed acyclic graph (or *dag*).[5] For notational convenience, we will sometimes identify an abstraction system with the graph induced by it.

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 this section to define the combination procedure, and will also be important in Section 5 to prove the termination of the procedure's extension to the case of equational theories with non-disjoint signatures.

DEFINITION 3.4 (Node Height). Let $\mathcal{G} := (\mathsf{N}, \mathsf{E})$ be a dag with finite sets of nodes and edges. A node $a \in \mathsf{N}$ is a *root* of $\mathcal{G}$ iff there is no $a' \in \mathsf{N}$ such that $(a', a) \in \mathsf{E}$.[6] The function $\mathsf{h}\colon \mathsf{N} \longrightarrow \mathbb{N}$ is defined as follows. For all $a \in \mathsf{N}$,

[5] Observe that $\mathcal{G}$ need not be a tree or even be connected.

[6] Because of the acyclicity condition, any finite dag has at least one root.

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**, **Ident1**, **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.

FIG. 1 The Combination Procedure.

- $\mathsf{h}(a) = 0$, if a is a root of $\mathcal{G}$;
- $\mathsf{h}(a)$ equals the maximum of the lengths of all the paths from the roots of $\mathcal{G}$ to a, otherwise.[7]

3.2. The Combination Procedure

Let now 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 $i = 1, 2$. Figure 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. This procedure repeatedly applies the transformation rules of Figure 2 until no more rules apply.

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 Figure 2. We describe these rules in the style of a sequent calculus. The premise of each rule lists all the formulae in S before the application of the rule, where T stands for all the formulae not explicitly listed. The conclusion of the rule lists all the formulae in S after the application of the rule. It is understood that any two formulae explicitly listed in the premise of a rule are distinct.

In essence, **Coll1** and **Coll2** remove from S collapse equations that are valid in E_1 or E_2, and identify throughout S the variable in their left-hand side with the variable their right-hand side collapses to. **Ident1** identifies any two variables equated to equivalent Σ_i-terms and then discards one of the corresponding equations. The ordering restriction in the precondition of **Ident1** is on the heights that the two equations involved have in the dag induced by S. It is there to prevent the creation of cycles in the relation $\prec$ over S.

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

[7] This maximum exists because $\mathcal{G}$ is finite and acyclic.

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

if $t \in T(\Sigma_i, V)$ and $y =_{E_i} t$ for $i = 1$ or $i = 2$.

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

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

$$\textbf{Ident1}\ \frac{T \qquad x \equiv s \quad y \equiv t}{T[x/y] \qquad\qquad y \equiv t}$$

if $s, t \in T(\Sigma_i, V)$ and $s =_{E_i} t$ for $i = 1$ or $i = 2$,
and
$\mathsf{h}(x \equiv s) \leq \mathsf{h}(y \equiv t)$.

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

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

FIG. 2 The Transformation Rules.

every occurrence of the variable x by the term t in the set T.[8]

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 optimized, complete implementations of the combination procedure.

We prove in Section 3.3 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.

3.3. 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, for all $j > 0$, we will denote by $\prec_j$ the relation $\prec$ on the equational part of S_j (cf. Definition 3.1).

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 3.2) will be facilitated by the following lemma, whose simple proof is omitted.

LEMMA 3.5. *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 $<_a$,[9] and consider the relations*

$$\begin{array}{rcl} <_1 & := & <_a \cup \{\langle d, e\rangle \mid d < a, b < e\} \\ <_2 & := & <_a \cup \{\langle d, b\rangle \mid d < a\}. \end{array}$$

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

Since the proof of the next lemma will be re-used also in the case of non-disjoint signatures, we will not assume in this proof that $\Sigma := \Sigma_1 \cap \Sigma_2$ is empty.

LEMMA 3.6. *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 definition of S_0 and Proposition 3.3. Thus, assuming that $j > 0$ and 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.[10]

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

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

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

[8]Notice that other authors, especially in programming languages theory, would denote the same substitution by $T[t/x]$ instead. We prefer our convention because we find it more intuitive, especially in the case of composed substitutions.

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

[10]Ignoring the trivial case in which S_j coincides with S_{j-1}.

If we take $\prec_{j-1}$ to be the relation $<$ of Lemma 3.5, $x \equiv t$ to be a, and $y \equiv r$ to be b, it is easy to see that $a < b$ and $\prec_j$ coincides with $<_1$ (as defined in the lemma). Now, $<$ is acyclic by induction and $b \not<^* a$ because $a < b$. By Lemma 3.5 then, $\prec_j$ is acyclic. This shows that Condition 2 of Definition 3.2 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 3.2 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 in $T(\Sigma_i, V) \setminus T(\Sigma, V)$; therefore, the replacement of x by r in T occurs only inside terms in $T(\Sigma_i, V) \setminus T(\Sigma, V)$ and produces terms still in $T(\Sigma_i, V) \setminus T(\Sigma, V)$. It follows that S_j satisfies both Condition 1 and 3(ii) of Definition 3.2.

Coll2. The proof is essentially a special case of the one above, with r replaced by y. The proof of Condition 2 of Definition 3.2 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 3.5. If $<$ is acyclic, then its subrelation $<_a$ is acyclic as well.

Ident1. By the rule's definition, S_{j-1} and S_j must have the following form:

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

Moreover, it is *not* the case that $(y \equiv t) \prec^+_{j-1} (x \equiv s)$, otherwise we would have that $\mathsf{h}(y \equiv t) < \mathsf{h}(x \equiv s)$. 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 3.5, 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 3.2. 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, to see that S_j also satisfies Condition 3a, notice that T is obviously unchanged if x does not occur in T. Also, if the height of $y \equiv t$ in S_{j-1} is zero, then the height of $x \equiv s$ is also zero, which means that x does not occur in T. If $\mathsf{h}(y \equiv t) > 0$ and x occurs in T, both s and t are elements of $T(\Sigma_i, V) \setminus T(\Sigma, V)$. But then we can argue that Condition 3b holds for S_j exactly as we did in the case of **Coll1**. It follows that S_j is an abstraction system with disequation $(u \not\equiv v)[x/y]$.

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 $T \subseteq T'$. ∎

Next, we show that the combination procedure always terminates.

LEMMA 3.7. *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 execution of Step 2 also needs only 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 left-hand side variables of 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. ∎

The next two lemmas show that the derivation rules preserve satisfiability.

LEMMA 3.8. *For all $j > 0$ let $\bar{v}_{j-1}$ be a sequence consisting of the left-hand side variables of S_{j-1} and $\bar{v}_j$ be a sequence consisting of the left-hand side variables of S_j. Then, $\exists \bar{v}_{j-1}.S_{j-1} \leftrightarrow \exists \bar{v}_j.S_j$ is valid in E.*

Proof. We can index all the possible cases by the derivation rule applied to S_{j-1} to obtain S_j. Let $\mathcal{A}$ be any model of E.

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 $i = 2$.

Let α be a valuation of V satisfying S_{j-1} in $\mathcal{A}$. It is enough to show that there exists a valuation α' that satisfies S_j in $\mathcal{A}$ and coincides with α on the free variables of $\exists \bar{v}_{j-1}.S_{j-1} \leftrightarrow \exists \bar{v}_j.S_j$.

Since $y \equiv t$ is valid in E, for being valid in E_i, α must assign both x and y with $[\![t]\!]^{\mathcal{A}}_{\alpha}$, 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 $\alpha(y) = [\![r]\!]^{\mathcal{A}}_{\alpha}$. It follows immediately that α satisfies S_j in $\mathcal{A}$. Thus, we can take $\alpha' := \alpha$.

Now, assume that the valuation α satisfies S_j in the model $\mathcal{A}$ of E. Again, we must show that there exists a valuation α' that satisfies S_{j-1} in $\mathcal{A}$ and coincides with α on the free variables of $\exists \bar{v}_{j-1}.S_{j-1} \leftrightarrow \exists \bar{v}_j.S_j$.

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 $T_1 \cup T_2$. Since α and α' differ only w.r.t. the value they assign to x, and x is a left-hand side variable in S_{j-1} and does not occur in S_j, this completes the proof that $\exists \bar{v}_{j-1}.S_{j-1} \leftrightarrow \exists \bar{v}_j.S_j$ is valid in E.

The proof for **Coll2** can be derived as a special case of the one for **Coll1** with r replaced by y. **Ident1** can be treated similarly.

When S_j is generated by an application of **Simpl**, 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 satisfied by a valuation α in $\mathcal{A}$, so is S_j. Conversely, assume that S_j is satisfied in $\mathcal{A}$ by some valuation α. Let α' be a valuation coinciding with α on all variables except x. For the variable x, let $\alpha'(x) := [\![t]\!]^{\mathcal{A}}_{\alpha}$. From the assumptions and the fact that S_{j-1} is an abstraction system, we know that x is not in $\mathcal{V}ar(t) \cup \mathcal{V}ar(T)$. This, together with the definition of $\alpha'(x)$, implies that α' satisfies S_{j-1}. In addition, α and α' coincide on the free variables of $\exists \bar{v}_{j-1}.S_{j-1} \leftrightarrow \exists \bar{v}_j.S_j$ since x is a left-hand side variable in S_{j-1} and does not occur in S_j. ∎

The lemma above immediately entails the following weaker lemma (see the comment following Proposition 3.3).

LEMMA 3.9. *For all $j > 0$, the abstraction system S_j is satisfiable in E iff S_{j-1} is satisfiable in E.*

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

PROPOSITION 3.10 (Soundness). *If the combination procedure succeeds on an input* (s_0, t_0), *then* $s_0 =_E t_0$.

Proof. Let $\{S_j \mid j = 0, \ldots, n\}$ be the sequence of abstraction systems generated by the procedure on input (s_0, t_0). By the procedure's definition we know that, if the procedure succeeds, $S_n = \{v \not\equiv v\} \cup T$. Since S_n is clearly unsatisfiable in E, we can conclude by a repeated application of Lemma 3.9 that $S_0 = AS(s_0 \not\equiv t_0)$ is also unsatisfiable in E. By Proposition 3.3, it follows that $s_0 \not\equiv t_0$ is unsatisfiable in E, which means that $s_0 =_E t_0$. ∎

Finally, the combination procedure is also complete.

PROPOSITION 3.11. *The combination procedure succeeds on input* (s_0, t_0) *if* $s_0 =_E t_0$.

A simple proof of Proposition 3.11 can be found in [4]. It is based on the same basic satisfiability result used in [25] to prove the correctness of the Nelson-Oppen combination procedure. In the context of this section, that result states that the union $S_1 \cup S_2$ of a set S_1 of Σ_1-equations and disequations and a set S_2 of Σ_2-equations and disequations is satisfiable in $E_1 \cup E_2$ whenever $S_i \cup \Delta$ is satisfiable in E_i for $i = 1, 2$, where Δ is the set of all disequations between the variables shared by S_1 and S_2.

Since this satisfiability result applies only if E_1 and E_2 have disjoint signatures, the proof of Proposition 3.11 in [4] does not lift to the more general case treated in Section 5. As a consequence, we will provide a completeness proof only for the extension of our combination procedure to that case. The claim in Proposition 3.11 will then follow from the fact that the extended procedure reduces exactly to the procedure seen in this section whenever E_1 and E_2 have disjoint signatures.

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

THEOREM 3.12. *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 for the single theories, our combination procedure is polynomial.

COROLLARY 3.13. *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.*

4. COMBINING NON-DISJOINT EQUATIONAL THEORIES

The rest of this paper is concerned with the question of how the combination result stated in Theorem 3.12 can be lifted to the combination of equational theories whose signatures are not disjoint. As shown in the introduction, in that case 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 establish some useful properties of theories satisfying them. Some of the results in Subsections 4.1 and 4.2 below are closely related to results first described in [26]. We will discuss this relationship in more detail in Section 7.

4.1. Fusions of Algebras

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 $\mathcal{A}$.

When proving properties of a theory E obtained by putting together component theories it is often convenient to use models of E obtained by amalgamating models of the component theories. A simple type of amalgamated model is what [26] calls a *fusion*.

DEFINITION 4.1 (Fusion). A $(\Sigma_1 \cup \Sigma_2)$-algebra $\mathcal{F}$ is a *fusion* of a Σ_1-algebra $\mathcal{A}_1$ and a 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 $\mathcal{A}_2$.

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

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

PROPOSITION 4.2. *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 $E_1 \cup E_2$. ∎

Not every two algebras have fusions. We show below that they do exactly when they have the same cardinality and interpret in the same way the symbols shared by their signatures.

PROPOSITION 4.3. *Let $\mathcal{A}$ be a Σ_1-algebra, $\mathcal{B}$ a Σ_2-algebra, and $\Sigma := \Sigma_1 \cap \Sigma_2$. Then, $\mathit{Fus}(\mathcal{A}, \mathcal{B}) \neq \emptyset$ iff $\mathcal{A}^{\Sigma}$ is Σ-isomorphic to $\mathcal{B}^{\Sigma}$.*

Proof. ($\Rightarrow$) Let $\mathcal{F} \in \mathit{Fus}(\mathcal{A}, \mathcal{B})$. 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 $(\Sigma_1 \cup \Sigma_2)$-algebra $\mathcal{F}$ 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 $b_1, \ldots, b_n \in B$,

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

[11] But note that $\mathit{Fus}(\mathcal{A}_1, \mathcal{A}_2)$ may contain non-isomorphic algebras.

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 Σ_2-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 4.1 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

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

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

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

∎

The proof of the proposition above also shows that every (Σ-)isomorphism between the Σ-reducts of two algebras $\mathcal{A}_1$ and $\mathcal{A}_2$ to their common signature Σ induces a *canonical* fusion of $\mathcal{A}_1$ and $\mathcal{A}_2$. We will use this sort of fusion in many of the proofs to follow.

COROLLARY 4.4. *Let Σ_1 and Σ_2 be two functional signatures with intersection $\Sigma := \Sigma_1 \cap \Sigma_2$. For $i = 1, 2$ let $\mathcal{A}_i$ be a Σ_i-algebra. Then, for every isomorphism h of ${\mathcal{A}_1}^{\Sigma}$ onto ${\mathcal{A}_2}^{\Sigma}$, there is a fusion $\mathcal{A}$ of $\mathcal{A}_1$ and $\mathcal{A}_2$ such that*

- *h is a Σ_1-isomorphism of $\mathcal{A}_1$ onto $\mathcal{A}^{\Sigma_1}$,*
- *the identity mapping on A_2 is a Σ_2-isomorphism of $\mathcal{A}_2$ onto $\mathcal{A}^{\Sigma_2}$.*

4.2. Theories Admitting Constructors

In the rest of the paper we will focus on equational theories whose free models over infinitely many generators have certain reducts that are themselves free. Now, 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

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

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 all the negative ones with predecessor function. Now, for $\Sigma := \{\mathsf{s}\}$, $\mathcal{Z}^{\Sigma}$ is definitely not free because it does not even admit a non-redundant set of generators,[12] which is a necessary condition for an algebra to be free.

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

Nonetheless, there are free algebras some of whose reducts, although requiring a possibly larger set of generators, are still free. In that case, we say that their equational theory admits *constructors*. A formal definition of this notion of constructors is given below.

In the following, Ω will be a countable functional signature, and Σ a subset of Ω. We will fix a non-trivial equational theory E over Ω and 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 4.5 (Constructors). The subsignature Σ of Ω is a *set of constructors for* E iff for every Ω-algebra $\mathcal{A}$ free in E over a countably infinite set X, $\mathcal{A}^{\Sigma}$ is free in E^{Σ} over a set Y including X.

It is immediate that the whole signature Ω is a set of constructors for the theory E. Similarly, the empty signature is a set of constructors for E, as any model of E is free over its whole carrier in the restriction $E^{\emptyset}$, which is just $\{v \equiv v \mid v \in V\}$. The constant symbols of Ω are easily shown to be a set of constructors for E. Also, when E is axiomatized by the union of two theories E_1, E_2 of respective, *disjoint* signatures, Σ_1, Σ_2, Σ_i $(i = 1, 2)$ is a set of constructors for E. This is not immediate but it can be shown as a consequence of some results in [2].

The abstractness of Definition 4.5 may make it difficult to say for a given theory E and signature Σ whether Σ is a set of constructors for E. For this reason we provide in the following a more concrete, syntactic characterization of theories admitting constructors. But first, some more notation is necessary.

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 t of $T(\Sigma, G)$ is obtained from a term $s(\bar{v}) \in T(\Sigma, V)$ by replacing the variables $\bar{v}$ of s with terms from G. In accordance with our notational conventions, we will denote such a term t by $s(\bar{r})$ where $\bar{r}$ is the tuple made, without repetitions, of the terms of G that replace the variables $\bar{v}$. We will refer to these terms as the *G-variables* of t. Notice that the 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.

DEFINITION 4.6 (Σ-base). A subset G of $T(\Omega, V)$ is a Σ-*base of* 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 $s_1(\bar{r}_1), s_2(\bar{r}_2) \in T(\Sigma, G)$,
$$s_1(\bar{r}_1) =_E s_2(\bar{r}_2) \quad \text{iff} \quad s_1(\bar{v}_1) =_E s_2(\bar{v}_2),$$
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.

We say that E *admits a Σ-base* if some subset G of $T(\Omega, V)$ is a Σ-base of E.

THEOREM 4.7 (Characterization of constructors). *The signature Σ is a set of constructors for E iff E admits a Σ-base.*

Proof. Let $\mathcal{A}$ be an Ω-algebra free in E over some countably infinite set X, and α any bijective valuation of V onto X.[13]

($\Rightarrow$) Assume that Σ is a set of constructors for E, which implies that $\mathcal{A}^\Sigma$ is free in E^Σ over some set Y such that $X \subseteq Y$. First notice that, since $\mathcal{A}$ is generated by X, for every element y of Y there is a term r in $T(\Omega, V)$ such that $y = [\![r]\!]^{\mathcal{A}}_\alpha$. Then let

$$G \quad := \quad \{r \in T(\Omega, V) \mid [\![r]\!]^{\mathcal{A}}_\alpha \in Y\}.$$

We show that G is a Σ-base of E.

Since $X \subseteq Y$, it is immediate that every $v \in V$ is in G, which means that G satisfies the first condition in Definition 4.6. The second condition easily follows from the fact that $\mathcal{A}^\Sigma$ is Σ-generated by Y. Similarly, the third condition follows from Point 3 of Proposition 2.1.

($\Leftarrow$) Where G is any Σ-base of E, let

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

Since $V \subseteq G$ by definition of Σ-base, it is immediate that $X \subseteq Y$. We show that $\mathcal{A}^\Sigma$ is free in E^Σ over Y.

Let us 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 of Definition 4.6, the term $t \in T(\Omega, V)$ is equivalent in 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 2.1. To show that it is free in E^Σ 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 free in E over X and α is injective as well we can conclude by Point 3 of Proposition 2.1 that $s_1(\bar{r}_1) =_E s_2(\bar{r}_2)$.

Because of the assumption that 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 around, 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 $\bar{r}_1, \bar{r}_2$ so that two terms are abstracted by the same variable iff they are equivalent in E. By Point 3 of Definition 4.6 then we obtain that $s_1(\bar{v}_1) =_E s_2(\bar{v}_2)$. Considering that the terms $s_1(\bar{v}_1), s_2(\bar{v}_2)$ are Σ-terms, this is the same as saying that $s_1(\bar{v}_1) =_{E^\Sigma} s_2(\bar{v}_2)$. ∎

We will use sets such as the set Y defined in the proof of the if-direction above often enough to justify the following notation. If T is a subset of $T(\Omega, V)$, $\mathcal{A}$ an Ω-algebra free in E over a countably-infinite set X, and α a bijective valuation of

[13] Such a valuation α exists since V is assumed to be countably infinite.

V onto X we will denote by $[\![T]\!]_\alpha^\mathcal{A}$ the set of element of $\mathcal{A}$ denoted by the terms of T, i.e., $[\![T]\!]_\alpha^\mathcal{A} := \{[\![t]\!]_\alpha^\mathcal{A} \mid t \in T\}$.

From the proof of Theorem 4.7 we can also conclude that a Σ-base actually denotes a set of generators for the Σ-reduct of the E-free algebra.

COROLLARY 4.8. *Let G be a Σ-base of E, $\mathcal{A}$ an Ω-algebra free in E over a countably infinite set X, and α a bijective valuation of V onto X. Then, $\mathcal{A}^\Sigma$ is free in E^Σ over the set $Y := [\![G]\!]_\alpha^\mathcal{A}$, and $X \subseteq Y$.*

It should be clear that a theory E with constructors Σ admits many Σ-bases. For instance, if G is a Σ-base of E, any set equal to G modulo equivalence in E is also a Σ-base of E. It is still an open question, however, whether a theory may have *essentially* different Σ-bases.[14] For now, we only know that this is impossible if the theory's restriction to Σ is collapse-free.

PROPOSITION 4.9. *Assume that Σ is a set of constructors for E and E^Σ is collapse-free. Then, every Σ-base of E is equal modulo equivalence in E to the set*

$$G_E(\Sigma, V) \quad := \quad \{r \in T(\Omega, V) \mid r \neq_E t \text{ for all } t \in T(\Omega, V) \text{ with } t(\epsilon) \in \Sigma\}.$$

Proof. Let G be a Σ-base of E. We prove the claim by showing that (a) every element of G is in $G_E(\Sigma, V)$ and (b) every element of $G_E(\Sigma, V)$ is equivalent in E to some element of G.

(a) Let $r \in G$ and $t \in T(\Omega, V)$ with $t(\epsilon) \in \Sigma$. It is enough to show that $r \neq_E t$. Assume the contrary. Then, since G is a Σ-base of E and $t(\epsilon) \in \Sigma$, there is a term $s(\bar{r}) \in T(\Sigma, G)$ with s *non-variable* such that $r =_E s(\bar{r})$. By Condition 3 of Definition 4.6 then, there is a variable v and a tuple $\bar{v}$ of variables such that $v =_E s(\bar{v})$. But this contradicts the assumptions that E is non-trivial and E^Σ is collapse-free.

(b) Let $t \in G_E(\Sigma, V)$. By Condition 2 of Definition 4.6, there is a term $s(\bar{r}) \in T(\Sigma, G)$ such that $t =_E s(\bar{r})$. Since t is equivalent in E to no terms starting with a Σ-symbol, s is necessarily a variable and $\bar{r}$ is actually the one-element tuple (r) for some $r \in G$. It follows that $t =_E r$. ∎

Proposition 4.9 also entails that, whenever Σ is a set of constructors and E^Σ is collapse-free, the set $G_E(\Sigma, V)$ above is the largest Σ-base of E. That it is one follows from the fact that, in this case, E admits a Σ-base G, and that this Σ-base is equal to G modulo equivalence in E by the proposition. That it is the largest is just what we have shown in part (a) of the above proof.

Examples

We provide below some examples of equational theories admitting constructors in the sense of Definition 4.5. But first, let us consider some immediate counter-examples:

- The signature $\Sigma := \{\mathsf{s}\}$ is not a set of constructors for the theory E axiomatized by $\{x \equiv \mathsf{p}(\mathsf{s}(x)),\ x \equiv \mathsf{s}(\mathsf{p}(x))\}$. As argued at the beginning of this section for the case of one generator, in constrast with the definition of constructors, the Σ-reduct of any free model of E over a countably infinite set is not itself free, because it does not admit a non-redundant set of generators.

[14] In the sense of not denoting the same set Y in Corollary 4.8.

- The signature $\Sigma := \{f\}$ is not a set of constructors for the theory E axiomatized by $\{g(x) \equiv f(g(x))\}$. In fact, since E^Σ is clearly collapse-free we know that any Σ-base of E, if any, is included in the set $G_E(\Sigma, V)$ defined in Proposition 4.9. But $G_E(\Sigma, V)$ is simply V in this case, and it is immediate that no subset of V satisfies Condition 2 of Definition 4.6.

- Finally, the signature $\Sigma := \{f\}$ is not a set of constructors for theory E axiomatized by $\{f(g(x)) \equiv f(f(g(x)))\}$. Again, E^Σ is clearly collapse-free. Moreover, $G_E(\Sigma, V) = V \cup \{g(t) \mid t \in T(\Omega, V)\}$. It is easy to see that Conditions 1 and 2 of Definition 4.6 hold for $G_E(\Sigma, V)$. However, Condition 3 does not since $f(g(x)) =_E f(f(g(x)))$, although $f(y) \neq_E f(f(y))$.

EXAMPLE 4.1. 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:

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

The signature $\Sigma := \{0, \mathsf{s}\}$ is a set of constructors for E_1 in the sense of Definition 4.5. A direct proof of this can be found in [4]. Here, we will obtain it later as a consequence of a more general result discussed in Section 7.2.

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

EXAMPLE 4.2. Consider the signature $\Sigma_2 := \{0, 1, \mathsf{rev}, \cdot\}$ and the equational theory E_2 axiomatized by the equations below:

$$\begin{array}{rcl} x \cdot (y \cdot z) & \equiv & (x \cdot y) \cdot z, \\ \mathsf{rev}(0) & \equiv & 0, \\ \mathsf{rev}(1) & \equiv & 1, \\ \mathsf{rev}(x \cdot y) & \equiv & \mathsf{rev}(y) \cdot \mathsf{rev}(x), \\ \mathsf{rev}(\mathsf{rev}(x)) & \equiv & x. \end{array} \tag{3}$$

Note that orienting the equations from left to right yields a canonical term rewriting system R_2. Let us denote the normal form of a term t w.r.t. this rewrite system by $t{\downarrow_{R_2}}$. It is easy to see that the restriction of E_2 to $\Sigma' := \{0, 1, \cdot\}$ is axiomatized by the first equation above.

We show that the signature Σ' is a set of constructors for E_2 in the sense of Definition 4.5, by showing that the set

$$G \;:=\; V \cup \{\mathsf{rev}(v) \mid v \in V\}$$

is a Σ'-base of E_2.

It is immediate from the definition of G that $V \subseteq G$, and thus Condition 1 of Definition 4.6 is satisfied by E_2 and Σ'. To see that Condition 2 is satisfied, it is sufficient to show that the R_2-normal form of any term $t \in T(\Sigma_2, V)$ is of the form

$$t{\downarrow_{R_2}} = (\cdots((r_1 \cdot r_2) \cdot r_3) \cdot \;\ldots\; \cdot r_k)$$

where $r_i \in \{0,1\} \cup V \cup \{\mathsf{rev}(v) \mid v \in V\}$. This can be easily proved by showing that, to any term not in this form, one of the rules of R_2 applies.

To see that Condition 3 of Definition 4.6 holds, we consider a term $s(\bar{r}) \in T(\Sigma', G)$—where $s(\bar{v})$ is a Σ'-term and every element of $\bar{r}$ belongs to G. It is easy to see that the R_2-normal form of $s(\bar{r})$ can be obtained by computing the normal form of $s(\bar{v})$ w.r.t. the rewrite rule $x \cdot (y \cdot z) \to (x \cdot y) \cdot z$, and then inserting into this term the terms in $\bar{r}$. Now, Condition 3 of Definition 4.6 is an easy consequence of this fact.

In the examples above, the restriction of each theory to the constructor symbols is collapse-free. That is not the case for the theory in the next example.

EXAMPLE 4.3. Consider the signature $\Sigma_3 := \{0, \mathsf{p}, \mathsf{s}, -\}$ and the equational theory E_3 axiomatized by the equations:

$$\begin{array}{rcl} \mathsf{s}(\mathsf{p}(x)) & \equiv & x \\ \mathsf{p}(\mathsf{s}(x)) & \equiv & x \\ -0 & \equiv & 0, \\ -(-x) & \equiv & x, \\ -\mathsf{s}(x) & \equiv & \mathsf{p}(-x), \\ -\mathsf{p}(x) & \equiv & \mathsf{s}(-x). \end{array} \tag{4}$$

The signature $\Sigma'' := \{0, \mathsf{p}, \mathsf{s}\}$ is a set of constructors for E_3. To prove it we show that the set $G := V \cup \{-v \mid v \in V\}$ is a Σ''-base of E_3.

By definition, $V \subseteq G$. To show the remaining two conditions of Definition 4.6, note that orienting the axioms above from left to right produces a confluent and terminating rewrite system R_3. Thus, two terms are equal modulo E_3 iff their R_3-normal forms are syntactically identical.

Now, Condition 2 of Definition 4.6 is satisfied since, given an Σ_3-term, its R_3-normal form is in $T(\Sigma'', G)$. This is an immediate consequence of the fact that (because of the last four rules of R_3) any term containing a minus symbol in front of $-$, 0, p, or s is R_3-reducible. Therefore, in R_3-normal forms, minus can only occur in front of variables.

All we need to show then is that Condition 3 of Definition 4.6 is also satisfied. Thus, let $s_1(\bar{r}_1), s_2(\bar{r}_2)$ be terms in $T(\Sigma'', G)$ such that $s_1(\bar{r}_1) =_{E_3} s_2(\bar{r}_2)$. Since R_3 is confluent and terminating, there exists a term t such that $s_1(\bar{r}_1) \xrightarrow{*}_{R_3} t$ and $s_2(\bar{r}_2) \xrightarrow{*}_{R_3} t$. Since in the terms $s_1(\bar{r}_1), s_2(\bar{r}_2)$ (as well as in any term occurring in the reduction chains) the minus symbol can only occur in front of variables, the reduction chains make use of the first two rules of R_3 only. Consequently, $s_1(\bar{r}_1)$ and $s_2(\bar{r}_2)$ are equal modulo the first two axioms of E_3. Given that these axioms do not contain the minus symbol, it is easy to see that this implies that $s_1(\bar{v}_1) =_{E_3} s_2(\bar{v}_2)$. Since the other direction of the bi-implication of Condition 3 is trivial, this completes the proof that $G = V \cup \{-v \mid v \in V\}$ is a Σ''-base of E_3.

More examples of theories with constructors can be found in the usual axiomatizations of abstract data types.

Normal Forms

Let us now assume that E is an equational theory over the signature Ω, which has a set of constructors Σ. Let G be a Σ-base for E.

According to Definition 4.6, every Ω-term t is equivalent in E to a term $s(\bar{r}) \in T(\Sigma, G)$. We call $s(\bar{r})$ *a G-normal form of t in E*.[15] We say that a term $t \in T(\Omega, V)$ is *in G-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 G-normal form, as are terms in G. We will say just *normal form* instead of G-normal form whenever the Σ-base G in question is clear from the context or irrelevant.

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

DEFINITION 4.10 (Computable Normal Forms). We say that *G-normal forms are computable for Σ and E* if there is a computable function

$$NF_G \colon T(\Omega, V) \longrightarrow T(\Sigma, G)$$

such that $NF_G(t)$ is a G-normal form of t, i.e., $NF_G(t) =_E t$.

Note that the terms of G may as well start with a Σ-symbol themselves.[16] This means that, for any given term t in G-normal form, it may not be possible to effectively identify its G-variables, i.e., those terms $\bar{r}$ of G such that $t = s(\bar{r})$ for some Σ-term s. Now, in the combination procedure introduced in Section 5, sometimes we will need to first compute the normal form $s(\bar{r})$ of a term and then decompose this normal form into its components s and $\bar{r}$. To be able to do this it will be enough to assume (in addition to the computability of normal forms) that G is a recursive set, thanks to the proposition below.

PROPOSITION 4.11. *When G is recursive, for every $t \in T(\Sigma, G)$ there is an effective way of computing from t a term $s(\bar{v}) \in T(\Sigma, V)$ and a sequence $\bar{r}$ of terms in G such that $t = s(\bar{r})$.*

Proof. Let $t \in T(\Sigma, G)$. We prove by structural induction that we can identify a Σ-term $s(\bar{v})$ and a tuple $\bar{r}$ of terms in G such that $t = s(\bar{r})$.[17]

(Base Case) If $t \in V$ the claim is trivially true because $t \in G$ by the definition of Σ-bases.

(Inductive Step) Let t be the term $f(t_1, \ldots, t_n)$ with $f \in \Omega$. If t is in G, which we can effectively check because G is recursive, we can choose any $s \in V$ and let $\bar{r}$ be made of just t itself. If t is in not in G, then f must be a Σ-symbol since $t \in T(\Sigma, G)$ by assumption. Also, the terms $t_1, \ldots, t_n$ must belong to $T(\Sigma, G)$ or else t would not be an element of $T(\Sigma, G)$. For $j \in \{1, \ldots, n\}$, let $s_j(\bar{r}_j)$ be an appropriate decomposition of the term t_j into a Σ-term s_j and a tuple $\bar{r}_j$ of elements of G. This decomposition is computable by induction. Let $f(s_1, \ldots, s_n)(\bar{v})$ be the term obtained from t by replacing with fresh variables $\bar{v}$ all the occurrences in t of the terms in $\bar{r}_1, \ldots, \bar{r}_n$ so that identical occurrences are replaced by the same variable. Where $\bar{r}$ consists, in order, of the terms of G abstracted by $\bar{v}$, it is immediate that $s(\bar{v}) = f(s_1, \ldots, s_n)(\bar{v}) \in T(\Sigma, V)$, $\bar{r}$ is a tuple of elements of G, and $t = s(\bar{r})$. ∎

If a term $t \in T(\Omega, V)$ is equivalent in E to a Σ-term s, then s is a normal form of t. On the other hand, not every normal form of t needs to be a Σ-term. In our

[15] Notice that in general a term may have more than one G-normal form.

[16] Unless E^{Σ} is collapse-free (cf. Proposition 4.9).

[17] Note that this decomposition of t need not be unique since terms in G may start with a Σ symbol.

combination procedure, however, it will be convenient to assume that every given normal form function returns a Σ-term whenever its input term is equivalent to one. The following lemma implies that this assumption can be made without loss of generality.

LEMMA 4.12. *Let the word problem for E be decidable and G-normal forms computable for Σ and E. Then, for all $t \in T(\Omega, V)$ it is decidable whether t is equivalent in E to a Σ-term. If this is the case, a term $s \in T(\Sigma, V)$ such that $t =_E s$ is effectively computable from t.*

Proof. Let us say that a term t is *independent in E* from one of its variables v if substituting v by a fresh variable (i.e., a variable not occurring in t) yields a term equivalent to t in E. Now, let $t \in T(\Omega, V)$ and $s(\bar{r}) = NF_G(t)$ with $\bar{r} = (r_1, \ldots, r_m)$. Since the word problem for E is decidable, we can assume with no loss of generality that all the elements in $\bar{r}$ are pairwise inequivalent in E—otherwise we can effectively replace by a single representative term all those that are not.

Let $s(\bar{v})$ with $\bar{v} = (v_1, \ldots, v_m)$ be the Σ-term obtained from $s(\bar{r})$ by replacing the occurrences of r_j in $s(\bar{r})$ by a fresh variable v_j for every $j \in \{1, \ldots, m\}$. Then let $\bar{q} := (q_1, \ldots, q_m)$ where, for each $j \in \{1, \ldots, m\}$, $q_j := u_j$ if u_j is a variable such that $u_j =_E r_j$, $q_j := v_j$ if $s(\bar{v})$ is independent from v_j in E, and $q_j := r_j$ otherwise. Since E is non-trivial and has a decidable word problem, the tuple $\bar{q}$ is effectively constructible. Moreover, its elements are pairwise inequivalent and each of them is equivalent in E to a variable only if it is one.

Now consider the term $s(\bar{q}) \in T(\Sigma, G)$ obtained from $s(\bar{v})$ by substituting v_j by q_j for all $j \in \{1, \ldots, m\}$. By construction, we have $s(\bar{q}) =_E s(\bar{r}) =_E t$. We prove below that whenever t is equivalent in E to a Σ-term, each element of $\bar{q}$ is in fact a variable and so $s(\bar{q}) \in T(\Sigma, V)$. Conversely, if $s(\bar{q}) \in T(\Sigma, V)$, then t is obviously equivalent to a Σ-term. Since $s(\bar{q})$ is effectively computable from t, this will conclude our proof.

Assume that $t =_E s_2(\bar{r}_2)$ for some $s_2(\bar{r}_2) \in T(\Sigma, V)$. Since t is equivalent in E to $s(\bar{q})$, we have that $s(\bar{q}) =_E s_2(\bar{r}_2)$. Given that G is a Σ-base of E, we also have that $s(\bar{v}_1) =_E s_2(\bar{v}_2)$, for some tuples $\bar{v}_1, \bar{v}_2$ of fresh variables abstracting the elements of $\bar{q}, \bar{r}_2$ as in Condition 3 of Definition 4.6. Recalling that only equivalent terms get abstracted by the same variable, we can then conclude that $\bar{q}$ contains only variables. In fact, let q_j be an element of $\bar{q}$ and let v_{q_j} be the variable of $\bar{v}_1$ abstracting q_j. If v_{q_j} occurs in $\bar{v}_2$, it is because q_j is equivalent in E to an element of $\bar{r}_2$. Since every element of $\bar{r}_2$ is a variable, it follows by construction of $\bar{q}$ that q_j is a variable. If v_{q_j} does not occur in $\bar{v}_2$, the equivalence $s(\bar{v}_1) =_E s_2(\bar{v}_2)$ entails that $s(\bar{v}_1)$ is independent from v_{q_j} in E. Now, $\bar{v}_1$ is just a bijective renaming of $\bar{v}$ given that the elements of $\bar{q}$ are pairwise inequivalent in E. It follows that $s(\bar{v})$ is independent from v_j, the variable corresponding to v_{q_j} in the renaming. But then $q_j = v_j$ by construction of $\bar{q}$. ∎

From now on, we will make the following assumptions on the functions computing normal forms.

Assumption 4.1. The computed normal form $s(\bar{r})$ of a term t is always in $T(\Sigma, V)$ if t is equivalent to a Σ-term in the theory E in question. Moreover, the elements of $\bar{r}$ are pairwise inequivalent in E, with the non-variable ones non-collapsing in E.

As we have seen above, all these assumptions can be made without loss of generality whenever E is non-trivial, normal forms are computable and the word problem is decidable in E.

We are interested in theories admitting constructors because, under the right conditions, the decidability of the word problem is modular with respect to their union. We start looking at these conditions and some of their implications in the next subsection.

4.3. Combination of Theories Sharing Constructors

Going back to the problem of combining theories, let us now consider two non-trivial equational theories E_1, E_2 with respective signatures Σ_1, Σ_2 such that, for $i = 1, 2$

- $\Sigma := \Sigma_1 \cap \Sigma_2$ is a set of constructors for E_i;
- ${E_1}^{\Sigma} = {E_2}^{\Sigma}$;
- E_i admits a recursive Σ-base G_i closed under bijective renaming of V;
- G_i-normal forms are computable for Σ and E_i by a function NF_i that satisfies Assumption 4.1.
- the word problem for E_i is decidable.

Of the above assumptions on E_i, only the closure of G_i under bijective renaming has not been mentioned before. We need this assumptions for technical reasons in the remainder of this paper, but we have not been able to show so far that it is without loss of generality. Even if it is a real restriction, however, it appears to be a rather mild one, which is satisfiable in all the examples of theories with constructors we can think of, including those given above.

As before, let

$$E \quad := \quad E_1 \cup E_2.$$

In the rest of this section, we prove a number of important facts about E. We will use these facts in the next two sections to show that, under the above assumptions on E_1 and E_2, E has a decidable word problem and admits a recursive Σ-base with computable normal forms. A very useful tool for our proofs will be a specific model of E, obtained by a fusion of the free models of E_1 and E_2 as described below.

In what follows, if S is any set, $Card(S)$ will denote the cardinality of S.

A Fusion Model for E

For $i = 1, 2$, let us fix a Σ_i-algebra $\mathcal{A}_i$ free in E_i over a countably infinite set X_i. Let us also fix an arbitrary bijective valuation α_i of V onto X_i, and consider the set

$$Y_i \quad := \quad [\![G_i]\!]^{\mathcal{A}_i}_{\alpha_i} \ .$$

We know from Corollary 4.8 that $X_i \subseteq Y_i$ and ${\mathcal{A}_i}^{\Sigma}$ is free in ${E_i}^{\Sigma}$ over Y_i. Observe that $\mathcal{A}_i$ is countably infinite, given our assumption that X_i is countably infinite and Σ_i is countable. As a consequence, Y_i is countably infinite as well.

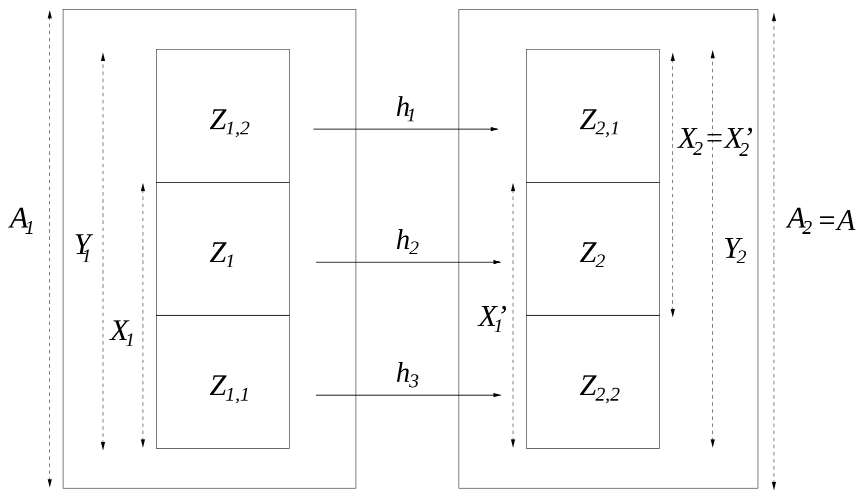

FIG. 3 The Fusion $\mathcal{A}$ of $\mathcal{A}_1$ and $\mathcal{A}_2$.

Now let $Z_{i,2} := Y_i \setminus X_i$ for $i = 1, 2$, and let $\{Z_{1,1}, Z_1\}$ be a partition of X_1 such that Z_1 is countably infinite and $Card(Z_{1,1}) = Card(Z_{2,2})$.[18] Similarly, let $\{Z_{2,1}, Z_2\}$ be a partition of X_2 such that $Card(Z_{2,1}) = Card(Z_{1,2})$ and Z_2 is countably infinite. Then consider 3 arbitrary bijections

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

as shown in Figure 3. Observing that $\{Z_{i,1}, Z_i, Z_{i,2}\}$ is a partition of Y_i for $i = 1, 2$, it is immediate that $h_1 \cup h_2 \cup h_3$ is a well-defined bijection of Y_1 onto Y_2. This bijection induces a fusion of $\mathcal{A}_1$ and $\mathcal{A}_2$, whose main the properties are listed in the lemma below.

Lemma 4.13. *The algebras $\mathcal{A}_1$ and $\mathcal{A}_2$ admit a fusion $\mathcal{A}$ such that:*

1. *$\mathcal{A}^{\Sigma_1}$ is free in E_1 over $X_1' := Z_{2,2} \cup Z_2$;*
2. *$\mathcal{A}^{\Sigma_2}$ is free in E_2 over $X_2' := Z_{2,1} \cup Z_2$;*
3. *$\mathcal{A}^{\Sigma}$ is free in ${E_1}^{\Sigma} = {E_2}^{\Sigma}$ over $Y_2 = Z_{2,1} \cup Z_2 \cup Z_{2,2}$.*
4. *$Y_2 = [\![G_2]\!]^{\mathcal{A}^{\Sigma_2}}_{\alpha_2} = [\![G_1]\!]^{\mathcal{A}^{\Sigma_1}}_{h \circ \alpha_1}$, for some Σ-isomorphism h of ${\mathcal{A}_1}^{\Sigma}$ onto ${\mathcal{A}_2}^{\Sigma}$.*

Proof. Since ${E_1}^{\Sigma} = {E_2}^{\Sigma}$ and both Y_1 and Y_2 are countably infinite, ${\mathcal{A}_1}^{\Sigma}$ and ${\mathcal{A}_2}^{\Sigma}$ are both free in the same Σ-variety over sets with the same cardinality. By well-known results from Universal Algebra[19] then, the bijection $h_1 \cup h_2 \cup h_3 \colon Y_1 \longrightarrow Y_2$ can be extended to a Σ-isomorphism h of ${\mathcal{A}_1}^{\Sigma}$ onto ${\mathcal{A}_2}^{\Sigma}$. It follows from Corollary 4.4 that there is a fusion $\mathcal{A}$ of $\mathcal{A}_1$ and $\mathcal{A}_2$ such that the identity on the carrier of $\mathcal{A}_2$ is a Σ_2-isomorphism of $\mathcal{A}_2$ onto $\mathcal{A}^{\Sigma_2}$, and h is a Σ_1-isomorphism of $\mathcal{A}_1$ onto $\mathcal{A}^{\Sigma_1}$.

The first three points then are an immediate consequence of the construction of h and the choice of $\mathcal{A}$.

[18]This is possible because $Z_{2,2}$ is countable (possibly finite).
[19]See, e.g., [1], Theorem 3.3.3.

Now, $Y_2 = [\![G_2]\!]_{\alpha_2}^{\mathcal{A}^{\Sigma_2}}$ because $\mathcal{A}_2$ and $\mathcal{A}^{\Sigma_2}$ coincide by construction of $\mathcal{A}$ and $Y_2 = [\![G_2]\!]_{\alpha_2}^{\mathcal{A}_2}$ by definition.

Finally, we show that for each $r \in G_1$ we have $[\![r]\!]_{h\circ\alpha_1}^{\mathcal{A}^{\Sigma_1}} = h([\![r]\!]_{\alpha_1}^{\mathcal{A}_1})$. This implies then that $[\![G_1]\!]_{h\circ\alpha_1}^{\mathcal{A}^{\Sigma_1}} = h([\![G_1]\!]_{\alpha_1}^{\mathcal{A}_1}) = h(Y_1) = Y_2$. Thus, let $r(\bar{v}) \in G_1$. We have[20]

$$\begin{array}{rcll}
[\![r(\bar{v})]\!]_{h\circ\alpha_1}^{\mathcal{A}^{\Sigma_1}} & = & r^{\mathcal{A}^{\Sigma_1}}(h(\alpha_1(\bar{v}))) & \text{(by definition of term function)} \\
& = & h(h^{-1}(r^{\mathcal{A}^{\Sigma_1}}(h(\alpha_1(\bar{v}))))) & \text{(since } h \text{ is a bijection)} \\
& = & h(r^{\mathcal{A}_1}(\alpha_1(\bar{v}))) & \text{(since } h^{-1} \text{ is a } \Sigma_1\text{-isomorphism)} \\
& = & h([\![r(\bar{v})]\!]_{\alpha_1}^{\mathcal{A}_1}). &
\end{array}$$

∎

For being a fusion of a model of E_1 and a model of E_2, the algebra $\mathcal{A}$ above is a model of $E = E_1 \cup E_2$ by Proposition 4.2. The first interesting fact we can prove about E using $\mathcal{A}$ is that E is a conservative extension of both E_1 and E_2.

PROPOSITION 4.14. *For all $j \in \{1,2\}$ and $t_1, t_2 \in T(\Sigma_j, V)$*

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

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

Consider then the algebra $\mathcal{A}$ as described in Lemma 4.13, and recall that $\mathcal{A}^{\Sigma_2}$ is free in E_2 over X_2'. Since $t_1 =_E t_2$ and $\mathcal{A}$ is a model of E, we have that $\mathcal{A}, \alpha \models t_1 \equiv t_2$ for any valuation α of $\mathcal{V}ar(t_1 \equiv t_2)$ into A. 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 $\mathcal{A}^{\Sigma_2}, \alpha \models t_1 \equiv t_2$. It follows by Proposition 2.1 that $t_1 =_{E_2} t_2$. ∎

The following is an immediate consequence of the above result.

COROLLARY 4.15. *E is non-trivial and $E^{\Sigma} = {E_1}^{\Sigma} = {E_2}^{\Sigma}$.*

Another important property of E is represented by the interpolation result in Lemma 4.18 below. To prove that result we will need some more properties of the algebra $\mathcal{A}$ defined in the proof of Lemma 4.13.

LEMMA 4.16. *Let $i \in \{1,2\}$ and r a term of $G_i \setminus V$ non-collapsing in E. Then,*

$$[\![r]\!]_{\alpha}^{\mathcal{A}} \in Z_{2,i}$$

for every injective valuation α of $\mathcal{V}ar(r)$ into X_i'.

Proof. First let $i = 2$ and so let $r \in G_2 \setminus V$ be non-collapsing in E. We start by showing that $[\![r]\!]_{\alpha}^{\mathcal{A}} \in Y_2$. Since α is an injective valuation of $\mathcal{V}ar(r)$ into X_2', and the valuation α_2 is a bijection of V into X_2', there is a term r' obtained by a bijective renaming of the variables in r such that $[\![r]\!]_{\alpha}^{\mathcal{A}} = [\![r']\!]_{\alpha_2}^{\mathcal{A}_2}$. Since G_2 is closed under renaming by our assumptions, we have that $r' \in G_2$, and thus $[\![r']\!]_{\alpha_2}^{\mathcal{A}_2} \in Y_2$ by definition of Y_2. Now we prove by contradiction that $[\![r]\!]_{\alpha}^{\mathcal{A}} \notin X_2'$. If $[\![r]\!]_{\alpha}^{\mathcal{A}} \in X_2'$, it is easy to show that there is a $v \in V$ and an injective valuation γ of $\mathcal{V}ar(v \equiv r)$ into X_2' such that $\mathcal{A}, \gamma \models v \equiv r$. Recalling that $\mathcal{A}^{\Sigma_2}$ is free in E_2 over X_2' we then obtain

[20] In the identities below, an expression like $\alpha_1(\bar{v})$ should be read as an abbreviation for $(\alpha_1(v_1), \ldots, \alpha_1(v_m))$ where $\bar{v} = (v_1, \ldots, v_m)$.

by Proposition 2.1 that $v =_{E_2} r$, against the assumption that r is non-collapsing in E. It follows that $[\![r]\!]^{\mathcal{A}}_{\alpha} \in Z_{2,2} = Y_2 \setminus X'_2$.

Now let $i = 1$ and so let $r \in G_1 \setminus V$ be non-collapsing in E. Again, first we show that $[\![r]\!]^{\mathcal{A}}_{\alpha} \in Y_2$. Let $\beta_1 := h \circ \alpha_1$, as in Lemma 4.13. Since α is an injective valuation of $\mathcal{V}ar(r)$ onto X'_1, β_1 is a bijective valuation of V onto X'_1, there is a term r' obtained by a bijective renaming of the variables in r such that $[\![r]\!]^{\mathcal{A}}_{\alpha} = [\![r']\!]^{\mathcal{A}^{\Sigma_1}}_{\beta_1}$. Again, $r' \in G_1$ as G_1 is closed under renaming, and thus $[\![r']\!]^{\mathcal{A}^{\Sigma_1}}_{\beta_1} \in Y_2$ by Lemma 4.13. As in the previous case, using the fact that $\mathcal{A}^{\Sigma_1}$ is free in E_1 over X'_1, we can prove that $[\![r]\!]^{\mathcal{A}}_{\alpha} \notin X'_1$. It follows that $[\![r]\!]^{\mathcal{A}}_{\alpha} \in Z_{2,1} = Y_2 \setminus X'_1$. ∎

LEMMA 4.17. *For $i = 1, 2$, let $t_i \in T(\Sigma_i, V)$ and let α be an injective valuation of $\mathcal{V}ar(t_1) \cup \mathcal{V}ar(t_2)$ into $Y_2 = X'_1 \cup X'_2$ such that $\alpha(v) \in X'_i$ for all $v \in \mathcal{V}ar(t_i)$. If $[\![t_1]\!]^{\mathcal{A}}_{\alpha} = [\![t_2]\!]^{\mathcal{A}}_{\alpha}$ then $t_1 =_E t_2$.*

Proof. Let $s_i(\bar{r}_i) := NF_i(t_i)$ for $i = 1, 2$ and assume without loss of generality that α is defined on all the variables of $s_i(\bar{r}_i)$ and maps them into X'_i.[21] From the assumptions and the equivalence in E of $s_i(\bar{r}_i)$ with t_i it follows that

$$[\![s_1(\bar{r}_1)]\!]^{\mathcal{A}}_{\alpha} \quad = \quad [\![s_2(\bar{r}_2)]\!]^{\mathcal{A}}_{\alpha} \,. \tag{5}$$

Since every non-variable element r of $\bar{r}_2$ is a non-collapsing term of G_2 by Assumption 4.1, and α is an injection of $\mathcal{V}ar(r)$ into X'_2, we have by Lemma 4.16 that $[\![r]\!]^{\mathcal{A}}_{\alpha} \in Z_{2,2} \subseteq X'_1$.

Now, we modify $s_2(\bar{r}_2)$ as follows: every non-variable component r of the tuple $\bar{r}_2$ is replaced by a variable. To be more precise, let $a := [\![r]\!]^{\mathcal{A}}_{\alpha}$. We replace r by the variable v_a, where v_a is a fresh variable if a is not in the image of α, and v_a is the variable v satisfying $\alpha(v) = a$ otherwise. Let $s(\bar{v})$ be the Σ-term obtained this way. We extend α to an injection β by defining $\beta(v_a) := a$ for all the fresh variables v_a. By construction, we have $[\![s_1(\bar{r}_1)]\!]^{\mathcal{A}}_{\beta} = [\![s_1(\bar{r}_1)]\!]^{\mathcal{A}}_{\alpha} = [\![s_2(\bar{r}_2)]\!]^{\mathcal{A}}_{\alpha} = [\![s(\bar{v})]\!]^{\mathcal{A}}_{\beta}$, and thus $\mathcal{A}^{\Sigma_1}, \beta \models s_1(r_1) \equiv s(\bar{v})$.

Recalling that $\mathcal{A}^{\Sigma_1}$ is free in E_1 over X'_1, we can conclude by Proposition 2.1 that $s_1(\bar{r}_1) =_{E_1} s(\bar{v})$. By Assumption 4.1 this entails that all the elements of $\bar{r}_1$ are variables.

In a completely symmetric way we can prove that all the elements of $\bar{r}_1$ are variables as well. From equation (5) above then we have that $\mathcal{A}^{\Sigma}, \alpha \models s_1 \equiv s_2$ with α injecting $\mathcal{V}ar(s_1 \equiv s_2)$ into Y_2. Since $\mathcal{A}^{\Sigma}$ is free in E^{Σ} over Y_2, this entails that $s_1 =_E s_2$. Given that each t_i is equivalent to $s_i = NF_i(t_i)$ in E_i, and so in E, we obtain that $t_1 =_E t_2$, as claimed. ∎

LEMMA 4.18 (Interpolation Lemma). *For $i = 1, 2$ let $t_i \in T(\Sigma_i, V)$ such that $t_1 =_E t_2$. Then, there is a term $s \in T(\Sigma, V)$ such that*

$$t_1 =_{E_1} s \quad \textit{and} \quad s =_{E_2} t_2.$$

Proof. Let α be a valuation of $\mathcal{V}ar(t_1) \cup \mathcal{V}ar(t_2)$ as in Lemma 4.17. Notice that such a valuation can alway be constructed, for instance, by injecting $\mathcal{V}ar(t_1) \cup \mathcal{V}ar(t_2)$ into the (infinite) set $Z_2 = X'_1 \cap X'_2$. From $t_1 =_E t_2$ and the fact that $\mathcal{A}$ is a model of E we have that $[\![t_1]\!]^{\mathcal{A}}_{\alpha} = [\![t_2]\!]^{\mathcal{A}}_{\alpha}$. Exactly as in the proof of Lemma 4.17 then, we can show that there is a Σ-term s such that $t_1 =_{E_1} s$. The equivalence $s =_{E_2} t_2$ then follows from the fact that $s =_E t_1 =_E t_2$ and Proposition 4.14. ∎

[21] Otherwise, we extend α so that it maps the extra variables of $s_i(\bar{r}_i)$ to new distinct elements of the infinite set $Z_2 = X'_1 \cap X'_2$.

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**, **Ident1**, **Ident2**, **Simpl**, **Shar1**, **Shar2** to S until none of them is applicable.
3. Succeed if S has the form $\{v \not\equiv v\} \cup T$ and fail otherwise.

FIG. 4 The Extended Combination Procedure.

The interpolation lemma above already provides a partial result on the decidability of the word problem in the combined theory E.

PROPOSITION 4.19. *Let t_1, t_2 be two pure terms, i.e., $t_1, t_2 \in T(\Sigma_1, V) \cup T(\Sigma_2, V)$. Then, the equivalence of t_1 and t_2 in E is decidable.*

Proof. By Proposition 4.14 the claim is trivial if t_1, t_2 are both Σ_1- or both Σ_2-terms. Therefore assume that for $i = 1, 2$, $t_i \in T(\Sigma_i, V)$, say.

By Lemma 4.18, t_1 and t_2 are equivalent in E iff they are equivalent in their respective theories to a same Σ-term. By Assumption 4.1, their normal form is itself a Σ-term whenever they are equivalent to a Σ-term. This entails that the problem of proving that $t_1 =_E t_2$ can be reduced to the problem of verifying that $NF_1(t_1)$, say, is a Σ-term and then proving that $NF_1(t_1) =_{E_2} t_2$. The claim then follows from the assumption that NF_1 is computable and the word problem in E_2 is decidable. ∎

In the next section, we lift this result to arbitrary terms in $T(\Sigma_1 \cup \Sigma_2, V)$ by using an extension of the combination procedure in Section 3.

5. AN EXTENDED COMBINATION PROCEDURE

In the following, we show that the combination procedure introduced in Section 3 can be extended to solve the word problem for unions of theories sharing constructors. More precisely, we will consider an equational theory $E := E_1 \cup E_2$ where, for $i = 1, 2$,

- $\Sigma := \Sigma_1 \cap \Sigma_2$ is a set of constructors for E_i;
- ${E_1}^{\Sigma} = {E_2}^{\Sigma}$;
- E_i admits a recursive Σ-base G_i closed under bijective renaming of V;
- G_i-normal forms are computable for Σ and E_i by a function NF_i that satisfies Assumption 4.1.
- the word problem for E_i is decidable.

In Section 4, 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_i. Considering that G_i contains V, we will now use a more descriptive notation. We will distinguish the variables in $\bar{q}$ from the non-variable terms and write $s(\bar{y}, \bar{r})$

$$\textbf{Ident2}\ \frac{T \quad u \not\equiv v \quad u \equiv s \quad v \equiv t}{v \not\equiv v}$$

if $s \in T(\Sigma_i, V)$ and $t \in T(\Sigma_j, V)$ with $\{i, j\} = \{1, 2\}$,
and $s =_E t$.

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

if (a) $x \in \mathcal{V}ar(T)$,
(b) $t \in T(\Sigma_i, V) \setminus G_i$ for $i = 1$ or $i = 2$,
(c) $NF_i(t) = s(\bar{y}, \bar{r}) \in T(\Sigma, G_i) \setminus V$,
(d) $\bar{r}$ nonempty and $\bar{r} \subseteq G_i \setminus T(\Sigma, V)$,
(e) $\bar{z}$ fresh variables with no repetitions,
(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 $(y \equiv r) \in T$.

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

if (a) $x \in \mathcal{V}ar(T)$,
(b) $t \in T(\Sigma_i, V) \setminus G_i$ for $i = 1$ or $i = 2$,
(c) $NF_i(t) = s \in T(\Sigma, V) \setminus V$,
(d) $\bar{y}_1 \subseteq \mathcal{V}ar(s)$,
(e) $(x \equiv s) \prec (y \equiv r)$ for no $(y \equiv r) \in T$.

FIG. 5 The New Transformation Rules.

instead, where $\bar{y}$ collects the elements of $\bar{q}$ that are in V and $\bar{r}$ those that are in $G_i \setminus V$.

The extended combination procedure is described in Figure 4. Its only difference with the previous one is the presence of three new derivation rules, **Ident2**, **Shar1** and **Shar2**, which apply when Σ_1 and Σ_2 are not disjoint, i.e., when the shared signature Σ is nonempty. The new rules are used to propagate the constraint information represented by shared terms.

The goal of **Ident2** is to identify the variables in the system's disequation whenever they are equated to terms that have different signature but are both equivalent to the same shared term.[22] By Lemma 4.18 this occurs exactly when the two terms are equivalent in E, a condition that, as explained in Proposition 4.19, is decidable because it reduces (thanks to Assumption 4.1) to checking that $NF_i(s) =_{E_j} t$.

The goal of both **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 right-hand side t of an equation $x \equiv t$ by its normal form, and then plug the "shared part" of the normal form into all equations whose right-hand sides contain x. The exact formulation of the rules is somewhat more complex since we must ensure that the rules do not apply repeatedly to the same equation and the resulting system is again an abstraction system. In particular, the rules must preserve the "alternating signature" requirement in Condition 3b of Definition 3.2.

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_i. 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 $j \in \{1, \ldots, n\}$.

We make one assumption on **Shar1** and **Shar2** that is not explicitly listed in their preconditions.

Assumption 5.1. We assume that NF_i $(i = 1, 2)$ is such that, whenever the set $V_0 := \mathcal{V}ar(NF_i(t)) \setminus \mathcal{V}ar(t)$ is non-empty,[23] each variable in V_0 is fresh with respect to the current set S.

Such an assumption can be made without loss of generality. In fact, since each G_i is closed under bijective variable renaming, applying any such renaming to $NF_i(t)$ yields a term still in $T(\Sigma, G_i)$. In particular, we can choose a renaming that fixes the variables in $\mathcal{V}ar(t)$ and moves those in V_0 to fresh variables. This process is clearly effective and yields a term also equivalent to t in E_i.

In both **Shar** rules it is required that the normal form of t be a non-variable term—a consequence of Condition (c) in both rules. The reason for this restriction is that the rules **Coll1** and **Coll2** already take care of the case in which a Σ_i-term is equivalent in E_i to a variable. Notice that **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

[22]Strictly speaking then, **Ident2** can apply even if Σ_1 and Σ_2 are disjoint provided that the terms t_1 and t_2 in its premise are equivalent to the same variable. But in that case, its effect can be also achieved by **Coll1** and **Coll2**.

[23]This might happen because Definition 4.10 and Assumption 4.1 do not entail that all the variables of $NF_i(t)$ occur in t.

every $\prec$-chain is made of equations with alternating signatures (cf. Condition 3b of Definition 3.2). 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 $x \equiv t$.[24] **Shar2** restores this property by merging into $x \equiv s$ all of its immediate successors—which are collected, if any, in the set $\bar{y}_1 \equiv \bar{r}_1$ thanks to Condition (e) in the rule. The replacement of $\bar{y}_1$ by $\bar{r}_1$ in **Shar1** is done for similar reasons.

In both **Shar** rules the condition $x \in \mathcal{V}ar(T)$ is necessary to ensure termination.

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 terminates on all inputs and is sound and complete.

5.1. The Correctness Proof

In this subsection, we will consider a countable family $\mathcal{S} := \{S_j \mid j \geq 0\}$ such that S_0 is an abstraction system and for all $j > 0$, S_j is either identical to S_{j-1} or is derived from S_{j-1} by an application of **Coll1**, **Coll2**, **Simpl**, **Ident1**, **Ident2**, **Shar1**, or **Shar2**. In particular, $\mathcal{S}$ may correspond to the family generated by one execution of the combination procedure, defined in the same way as in Subsection 3.3. In general, however, the first element of $\mathcal{S}$ may be an arbitrary abstraction system, not necessarily one produced by the purification procedure described in Section 3.1. As before, we will denote by $\prec_j$ the restriction of $\prec$ to S_j.

We start by showing that all the elements of $\mathcal{S}$ are in fact abstraction systems.

LEMMA 5.1. *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 assumption. The induction step is proved exactly as in Lemma 3.6 for the cases in which S_j is derived from S_{j-1} by an application of **Coll1**, **Coll2**, **Simpl**, or **Ident1**. Since the **Ident2** case is trivial, we show below that S_j is an abstraction system also when it is derived by **Shar1** or **Shar2**.

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

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

To see that S_j satisfies Condition 1 of Definition 3.2, first notice that $s(\bar{y}, \bar{r})$ is not a variable by precondition (c) 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 with $\iota \in \{1, 2\} \setminus \{i\}$. 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

- every right-hand side $t'[x]$ of T, which is a term in $T(\Sigma_\iota, V) \setminus T(\Sigma, V)$ by induction hypothesis (cf. observation after Definition 3.2), is replaced by the term $t'[x/s(\bar{y}, \bar{z})[\bar{y}_1/\bar{r}_1]]$, which is also in $T(\Sigma_\iota, V) \setminus T(\Sigma, V)$ by the above;

[24] As explained above, we assume that the variables in $\mathcal{V}ar(s) \setminus \mathcal{V}ar(t)$ do not occur in the abstraction system. Thus, the equations in $\bar{y}_1 \equiv \bar{r}_1$ are in fact successors of $x \equiv t$.

- the elements of $\bar{r}$ are not Σ-terms, have 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};
- all the immediate successors of $x \equiv s(\bar{y}, \bar{r})$ are inherited from $x \equiv t$ because, thanks to 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} (and without loss of generality also not in $\bar{z}$);
- $s(\bar{y}, \bar{r})$ is not a Σ-term because the tuple $\bar{r}$ is non-empty and made of non-Σ-terms;
- 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} induces 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 one of these nodes 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:

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

We can show that S_j satisfies Conditions 1, 2, 3a, and 3b of Definition 3.2 essentially in the same way as in the **Shar1** case. For Condition 3a, additionally observe that we cannot use $x \equiv s$ in S_j because s is a shared term. By using $x \equiv s[\bar{y}_1/\bar{r}_1]$ instead, where the terms of $\bar{r}_1$ are non-shared by induction, we make sure that any successors of this equation is a successor of an equation in $\bar{y}_1 \equiv \bar{r}_1$. Since every equation in $\bar{y}_1 \equiv \bar{r}_1$ is a successor of $x \equiv t$ in S_{j-1},[25] and S_{j-1} satisfies Condition 3a by induction, all the equations in $\bar{y}_1 \equiv \bar{r}_1$ have the same signature, which is also the signature of $x \equiv s[\bar{y}_1/\bar{r}_1]$. Thus, Condition 3a for $x \equiv s[\bar{y}_1/\bar{r}_1]$ and its successors in S_j is satisfied since it is satisfied for the equations in $\bar{y}_1 \equiv \bar{r}_1$ and their successors in S_{j-1}. If the tuple $\bar{y}_1$ is empty, then $s[\bar{y}_1/\bar{r}_1] = s$ is a shared term, but this is not a problem since in this case the equation $x \equiv s$ does not have any predecessors or successors in S_j. ∎

Termination

The extended combination procedure as well halts on all inputs, but to prove it we will need a more sophisticated argument that uses an appropriate well-founded ordering[26] on abstraction systems, defined in the following.

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

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

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 [8] for more details).

DEFINITION 5.2 ($\sqsupset$). For all $M, N \in \mathcal{M}(P)$, $M \sqsupset N$ iff there exist $X, Y \in \mathcal{M}(P)$ 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 $x >_l y$.

It is possible to show that $\sqsupset$ is a well-founded total ordering on $\mathcal{M}(P)$ [8]. 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 $\sqsupset$.

In Section 3, we saw that the equations of an abstraction system can be considered as the nodes of a graph whose edges are induced by the relation $\prec$. In what follows we will use a notion of *reducibility* for such nodes.

DEFINITION 5.3 (Node Reducibility). Let $(T, \prec)$ be the dag induced by an abstraction system $\{x \neq y\} \cup T$ and let $e \in T$. We say that e is *irreducible*, or that its *reducibility* is 0, and write $\mathsf{r}(e) = 0$, if the right-hand side of e is a member of G_1 or G_2 (the Σ-bases of E_1, E_2, respectively). We say that e is *reducible*, or that its *reducibility* is 1, and write $\mathsf{r}(e) = 1$, otherwise.

Now, for all $j \geq 0$ let h_j and r_j be the height (cf. Definition 3.4) and the reducibility function on the nodes of the dag induced by the abstraction system S_j. These functions can be used to associate a finite multiset to S_j: the multiset M_j consisting of the pairs $(\mathsf{h}_j(e), \mathsf{r}_j(e))$ for every equation e 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 $(n, 1)$.

Our interest in the multiset ordering $\sqsupset$ is motivated by the fact that each application of a derivation rule in the procedure reduces, with respect to $\sqsupset$, the multiset associated with the current abstraction system. To show that, we will appeal to the following easily provable properties of the height functions h_j.

LEMMA 5.4. *The following holds for every finite dag $\mathcal{G}$ and associated height function* h.

1. *For all nodes a, b of $\mathcal{G}$, 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 $\mathcal{G}$ to another of greater height does not change the height of any node of $\mathcal{G}$.*
3. *Removing an edge in $\mathcal{G}$ does not increase the height of any node of $\mathcal{G}$ (although it may decrease the height of some).*

4. Removing a node and relative edges from $\mathcal{G}$ does not increase the height of the remaining nodes (although it may decrease the height of some).

LEMMA 5.5. *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**, **Coll2**, **Simpl**, **Ident1**, **Ident2**, **Shar1**, *or* **Shar2**.

Proof. We consider only the application of **Coll1**, **Ident1**, **Shar1**, and **Shar2**. The proof for **Coll2** is very similar to that for **Coll1**, and the proof for **Ident2** and **Simpl** is trivial.

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

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

As in the proof of Lemma 5.1 we can easily show that S is an abstraction system as well. Then, where M is the multiset associated to S, we show that $M_j \sqsupseteq M \sqsupset M_{j+1}$.

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

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

we know from Point 1 of Lemma 5.4 that every $v \equiv t$ in S such that $(v_2 \equiv s_2) \prec (v \equiv t)$ has a greater height in S_j 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_j to S, the only new edges are from a node of S_j to one that is already higher. By Point 2 of Lemma 5.4 then no node in S_j 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_j to M. This, however, is not a problem because at least one greater pair, $(n+1, \mathsf{r}_j(v_1 \equiv s_1))$, is replaced by a smaller one. 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, whereas it was $n + 1 > 0$ before. By definition of $\sqsupset$, we can conclude that $S_j \sqsupseteq M$.

($M \sqsupset M_{j+1}$) As S_{j+1} is obtained from S by removing the node $v_1 \equiv s_1$, we can use Point 4 of Lemma 5.4 to conclude 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, we have that $M \sqsupset M_{j+1}$.

Ident1. We have that $S_j = T \cup \{x \equiv s, y \equiv t\}$ 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, Point 2 of Lemma 5.4 shows that this does not change the height of any node. Then, remove the node $x \equiv s$. By Point 4 of Lemma 5.4, 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' height 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:

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

Observe that there may be more nodes in S_{j+1} than in S_j: those corresponding to the equations in $\bar{z} \equiv \bar{r}$. Let n be the height of $x \equiv t$ in S_j, which is at least 1 as x occurs in T by assumption. 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 $\bar{z}$ and does not occur in the tuple $\bar{y}_1$, we then obtain that

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

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 $s(\bar{y},\bar{z})[\bar{y}_1/\bar{r}_1]$ in T may introduce new edges in S_{j+1} between old nodes,[27] 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 5.4 then some old nodes may move to a smaller height in S_{j+1} but none will move to a greater height after 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 other hand, all the nodes at height $> n$ keep the same height, to prove that $M_j \sqsupset M_{j+1}$ we argue that the number of reducible nodes at height n decreases. 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 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 in G_i. Third, the node $x \equiv t$ of S_j, which by assumption had height $n > 0$ and was

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

reducible, is replaced by the node $x \equiv s(\bar{y}, \bar{r})$ whose height in S_{j+1} is 0, because x occurs in no right-hand side of S_{j+1}.

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

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

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 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. But this is an immediate consequence of the fact that the node $x \equiv t$ in S_j, which by assumption had height $n > 0$ and was reducible, is replaced by the node $x \equiv s[\bar{y}_1/\bar{r}_1]$ whose height in S_{j+1} is 0. ∎

PROPOSITION 5.6 (Termination). *The combination procedure halts on all inputs.*

Proof. By Lemma 5.5 and the well-foundedness of $\sqsupset$ we are guaranteed that the procedure applies the various rules only finitely many times. As in the proof of Proposition 3.7 then, all we need to show is that the preconditions of each rule can be tested in finite time. We already know this to be true for the rules in Figure 2. Thanks to Proposition 4.19, it also true for **Ident2**.

For **Shar1**, it should be clear that the test on the preconditions (a), (e) and (f) is effective. The test on conditions (b) and (d) is effective because G_i is recursive by assumption. The computation of the normal form of t in (c) is effective because G_i-normal forms are computable for $i = 1, 2$ by assumption; its decompositions into the terms $s, \bar{r}$ is effective by Proposition 4.11 because G_i is recursive. A similar argument applies to the preconditions of **Shar2**. ∎

Soundness

The next two lemmas show that the derivation rules preserve satisfiability.

LEMMA 5.7. *Let $\bar{v}_{j-1}$ be a sequence consisting of the left-hand side variables of S_{j-1} and $\bar{v}_j$ be a sequence consisting of the left-hand side variables of S_j. Then, $\exists \bar{v}_{j-1}.S_{j-1} \leftrightarrow \exists \bar{v}_j.S_j$ is valid in E.*

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**, **Ident1**, **Simpl** are proved exactly as in Lemma 3.9. Below we give a proof of the **Ident2** and the **Shar1** case. The proof for **Shar2** is almost identical to that for **Shar1**.

Ident2. We know that S_{j-1} and S_j have the form

$$\begin{array}{lllllllll} S_{j-1} & = & T & \cup & \{u \not\equiv v\} & \cup & \{u \equiv s\} & \cup & \{v \equiv t\} \\ S_j & = & & & \{v \not\equiv v\} & & & & \end{array}$$

It is then enough to show that S_{j-1} is unsatifiable in every model of E. But this is immediate, given that s and t are equivalent in E.

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

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

Let $\mathcal{A}$ be any model of E. First, assume that some valuation α of V satisfies S_j in $\mathcal{A}$. Since S_j contains the equation $x \equiv s(\bar{y}, \bar{r})$ and $t =_E s(\bar{y}, \bar{r})$, we know that $\alpha(x) = [\![t]\!]^{\mathcal{A}}_{\alpha}$. In addition, since S_j also contains the equations $\bar{y}_1 \equiv \bar{r}_1$ and $\bar{z} \equiv \bar{r}$, we also know that $\alpha(x) = [\![s(\bar{y}, \bar{z})[\bar{y}_1/\bar{r}_1]]\!]^{\mathcal{A}}_{\alpha}$. Obviously, this implies that α satisfies S_{j-1} in $\mathcal{A}$.

Conversely, assume that some valuation α satisfies S_{j-1} in $\mathcal{A}$. Let α' be a valuation coinciding with α on all variables except those in $\bar{z}$. For each component $z_i \equiv r_i$ of $\bar{z} \equiv \bar{r}$ we define $\alpha'(z_i) := [\![r_i]\!]^{\mathcal{A}}_{\alpha}$. As above, it is easy to show that $\alpha'(x) = \alpha(x) = [\![s(\bar{y}, \bar{r})]\!]^{\mathcal{A}}_{\alpha'}$ and $\alpha'(x) = [\![s(\bar{y}, \bar{z})[\bar{y}_1/\bar{r}_1]]\!]^{\mathcal{A}}_{\alpha'}$. This implies that α' satisfies S_j in $\mathcal{A}$. Since the variables in $\bar{z}$ are left-hand side variables of S_j, which do not occur in S_{j-1}, the valuations α and α' coincide on the free variables of $\exists \bar{v}_{j-1}.S_{j-1} \leftrightarrow \exists \bar{v}_j.S_j$. ∎

Again, we immediately have the following weaker lemma.

LEMMA 5.8. *For all $j > 0$, the abstraction system S_j is satisfiable in E iff S_{j-1} is satisfiable in E.*

Exactly as we did in Section 3.3 we can now prove that the extended combination procedure is sound.

PROPOSITION 5.9 (Soundness). *If the combination procedure succeeds on an input (s_0, t_0), then $s_0 =_E t_0$.*

Completeness

To show completeness we will prove that, if the combination procedure fails on input (s_0, t_0), then $s_0 \neq_E t_0$. The following lemma provides important information on the structure of the final abstraction system obtained by a failed run of the procedure.

LEMMA 5.10. *Let S_n be the final abstraction system S_n generated by a failed execution of the combination procedure and h_n the height function defined over the dag induced by S_n. Then, S_n can be partitioned into the sets*

$$D := \{x_1 \not\equiv x_2\} \qquad T_1 := \{v^1_j \equiv r^1_j\}_{j \in J_1}$$

$$T := \{v \equiv t \in S_n \mid \mathsf{h}_n(v \equiv t) = 0\} \qquad T_2 := \{v^2_j \equiv r^2_j\}_{j \in J_2}$$

where

1. *x_1 and x_2 are distinct, and J_1 and J_2 are finite;*
2. *v occurs exactly once in $S_n \setminus D$ for every $v \equiv t \in T$;*
3. *v^i_j occurs exactly once as a left-hand side of S_n for every $i \in \{1, 2\}$ and $j \in J_i$, and the height of the corresponding equation is non-zero;*
4. *$r^i_j \in G_i$ for every $i \in \{1, 2\}$ and $j \in J_i$.*

Proof. To start with, for $i = 1, 2$, let T_i be the set of all the Σ_i-equations of S_n that are not in T. As S_n is an abstraction system, it is immediate that D, T, T_1 and T_2 form a partition of S_n. Now, Point 1 is trivial because the procedure has failed and S_n is finite. Point 2 and Point 3 are again an immediate consequence of the fact that S_n is an abstraction system.

To prove Point 4, let $i = 1$, $j \in J_1$, and consider the equation $v_j^1 \equiv r_j^1$ of T_1 (the case for $i = 2$ is analogous). First notice that the variable v_j^1 must occur in the right-hand side of a term in S_n, or else the height of $v_j^1 \equiv r_j^1$ in S_n would be 0, making the equation a member of T instead. Then assume by contradiction that r_j^1 is not an element of G_1. But then, it is not difficult to see that one of **Coll1**, **Coll2**, **Shar1**, **Shar2** applies to $v_j^1 \equiv r_j^1$, against the assumption that S_n is the final abstraction system. ∎

LEMMA 5.11. *The final abstraction system S_n generated by a failed execution of the combination procedure is satisfiable in E.*

Proof. We prove the claim by constructing a valuation α that satisfies S_n in the model $\mathcal{A}$ of E introduced in Lemma 4.13. Consider the sets

$$D := \{x_1 \not\equiv x_2\} \qquad\qquad T_1 := \{v_j^1 \equiv r_j^1\}_{j \in J_1}$$

$$T := \{v \equiv t \in S_n \mid \mathsf{h}_n(v \equiv t) = 0\} \qquad T_2 := \{v_j^2 \equiv r_j^2\}_{j \in J_2}$$

from Lemma 5.10 partitioning S_n. Let U be a set made of all the elements of $\mathcal{V}ar(S_n \setminus D)$ that are not a left-hand side variable of S_n, and let $V_i := \{v_j^i\}_{j \in J_i}$ for $i = 1, 2$. Observe that $U \cup V_1 \cup V_2 \subseteq \mathcal{V}ar(T \cup T_1 \cup T_2)$ and that for each $v \equiv t \in S_n$, all the variables of t are in $U \cup V_1 \cup V_2$.[28]

Now, where α_0 is an arbitrary injective valuation of U into Z_2 (cf. Figure 3), we define α over $\mathcal{V}ar(T \cup T_1 \cup T_2)$ as follows:

$$\alpha(v) := \begin{cases} \alpha_0(v) & \text{if } v \in U \\ [\![t]\!]_\alpha^{\mathcal{A}} & \text{if } v \equiv t \in S_n \end{cases}$$

Because of its recursive definition we first need to prove that α is well-defined. We will do this by induction on the "inverse height" of equations in S_n. Where M is the maximum of the heights of all nodes in $T \cup T_1 \cup T_2$, let κ be the function from $\mathcal{V}ar(T \cup T_1 \cup T_2)$ into the non-negative integers defined as follows:

$$\kappa(v) := \begin{cases} 0 & \text{if } v \in U \\ (M+1) - \mathsf{h}_n(v \equiv t) & \text{if } v \equiv t \in S_n \end{cases}$$

Note that the only variables v with $\kappa(v) = 0$ are the elements of U. In addition, if $v \equiv t$ is an equation of S_n, then $\kappa(v) > 0$ and $\kappa(v) > \kappa(u)$ for all variables u occurring in t.

The well-definedness of α can now be easily proved by induction on κ. If $\kappa(v) = 0$, then $v \in U$ and α is obviously well-defined on U. If $\kappa(v) > 0$, then v is the left-hand side of some equation $v \equiv t$ of S_n. By induction hypothesis, α is well-defined on every variable u occurring in t because $\kappa(v) > \kappa(u)$ as mentioned above. Consequently, $\alpha(v) = [\![t]\!]_\alpha^{\mathcal{A}}$ is also well-defined.

Next, we show that the restriction of α to $U \cup V_1 \cup V_2$ is an injective extension of α_0 such that $[\![v_j^i]\!]_\alpha^{\mathcal{A}} \in Z_{2,i}$ for all $i \in \{1, 2\}$ and $j \in J_i$. This is again done by induction on κ.

Consider a variables v_j^i in $V_1 \cup V_2$ and the corresponding equation $v_j^i \equiv r_j^i$. Since S_n is an abstraction system, we know that each variable v of r_j^i is in $U \cup V_k$ with

[28]The only variables in $\mathcal{V}ar(T \cup T_1 \cup T_2)$ not contained in $U \cup V_1 \cup V_2$ are the left-hand side variables of equations in T.

$k \neq i$, and that $\kappa(v) < \kappa(v_j^i)$. We can conclude by induction hypothesis that α is an injection of $\mathcal{V}ar(r_j^i)$ into $Z_2 \cup Z_{2,k} = X_i'$.

To see that $[\![v_j^i]\!]_\alpha^\mathcal{A} \in Z_{2,i}$, simply observe that r_j^i is non-collapsing in E, since otherwise it would be collapsing in E_i by Proposition 4.14. But then, either **Coll1** or **Coll2** would apply to $v_j^i \equiv r_j^i$, against the fact that S_n is the final abstraction system. The claim then holds directly by Lemma 4.16 since r_j^i is in G_i, as seen in Point 4 of Lemma 5.10.

To see that α is injective over $U \cup V_1 \cup V_2$ it suffices to show by induction that $\alpha(v_j^i) \neq \alpha(v)$ for every variable v of $U \cup V_1 \cup V_2$ other than v_j^i such that $\kappa(v) \leq \kappa(v_j^i)$ Let v be any such variable.

If $v \in U$, then $\alpha(v_j^i) \neq \alpha(v)$ because $\alpha(v_j^i) \in Z_{2,i}$ as seen above, $\alpha(v) \in Z_2$ by definition of α_0, and $Z_{2,i} \cap Z_2 = \emptyset$. Similarly, if v is in V_k with $k \neq i$, then $\alpha(v_j^i) \neq \alpha(v)$ because $\alpha(v) \in Z_{2,k}$ and $Z_{2,i} \cap Z_{2,k} = \emptyset$. Finally, if v is in V_i, i.e. $v = v_\ell^i$ for some $\ell \in J_i$, assume by contradiction that $\alpha(v_j^i) = \alpha(v)$. Then, we have that $\mathcal{A}^{\Sigma_i}, \alpha \models r_j^i \equiv r_\ell^i$. Now, each variable u of $r_j^i \equiv r_\ell^i$ belongs to $U \cup V_k$, and $\kappa(u) < \kappa(v_j^i)$ if u occurs in r_j^i and $\kappa(u) < \kappa(v) \leq \kappa(v_j^i)$ if u occurs in r_ℓ^i. Thus, by the induction hypothesis, the variables of $r_j^i \equiv r_\ell^i$ are mapped by α to distinct values of $Z_2 \cup Z_{2,k}$. Since $\mathcal{A}^{\Sigma_i}$ is free in E_i over $X_i' = Z_2 \cup Z_{2,k}$, we obtain by Proposition 2.1 that $r_j^i =_{E_i} r_\ell^i$. But this is impossible because otherwise the rule **Ident1** would apply to $v_j^i \equiv r_j^i$ and $v_\ell^i \equiv r_\ell^i$.

In conclusion, we have shown that α is a well-defined valuation of $\mathcal{V}ar(T \cup T_1 \cup T_2)$ into $\mathcal{A}$ which, in addition, is injective over $U \cup V_1 \cup V_2$ and maps each variable of $U \cup V_k$ into $Z_2 \cup Z_{2,k} = X_i'$ for all $i, k \in \{1,2\}, i \neq k$. By construction, α satisfies $T \cup T_1 \cup T_2$ in $\mathcal{A}$. We show below that it satisfies, or can be extended to satisfy, the disequation $\{x_1 \not\equiv x_2\}$ as well, which will prove the claim that $S_n := \{x_1 \not\equiv x_2\} \cup T \cup T_1 \cup T_2$ is satisfiable in E.

Clearly, if α is undefined for x_1 or x_2 or both,[29] since $\mathcal{A}$ has an infinite carrier, α can be trivially extended so that it satisfies $x_1 \not\equiv x_2$. Therefore, assume that α is defined for both x_1 and x_2. We distinguish four cases, depending on where x_1, x_2 occur in S_n.

(a) Neither x_1 nor x_2 is a left-hand side variable of S_n. Then, they must both be (distinct) elements of U. In that case, $x_1 \not\equiv x_2$ is immediately satisfied by α because α is injective over U.

(b) x_1 is a left-hand side variable of S_n while x_2 is not. Then, x_1 must occur in an equation of the form $x_1 \equiv t_1$ and x_2 must be in U. Let $i, k \in \{1,2\}$ with $i \neq k$ and assume that t_1 is a Σ_i-term. Now assume by contradiction that $\alpha(x_1) = \alpha(x_2)$, which means that $\mathcal{A}^{\Sigma_i}, \alpha \models x_2 \equiv t_1$. Because of the alternating signature property of S_n, all the variables of t_1 are in $U \cup V_k$. From the above then we know that α maps $\mathcal{V}ar(x_2 \equiv t_1)$ to distinct elements of X_i'. Since $\mathcal{A}^{\Sigma_i}$ is free in E_i over X_i', we can conclude that $x_2 =_{E_i} t_1$. But again this is impossible because then either **Coll1** or **Coll2** applies to $x_1 \equiv t_1$, contradicting the assumption that S_n is the final abstraction system.

(c) x_2 is a left hand side variable of S_n while x_1 is not. Symmetrical to the previous case.

(d) Both x_1 and x_2 are a left-hand side variable of S_n. Then, x_j must occur in an equation of the form $x_j \equiv t_j$ for $j = 1, 2$. Again, assume by contradiction that $\alpha(x_1) = \alpha(x_2)$, which means that $[\![t_1]\!]_\alpha^\mathcal{A} = [\![t_2]\!]_\alpha^\mathcal{A}$. If t_1 and t_2 have the same

[29] The variables x_1, x_2 need not occur in $\mathcal{V}ar(T \cup T_1 \cup T_2)$.

signature Σ_i for some $i \in \{1,2\}$, we can argue as in case (b) that $t_1 =_{E_i} t_2$, which is impossible because then **Ident1** applies to $x_1 \equiv t_1$ and $x_2 \equiv t_2$. If t_1 and t_2 do not have the same signature, we can use Lemma 4.17 to show that $t_1 =_E t_2$. But this is also impossible because then **Ident2** applies to $x_1 \equiv t_1$ and $x_2 \equiv t_2$. ∎

With the above lemma, proving the completeness of the combination procedure is now straightforward.

PROPOSITION 5.12 (Completeness). *The combination procedure succeeds on input (s_0, t_0) if $s_0 =_E t_0$.*

Proof. By Lemma 5.6 the procedure either succeeds or fails; therefore, we can prove the claim by proving that whenever the procedure fails on input (s_0, t_0), the formula $s_0 \not\equiv t_0$ is satisfiable in E. Thus, assume that the procedure fails and let S_n be the abstraction system generated by the last rule application. Given Lemma 5.8 and the construction of S_0, it is enough to show that S_n is satisfiable in E. But this is true by Lemma 5.11. ∎

As an aside, we would like to point out that nowhere in the proof of Proposition 5.12 (and of the lemmas that it uses) 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, we obtain the following modularity result for the decidability of the word problem.

THEOREM 5.13. *Let E_1, E_2 be two non-trivial equational theories of signature Σ_1, Σ_2, respectively, such that $\Sigma := \Sigma_1 \cap \Sigma_2$ is a set of constructors for both E_1 and E_2, and ${E_1}^\Sigma = {E_2}^\Sigma$. Let G_1, G_2 be Σ-bases of E_1, E_2, respectively. If for $i = 1, 2$,*

- *G_i is closed under bijective renaming of V and recursive,*
- *G_i-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.

This result (properly) extends the result for the disjoint-signatures case given in Theorem 3.12. In fact, whenever the set Σ of symbols shared by E_1 and E_2 is empty, it is trivially a set of constructors for both E_1 and E_2. In that case, a Σ-base G_i of E_i is the whole set $T(\Sigma_i, V)$. Clearly, G_i is recursive, closed under renaming and, given that every Σ_i-term is in G_i, admits computable normal forms. Furthermore, ${E_1}^\Sigma$ and ${E_2}^\Sigma$ are the same because they both coincide with the set $\{v \equiv v \mid v \in V\}$.

The decidability result of Theorem 5.13 is actually extensible to the union of any (finite) number of theories, all (pairwise) sharing the same signature Σ and satisfying the same properties as E_1 and E_2 above. The reason is that, remarkably, all the needed properties are modular with respect to theory union, as we show in the next section.

We conclude this section by pointing out that, in contrast with the termination proof for the disjoint case, the termination argument employed in Lemma 5.5 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_i.

6. MODULARITY OF CONSTRUCTORS

In this section, we will see that the property of being a set of constructors is preserved by the union of theories. We will also see that normal forms are computable in a union theory whenever they are computable in its component theories and the word problem is decidable for those theories.

Again, we fix two non-trivial equational theories E_1, E_2 with respective signatures Σ_1, Σ_2 such that, for $i = 1, 2$

- $\Sigma := \Sigma_1 \cap \Sigma_2$ is a set of constructors for E_i;
- ${E_1}^{\Sigma} = {E_2}^{\Sigma}$;
- E_i admits a recursive Σ-base G_i closed under bijective renaming of V;
- G_i-normal forms are computable for Σ and E_i by a function NF_i satisfying Assumption 4.1;
- the word problem for E_i is decidable.

We will show that Σ is a set of constructors for $E := E_1 \cup E_2$ by explicitly constructing a Σ-base G^* of E out of the given Σ-base G_1 and G_2 of E_1 and E_2. In the course of proving that G^* is a Σ-base of E we will also prove that it is recursive, closed under bijective renaming, and such that G^*-normal forms are computable.

DEFINITION 6.1 (G^*). For $i = 1, 2$ let $G_i^* := \bigcup_{n=0}^{\infty} G_i^n$ where $\{G_i^n \mid n \geq 0\}$ is the family of sets defined as follows:

$$
\begin{array}{lcll}
G_i^0 & := & V & \\
G_i^{n+1} & := & G_i^n \cup \{r(r_1, \ldots, r_m) & \mid \; r(v_1, \ldots, v_m) \in G_i \setminus V, \\
 & & & \; r \text{ non-collapsing in } E, \\
 & & & \; r_j \in G_k^n \text{ for all } j = 1, \ldots, m \text{ with } k \neq i, \\
 & & & \; r_j \neq_E r_{j'} \text{ for all distinct } j, j' = 1, \ldots, m\}
\end{array}
$$

The set G^* is the union $G_1^* \cup G_2^*$.

It is easy to see that, for $i = 1, 2$, the set G_i^1 defined above consists of all the variables and the non-variable terms of G_i that are non-collapsing in E_i. Furthermore, for each $r \in G^*$ there is an $i \in \{1, 2\}$ and a smallest $n \geq 0$ such that $r \in G_i^n$. We call n the *number of layers* of r. The reason is that, for $n > 0$ every element of G_i^n has a stratified recursive structure. A term in $G_i^1 \setminus G_i^0$ has just one layer. A term $r(\bar{r})$ in $G_i^n \setminus G_i^{n-1}$ has n layers. Layer 1, the top layer, is made of the term r only; layer 2 is made of all the terms that are at layer 1 in an element of $\bar{r}$; and so on. Furthermore, terms in the same layer all belong to either G_1 or G_2, and if the terms in one layer are in G_i then the non-variable terms in the next layer are not in G_i.

Like each G_i, G^* is clearly closed under bijective variable renaming. We show below that it is recursive as well.

PROPOSITION 6.2. *It is decidable whether a* $(\Sigma_1 \cup \Sigma_2)$*-term is in* G^* *or not.*

Proof. Let $t \in T(\Sigma_1 \cup \Sigma_2, V)$. Recalling that $G^* := G_1^* \cup G_2^*$, we prove the claim by proving by term induction the stronger claim that, for $i = 1, 2$, it is decidable whether t is in G_i^* or not. Let $i, k \in \{1, 2\}$ with $i \neq k$.

Input: Abstraction system S.

1. Repeatedly apply (in any order) **Coll1**, **Coll2**, **Ident1**, **Ident2**, **Simpl**, **Shar1**, **Shar2** to S until none of them is applicable.

2. Succeed if S has the form $\{v \not\equiv v\} \cup T$ and fail otherwise.

FIG. 6 A variant of the combination procedure

(Base case) If t is a variable, the claim is trivial because all variables are in G_i^* by construction.

(Induction Step) If t is not a variable, then we can effectively compute the set of all decompositions of t into a term $r(v_1, \ldots, v_m) \in T(\Sigma_i, V) \setminus V$ and distinct terms $r_1, \ldots, r_m \in T(\Sigma_1 \cup \Sigma_2, V)$ such that $t = r(r_1, \ldots, r_m)$. Note that this set may be empty (if $t(\epsilon) \notin \Sigma_i$) or may be of cardinality greater than 1, but it is clearly always finite. From Definition 6.1 it is easy to see that $t \in G_i^*$ iff there is such a decomposition of t such that

- $r_j \neq_E r_{j'}$ for all distinct $j, j' \in \{1, \ldots, m\}$,
- r is in G_i and is non-collapsing in E, and
- $r_j \in G_k^*$ for all $j = 1, \ldots, m$.

Now, the first condition above is decidable because the word problem for E is decidable by Theorem 5.13; the second condition is decidable because G_i is recursive by assumption, E is non-trivial for being a conservative extension of E_i, and the word problem for E is decidable; the third condition is decidable by induction hypothesis. ∎

We now show that every $(\Sigma_1 \cup \Sigma_2)$-term can be effectively reduced to an E-equivalent term in $T(\Sigma, G^*)$. To do that we will appeal to the correctness of a slight modification of the combination procedure of Section 5. The only significant change in the new procedure, shown in Figure 6, is that its input is an abstraction system instead of a pair of terms. In the same way as in Section 5.1, one can show that the procedure is correct in the following sense:

PROPOSITION 6.3. *The procedure in Figure 6 terminates for all inputs S and succeeds iff S is unsatisfiable in E.*

The following property of the procedure is also an immediate consequence of the results proved in Section 5.1.

LEMMA 6.4. *The final set S_n generated by the procedure on some input S_0 is an abstraction system. Furthermore,*

$$E \models \exists \bar{v}_0 . S_0 \leftrightarrow \exists \bar{v}_n . S_n$$

where $\bar{v}_j$ is a sequence consisting of the left-hand side variables of S_j, for $j \in \{0, n\}$.

We have seen that, from every disequation $s \not\equiv t$ with $s, t \in T(\Sigma_1 \cup \Sigma_2, V)$, it is possible to produce an equivalent abstraction system. Specifically, one can use the purification procedure described in Subsection 3.1 to produce a system S such that

$$E \quad \models \quad (s \not\equiv t) \leftrightarrow \exists \bar{y}.S, \tag{7}$$

where $\bar{y}$ are the left-hand side variables of S. An inverse sort of process is also possible: given an abstraction system S, one can produce a disequation $s \not\equiv t$ such that (7) above holds.

In fact, if $S = \{x \not\equiv y\} \cup T$ is an abstraction system, the relation $\prec$ on T is acyclic. This means that its transitive closure $\prec^+$ is a strict partial ordering on the finite set T, and so it can be extended to a strict total ordering $<$ on T. Let

$$v_1 \equiv t_1 < v_2 \equiv t_2 < \cdots < v_k \equiv t_k$$

be the enumeration of T along this total ordering. We define θ_S to be the substitution obtained by the composition[30]

$$[v_1/t_1][v_2/t_2] \cdots [v_k/t_k].$$

In the following, we will call θ_S the *substitution induced by S*.

LEMMA 6.5. *Let $S = \{x \not\equiv y\} \cup T$ be the abstraction system above and $\bar{v}$ a sequence consisting of the left-hand side variables of S. Then, $E \models (x\theta_S \not\equiv y\theta_S) \leftrightarrow \exists \bar{v}.S$.*

Proof. For having been generated from an abstraction system, θ_S is easily shown to have the form $\theta_S := [v_1/t_1][v_2/t_2] \cdots [v_k/t_k]$ where v_i does not occur in t_j for all $j \geq i$ and $i \in \{1, \ldots, k\}$. The claim then follows from the general fact that

$$E \quad \models \quad \varphi[v/t] \leftrightarrow \exists v.(\varphi \wedge v \equiv t)$$

for every formula φ, term t, and variable v not occurring in t. ∎

It is useful to notice that, for all $v_i \equiv t_i \in S$, the term $v_i\theta_S$ coincides with the term $t_i\theta_S$, which in turn is obtained essentially by "plugging in" into t_i all the terms $v_j\theta_S$ such that $v_j \equiv t_j \in S$ and $v_i \equiv t_i \prec v_j \equiv t_j$.

LEMMA 6.6. *Let S_n be the final abstraction system generated by the procedure in Figure 6 on some input S_0. Let h_n be the height function over S_n and θ_n the substitution induced by S_n. Then, the following holds for all $i = 1, 2$ and $x \equiv r, y \equiv t \in S_n$ such that $r, t \in T(\Sigma_i, V)$:*

1. *if $x \neq y$, then $x\theta_n \neq_E y\theta_n$;*

2. *$x\theta_n$ is non-collapsing in E;*

3. *if $\mathsf{h}_n(x \equiv r) > 0$, then $x\theta_n \in G_i^*$.*

Proof. Let $i \in \{1, 2\}$ and $x \equiv r, y \equiv t \in S_n$ with $r, t \in T(\Sigma_i, V)$.

To prove Point 1, assume that $x \neq y$ and consider the abstraction system $S = \{x \not\equiv y\} \cup T$ obtained from S_n by replacing its disequation by $x \not\equiv y$. Since S's equational part coincides with S_n's, we have that θ_S, the substitution induced

[30] Note that θ_S does not depend on which total extension of $\prec^+$ we take.

by S, coincides with θ_n. It follows that $x\theta_n = x\theta_S$ and $y\theta_n = y\theta_S$. By Lemma 6.5 then, $x\theta_n \not\equiv y\theta_n$ is satisfiable in E iff S is satisfiable in E.

Now observe that no derivation rules apply to S. In fact, the rule **Ident2** does not apply to $x \equiv r$ and $y \equiv t$ because s and t have the same signature. The other rules do not apply because otherwise, as S and S_n have exactly the same equations, they would apply to S_n, which is impossible. Given that x and y are distinct, we can conclude that the procedure in Figure 6 fails on input S. By Proposition 6.3 then, S is satisfiable in E, which then entails that $x\theta_n \neq_E y\theta_n$.

Point 2 can be proven similarly to Point 1 by considering, for any variable v of $x\theta_n$, the abstraction system obtained from S_n by replacing its disequation by $x \not\equiv v$.[31] Again, the argument is based on the fact that v is distinct from x, which this time is a consequence of the fact that $\prec$ is acyclic over S_n. Also note that the acyclicity of S_n and the definition of θ_n imply that $v\theta_n = v$ for all variables v occurring in $x\theta_n$.

Finally, we prove Point 3 by induction on the "inverse height" of equations in S_n, similarly to what we did in the proof of Lemma 5.11. Where M is the maximum of the heights of all the equations in S_n, let κ be the function from the left-hand side variables of S_n into the non-negative integers such that

$$\kappa(v) \quad := \quad M - \mathsf{h}_n(v \equiv q)$$

for each $v \equiv q \in S_n$. Note that if $v_1 \equiv q_1, v_2 \equiv q_2$ are equations of S_n such that $(v_1 \equiv q_1) \prec (v_2 \equiv q_2)$, then $\kappa(v_1) > \kappa(v_2)$. Assume that $\mathsf{h}_n(x \equiv r) > 0$.

(Base case) If $\kappa(x) = 0$ then $x \equiv r$ is maximal in S_n w.r.t. $\prec$, which entails that $x\theta_n = r$. As in Lemma 5.10(4), we can show that r is in $G_i \setminus V$. For $x\theta_n$ to be in G_i^* then it is enough for it to be non-collapsing in E. But this is the case by Point 2 above.

(Induction Step) If $\kappa(x) > 0$, let $x_1, \ldots, x_m$ be r's variables. Then, $x\theta_n$ has the form $r(x_1\theta_n, \ldots, x_m\theta_n)$. We can argue exactly as in the base case that r is an element of $G_i \setminus V$ and is non-collapsing in E.

Now let $k \in \{1, 2\}$ with $k \neq i$. We show that $x_j\theta_n$ is an element of G_k^* for every $j \in \{1, \ldots, m\}$. In fact, if x_j is not in the domain of θ_n then $x_j\theta_n = x_j$, which is trivially in G_k^*. If x_j is in the domain of θ_n, then $x_j\theta_n = t_j\theta_n$ for some term $t_j \in T(\Sigma_k, V)$ such that $x_j \equiv t_j \in S_n$. From the fact that $(x \equiv r) \prec (x_j \prec t_j)$ because $x_j \in \mathcal{V}ar(r)$, we can conclude both that $\mathsf{h}_n(x_j \equiv t_j) > 0$ and $\kappa(x) > \kappa(x_j)$. It follows by induction hypothesis that $x_j\theta_n \in G_k^*$.

In conclusion, to show that $x\theta_n = r(x_1\theta_n, \ldots, x_m\theta_n)$ belongs to G_i^* it is enough to show that all the components of $\bar{r} := (x_1\theta_n, \ldots, x_m\theta_n)$ are pairwise inequivalent in E. Using the argument above about the form of of each $x_j\theta_n$, one can prove that a variable and a non-variable term of $\bar{r}$ are inequivalent by Point 2 whereas pairs of non-variable terms are inequivalent by Point 1. Finally, pairs of variable terms are inequivalent because E is non-trivial. ∎

We can now show that, given any term in $T(\Sigma_1 \cup \Sigma_2, V)$, it is possible to find an equivalent term in $T(\Sigma, G^*)$.

Proposition 6.7. *For every term $t \in T(\Sigma_1 \cup \Sigma_2, V)$, there is a term $t' \in T(\Sigma, G^*)$, effectively computable from t, such that $t =_E t'$.*

[31] The rule **Ident2** does not become applicable by this change of the disequation because v is not a left-hand side variable of S_n.

Proof. Since $V \subseteq G^*$ by construction, we only need to consider the case in which t is not a variable. Hence, assume that $t \in T(\Sigma_1 \cup \Sigma_2, V) \setminus V$.

Let v be a variable not in $\mathcal{V}ar(t)$ and let S_n be the final abstraction system generated by the procedure in Fig. 6 on input $S_0 := AS(v \not\equiv t)$. Then let $x \not\equiv y$ be the disequation of S_n and θ_n the substitution induced by S_n. We start by showing that $t =_E y\theta_n$.

By construction, S_0 has the form $\{v \not\equiv u\} \cup T$ with v not occurring in T. From the definition of the derivation rules used by the procedure it is easy to see that v is never replaced by other variables, which means that the disequation of S_n is in fact $v \not\equiv y$ and that $v\theta_n = v$. Then, by Proposition 3.3, Lemma 6.4 and Lemma 6.5 above it follows that the formulae:

$$(v \not\equiv t) \leftrightarrow \exists \bar{v}_0.S_0, \quad \exists \bar{v}_0.S_0 \leftrightarrow \exists \bar{v}_n.S_n, \quad \exists \bar{v}_n.S_n \leftrightarrow (v \not\equiv y\theta_n),$$

where $\bar{v}_j$ are the left-hand side variables of S_j for $j \in \{0, n\}$, are all valid in E. This entails that $E \models (v \equiv t) \leftrightarrow (v \equiv y\theta_n)$, from which it follows that $t =_E y\theta_n$.

Now notice that S_n has the form $\{v \neq y, y \equiv t_n\} \cup R$ where $t_n \in T(\Sigma_i, V)$ for $i = 1$ or $i = 2$, and that $y\theta_n = t_n\theta_n$. Let $s(\bar{r}) = NF_i(t_n)$ and $t' := s(\bar{r}\theta_n)$. As $t_n =_{E_i} s(\bar{r})$, it is immediate that

$$t =_E y\theta_n = t_n\theta_n =_E s(\bar{r}\theta_n) = t'.$$

It is also immediate that t' is effectively computable from $y\theta_n$, which was in turn computed from t. To prove the claim then it is enough to show that $t' \in T(\Sigma, G^*)$. We do that by showing that $r\theta_n \in G^*$ for all components r of $\bar{r}$.

Let $k \in \{1, 2\}$ with $k \neq i$. First consider the case is which r is some variable v. If v is not in the domain of θ_n, $v\theta_n$ is trivially in G^*. If v is in the domain of θ_n, it must occur in the (Σ_i-)term t_n because of our usual assumption that the extra variables of a normal form, if any, are fresh. Moreover, v must be the left-hand side of a Σ_k-equation of S_n with non-zero height. In that case, $v\theta_n \in G_k^* \subseteq G^*$ as a consequence of Lemma 6.6(3).

Now suppose that r is a non-variable term of G_i and let $v_1, \ldots, v_m$ be its variables. By Assumption 4.1 we know that r is non-collapsing in E_i, and so non-collapsing in E as well by Proposition 4.14. Using again the fact that every variable of r that is in the domain of θ_n must occur in t_n, we can argue as in the previous case that $v_j\theta_n \in G_k^*$ for all $i \in \{1, \ldots, m\}$. As in the proof Lemma 6.6 then, we can show that $r(v_1\theta_n, \ldots, v_m\theta_n)$ satisfies all the conditions to be in G_k^*, which means that $r\theta_n = r(v_1\theta_n, \ldots, v_m\theta_n)$ is in G^*.

It follows that $t' = s(\bar{r}\theta_n)$ is an element of $T(\Sigma, G^*)$. ∎

From what we have seen so far, G^* satisfies the first two requirements in Definition 4.6 to be a Σ-base of E. To show that it satisfies the third, we will use the following additional result about the model $\mathcal{A}$ of E constructed in Subsection 4.3 as a fusion of the countably infinitely generated E_i-free algebras $\mathcal{A}_i$ ($i = 1, 2$).

LEMMA 6.8. *Where $\mathcal{A}$ is the algebra given in Lemma 4.13, let α be an arbitrary bijective valuation of V onto Z_2. Then, for all $i = 1, 2$ and all $t_1, t_2 \in G_i^* \setminus V$,*

1. $[\![t_1]\!]_\alpha^\mathcal{A} \in Z_{2,i}$.
2. $t_1 =_E t_2$ *iff* $[\![t_1]\!]_\alpha^\mathcal{A} = [\![t_2]\!]_\alpha^\mathcal{A}$.

Proof. Let $i \in \{1,2\}$ and consider two terms $t_1, t_2 \in G_i^* \setminus V$. We prove both claims simultaneously by induction on the number of layers of t_1 and t_2 (cf. observation after Definition 6.1).

(Base case) If both t_1 and t_2 have just one layer, we know that they are non-collapsing terms of $G_i \setminus V$. Then, Point 1 holds by Lemma 4.16 as $Z_2 \subseteq X_i'$. To prove Point 2, assume that $[\![t_1]\!]_\alpha^\mathcal{A} = [\![t_2]\!]_\alpha^\mathcal{A}$. Since both t_1 and t_2 are Σ_i-terms, this means that $\mathcal{A}^{\Sigma_i}, \alpha \models t_1 \equiv t_2$ where $\mathcal{A}^{\Sigma_i}$ is free in E_i over X_i' by Lemma 4.13 and α is an injection of $\mathcal{V}ar(t_1 \equiv t_2)$ into X_i' by construction. It follows by Proposition 2.1 that $r =_{E_i} t$, and so $r =_E t$.

(Induction step) Let $k \in \{1,2\}, k \neq i$. If either t_1 or t_2 (or both) has more than one layer, then, for $\iota = 1, 2$, t_ι has the form

$$r_\iota(\bar{v}_\iota, \bar{r}_\iota)$$

where $r_\iota \in G_i \setminus V$, $\bar{v}_\iota \subseteq V$ and $\bar{r}_\iota \subseteq G_k^* \setminus V$—with either $\bar{v}_\iota$ or $\bar{r}_\iota$ possibly empty. Let $\bar{b}_\iota$ be the tuple of values that α assigns, in order, to the variables in $\bar{v}_\iota$, and $\bar{c}_\iota$ the tuple consisting, in order, of all the elements $[\![r]\!]_\alpha^\mathcal{A}$ with $r \in \bar{r}_\iota$.

To prove Point 1, first notice that $\bar{b}_\iota \subseteq Z_2$ by definition of α and $\bar{c}_\iota \subseteq Z_{2,k}$ by induction hypothesis. It is immediate that $\bar{b}_\iota$ contains no repetitions and has no elements in common with $\bar{c}_\iota$.[32] We claim that $\bar{c}_\iota$ contains no repetitions either. In fact, if $[\![r]\!]_\alpha^\mathcal{A} = [\![r']\!]_\alpha^\mathcal{A}$ for two distinct $r, r' \in \bar{r}_\iota$, we know by induction hypothesis that $r =_E r'$. But this contradicts the fact that the tuple $\bar{r}_\iota$ satisfies Definition 6.1. From the above it is now easy to see that there is a bijective renaming r_ι' of r_ι and an injective valuation α' of $\mathcal{V}ar(r_\iota')$ into $X_i' = Z_{2,k} \cup Z_2$ such that $[\![t_\iota]\!]_\alpha^\mathcal{A} = [\![r_\iota(\bar{v}_\iota, \bar{r}_\iota)]\!]_\alpha^\mathcal{A} = [\![r_\iota']\!]_{\alpha'}^\mathcal{A}$. The claim that $[\![t_1]\!]_\alpha^\mathcal{A} \in Z_{2,i}$ then follows again by Lemma 4.16.

To prove Point 2, assume that $[\![t_1]\!]_\alpha^\mathcal{A} = [\![t_2]\!]_\alpha^\mathcal{A}$ and therefore

$$\mathcal{A}, \alpha \models r_1(\bar{v}_1, \bar{r}_1) \equiv r_2(\bar{v}_2, \bar{r}_2) \ .$$

Let $\bar{u}_1, \bar{u}_2$ be tuples of variables abstracting $\bar{r}_1, \bar{r}_2$ in the equation above so that E-equivalent terms are abstracted by the same variable. From the proof of Point 1, it is clear that there is an injective valuation β into $X_i' = Z_{2,k} \cup Z_2$ such that

$$\mathcal{A}, \beta \models r_1(\bar{v}_1, \bar{u}_1) \equiv r_2(\bar{v}_2, \bar{u}_2) \ .$$

Since $r_1(\bar{v}_1, \bar{u}_1), r_2(\bar{v}_2, \bar{u}_2)$ are both Σ_i-terms and $\mathcal{A}^{\Sigma_i}$ is free in E_i over X_i', we can conclude that $r_1(\bar{v}_1, \bar{u}_1) =_{E_i} r_2(\bar{v}_2, \bar{u}_2)$, and so $r_1(\bar{v}_1, \bar{u}_1) =_E r_2(\bar{v}_2, \bar{u}_2)$. From this it easily follows that

$$t_1 = r_1(\bar{v}_1, \bar{r}_1) =_E r_2(\bar{v}_2, \bar{r}_2) = t_2$$

as well. ∎

We are now finally ready to prove that Σ is a set of constructors for E as well.

PROPOSITION 6.9. *G^* is a Σ-base of E.*

Proof. We show that G^*, E, and Σ satisfy Definition 4.6.

Now, Condition 1 of Definition 4.6 is a consequence of the definition of G^*, whereas Condition 2 holds by Proposition 6.7. To prove Condition 3 consider again the algebra $\mathcal{A}$ and the valuation α of the previous lemma.

[32] Recall that Z_2 and $Z_{2,k}$ are disjoint.

Let $s_1(\bar{r}_1), s_2(\bar{r}_2)$ be terms in $T(\Sigma, G^*)$ and $s_1(\bar{v}_1), s_2(\bar{v}_2)$ the terms obtained from them by abstracting E-equivalent terms in $\bar{r}_1, \bar{r}_2$ with the same variable. Clearly $s_1(\bar{v}_1) =_E s_2(\bar{v}_2)$ implies $s_1(\bar{r}_1) =_E s_2(\bar{r}_2)$. Therefore, suppose that $s_1(\bar{r}_1) =_E s_2(\bar{r}_2)$. Since $\mathcal{A}$ is a model of E, $s_1(\bar{r}_1) =_E s_2(\bar{r}_2)$ entails that

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

Recall that $\mathcal{A}^\Sigma$ is free in E^Σ over $Y_2 = Z_{2,1} \cup Z_2 \cup Z_{2,2}$ and notice that, by Lemma 6.8, $[\![r]\!]^{\mathcal{A}}_\alpha \in Y_2$ for all $r \in G^*$. From this it is again easy to see that there is an injective valuation β of the variables of $\bar{v}_1, \bar{v}_2$ into the generators of $\mathcal{A}^\Sigma$ such that $\mathcal{A}^\Sigma, \beta \models s_1(\bar{v}_1) \equiv s_2(\bar{v}_2)$. It follows by Proposition 2.1 that $s_1(\bar{v}_1) =_{E^\Sigma} s_2(\bar{v}_2)$, which implies immediately that $s_1(\bar{v}_1) =_E s_2(\bar{v}_2)$. ∎

To sum up, we have obtained the following strong modularity result:

THEOREM 6.10. *Let E_1, E_2 be two equational theories with respective signatures Σ_1, Σ_2 such that, for $i = 1, 2$*

- $\Sigma := \Sigma_1 \cap \Sigma_2$ *is a set of constructors for* E_i*;*
- E_i *is non-trivial and* ${E_1}^\Sigma = {E_2}^\Sigma$*;*
- E_i *admits a recursive* Σ*-base* G_i *closed under bijective renaming of* V*;*
- G_i*-normal forms are computable for* Σ *and* E_i*;*
- *the word problem for* E_i *is decidable.*

Then the following holds:

1. Σ *is a set of constructors for* $E := E_1 \cup E_2$.
2. E *is non-trivial and* $E^\Sigma = {E_1}^\Sigma = {E_2}^\Sigma$.
3. E *admits a recursive* Σ*-base* G^* *closed under bijective renaming of* V*;*
4. G^**-normal forms are computable for* Σ *and* E*;*
5. *The word problem for* E *is decidable.*

Proof. Point 1 holds by Proposition 6.9 and Theorem 4.7; Point 2 holds by Corollary 4.15; Point 3 holds by Proposition 6.2, Proposition 6.9, and the definition of G^*; given Point 3, Point 4 holds by Proposition 6.7; finally, Point 5 holds by Theorem 5.13. ∎

Because of its complete modularity, the above result extends immediately by iteration to the combination of more than two theories, all pairwise sharing the same set of constructors Σ and having the same Σ-restriction.

7. RELATED WORK

Before comparing this paper with work by others, we briefly comment on its origins. The notion of a *fusion* is taken straight from a joint work [26] of the second author with Christophe Ringeissen, where it is given for arbitrary first-order structures, not just for algebras (see [27] for an up-to-date account of this work). Proposition 4.2 and Proposition 4.3 were also first proved in [26], again in the more general setting of first-order logic, not just equational theories. We have provided an explicit proof of Proposition 4.3 here both because it is simpler for algebras, and because we need the fusion construction employed in that proof to obtain the algebra $\mathcal{A}$ in Lemma 4.13.

The notion of *constructors* presented here is a generalization (for the case of equational theories) of a notion developed in [26] in the more general context of first-order theories.[33] There, constructors are given a syntactical definition which states, in the terminology of this paper, that a signature Σ is a set of constructors for an equational theory E iff the set $G_E(\Sigma, V)$ defined in Proposition 4.9 is a Σ-base of E. In [26] it is also proved that a necessary condition for Σ to be a set of constructors for E in the sense just described is that E^Σ is collapse-free and the Σ-reduct of each free model of E over a countably infinite set X is a free model of E^Σ over a set including X. We were able to prove that this condition is also sufficient, and we adopted it as the (algebraic) *definition* of constructors in [4, 6], providing the syntactical version as an additional characterization. In [5, 7], we were then able to remove altogether the collapse-freeness requirement from the algebraic definition, and provide a syntactical characterization in terms of Σ-bases, as described in this paper.

The *rule-based combination procedure* described in this paper was first developed in [3] for the case of disjoint signatures. It was then extended to the case of theories sharing collapse-free constructors in [6], and finally to theories sharing constructors as described in this paper in [7]. Unfortunately, the combination procedures in [6, 7] were incomplete since the rule **Ident2** was missing. The completeness proofs given in [4, 5] contained an error,[34] which we have corrected in the present paper by providing a new completeness proof.

In the rest of this section we compare our modularity result on the decidability of the word problem with the few existing results in the literature for the case of component theories with symbols in common.

7.1. Combination of term rewriting systems

A finite, complete (i.e., confluent and terminating) term rewriting system for an equational theory E immediately yields a decision procedure for the word problem for E: one simply rewrites the two terms to be proven equivalent into normal form and then checks whether the produced normal forms are identical.

It follows that, whenever an equational theory E is the union of two theories both having a finite and complete term rewriting system, the word problem for E is decidable if the union of the two theories' term rewriting systems is itself complete. Therefore, the question arises whether the completeness of term rewriting systems is preserved under union.

Such modularity properties of term rewriting systems over *disjoint signatures* have been studied in detail. It has been shown that confluence is modular [29] whereas termination is not. In fact, in [28] it is shown that there exist two confluent and terminating rewrite systems over disjoint signatures whose union is non-terminating. Thus, in general the union of two complete term rewriting systems need not be complete. However, it can be shown that it is at least semi-complete

[33]That notion was in turn inspired by that in [9], which we discuss in more detail in Subsection 7.2.

[34]In both proofs it is said that one "can restrict the attention to the case $i = 1$, as the other case (which is even simpler) can be treated analogously", which unfortunately is not true.

(i.e., confluent and normalizing[35]), which is actually sufficient to obtain a decision procedure for the word problem.

This result has been extended to the non-disjoint case, again using an appropriate notion of constructors. In the literature on the modularity properties of term rewriting systems, a constructor is a function symbol not occurring at the top of a rule's left-hand side. For term rewriting systems sharing constructors in this sense, it can be shown that semi-completeness is a modular property (see, e.g., [20] for details).

In [27] it is shown that for semi-complete term rewriting systems this notion of constructors is in fact a special case of ours; therefore, the combination results for decision procedures for the word problem obtainable from the work on modularity properties of term rewriting systems are subsumed by those presented here.

Our results are, however, more general in two respects. First, the notion of constructors is strictly more general, and second we do not assume that the word problem in the component theories is decided by a (semi-)complete term rewriting system. The applicability of our approach does not depend on whether the decision procedures for the component theories are based on term rewriting or not.

7.2. Combination of theories sharing DKR-constructors

As mentioned in the introduction, the first work to present explicit combination results for the word problem in the case of equational theories with symbols in common was [9]. There too the shared symbols are required to be constructors in a certain sense.

In this subsection, we investigate the connection between the notion of constructors presented here and the one presented in [9]. We show that that notion is a special case of ours, and that the combination result for the word problem in [9] (Theorem 14) can be obtained as a corollary of our Theorem 5.13.

To be able to define the notion of constructors according to [9], 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 $f(\dots, s, \dots) > f(\dots, t, \dots)$ for all $s, t \in T(\Omega, V)$ and all function symbols $f \in \Omega$. Notice that it is always possible to construct a well-founded and monotonic (total) ordering on $T(\Omega, V)$ for any functional signature Ω.[36]

In the rest of the subsection, we will consider a non-trivial equational theory E of signature Ω and a subsignature Σ of Ω.

DEFINITION 7.1. 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.* $>$ iff

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. $f(t_1, \dots, t_n){\downarrow}_E^{>} = f(t_1{\downarrow}_E^{>}, \dots, t_n{\downarrow}_E^{>})$ for all $f \in \Sigma$ and Ω-terms $t_1, \dots, t_n$.

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 7.1.

[35] A term rewriting system is normalizing if every term has a normal form. See, e.g., [1], Theorem 9.2.1 for the (simple) proof that this property is modular for term rewriting systems over disjoint signatures.

[36] For instance, one can take the lexicographic path ordering induced by a total well-founded precedence on $\Omega \cup V$ (see [1]), where the variables are treated as constants—which is admissible as the ordering is not required to be closed under substitutions.

LEMMA 7.2. *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 $i \in \{1, \ldots, m\}$.*

EXAMPLE 7.1. We show that, for the theory E_1 in Example 4.1, the signature Σ is a set of DKR-constructors w.r.t. an appropriate well-founded and monotonic ordering $>_1$.

First observe that the first two equations of E_1 define the associativity and commutativity of $+$. Let us call the theory axiomatized by these two equations AC. It is easy to show (and well-known) that orienting the other equations in E_1 from left to right yields a canonical term rewriting system R modulo AC. Here "modulo AC" means that, instead of syntactic matching, AC-matching is used when determining whether a rule is applicable (see, e.g., [12] for details). We denote by $\rightarrow_{R,AC}$ the rewrite relation induced by R modulo 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.

To obtain an appropriate well-founded and monotonic ordering $>_1$, we cannot simply take the transitive closure of the rewrite relation $\rightarrow_{R,AC}$. 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 is that the ordering $>$ "picks" a least representative out of the AC-equivalent $\rightarrow_{R,AC}$-normal forms in each E_1-equivalence class. Therefore, Condition 1 of Definition 7.1 is satisfied. That Condition 2 is also satisfied is 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_2 of Example 4.2 since the restriction ${E_2}^{\Sigma'}$ of E_2 to Σ' is not the theory of syntactic equality on Σ'-terms. The same is true for the signature Σ'' and the theory E_3 of Example 4.3. Hence, a set of constructors in our sense need not be a set of DKR-constructors.

Let G be the set of terms defined as follows:

$$G \quad := \quad \{r \in T(\Omega, V) \mid r{\downarrow}_E^{>}(\epsilon) \notin \Sigma\}. \tag{8}$$

We prove our claim that DKR-constructors are a special case of ours by showing that G is a Σ-base of E whenever Σ is a set of DKR-constructors for E w.r.t. $>$.

LEMMA 7.3. *If Σ is a set of* DKR-*constructors for E w.r.t. $>$, then G is a Σ-base of E.*

Proof. We prove the claim by showing that the set G satisfies the three conditions of Definition 4.6.

(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 non-trivial, the term t must contain v. However, then $v > v{\downarrow}_E^{>} = t$, in contrast with 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 so belong to G by definition.

(3) Let $s_1(r_1, \ldots, r_k), s_2(r'_1, \ldots, r'_\ell) \in T(\Sigma, G)$, and assume that $s_1(v_1, \ldots, v_k)$, $s_2(v'_1, \ldots, v'_\ell)$ are obtained from $s_1(r_1, \ldots, r_k)$, $s_2(r'_1, \ldots, r'_\ell)$ by abstracting the terms $r_1, \ldots, r_k$, $r'_1, \ldots, r'_\ell$ so that two terms are replaced by the same variable iff they are equivalent in E. We must show that $s_1(r_1, \ldots, r_k) =_E s_2(r'_1, \ldots, r'_\ell)$ implies $s_1(v_1, \ldots, v_k) =_E s_2(v'_1, \ldots, v'_\ell)$ (since the converse is trivial).

If $s_1(r_1, \ldots, r_k) =_E s_2(r'_1, \ldots, r'_\ell)$, then their DKR-normal forms coincide (by Point 1 of Lemma 7.2). By Condition 2 of Definition 7.1 this implies that

$$s_1(r_1, \ldots, r_k){\downarrow}_E^{>} = s_1(r_1{\downarrow}_E^{>}, \ldots, r_k{\downarrow}_E^{>}) = s_2(r'_1{\downarrow}_E^{>}, \ldots, r'_\ell{\downarrow}_E^{>}) = s_2(r_1, \ldots, r_k){\downarrow}_E^{>}.$$

Since terms in the set $\{r_1{\downarrow}_E^{>}, \ldots, r_k{\downarrow}_E^{>}, r'_1{\downarrow}_E^{>}, \ldots, r'_\ell{\downarrow}_E^{>}\}$ do not start with a symbol from Σ and since two of these terms are syntactically equal iff the corresponding terms in $\{r_1, \ldots, r_k, r'_1, \ldots, r'_\ell\}$ are equivalent modulo E, this implies that $s_1(v_1, \ldots, v_k) = s_2(v'_1, \ldots, v'_\ell)$. ∎

From Theorem 4.7 we immediately obtain the following:

PROPOSITION 7.4. *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 4.5.*

Point (2) of the proof of Lemma 7.4 may seem to entail that normal forms for E and Σ are computable in the sense of Definition 4.10. This is not the case, however, because the argument in (2) actually relies on DKR-normal forms, whose computability is not assured by the sole assumption that Σ is a set of DKR-constructors for E w.r.t $>$. In [9], DKR-normal forms are shown to be computable by also assuming that the so-called symbol matching problem is decidable.

DEFINITION 7.5. We say that the *symbol matching problem on Σ modulo E* is decidable in $T(\Omega, V)$ iff there exists an algorithm that decides, for all $t \in T(\Omega, V)$, whether there is 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 theory E_1 of Example 4.1, 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., $\to_{R,AC}$). 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 in $\to_{R,AC}$.

As pointed out in [9], 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 they exist. In fact, once we know that an appropriate function symbol in Σ and a tuple of Ω-terms exist, we can simply enumerate all pairs consisting of a symbol $f \in \Sigma$ and a tuple $\bar{t}$ of Ω-terms,[37] 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_G for E and Σ with the following recursive definition.

DEFINITION 7.6. 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, where G is the set defined in (8), let NF_G be the function defined as follows: For every $t \in T(\Omega, V)$,

1. $NF_G(t) := f(NF_G(t_1), \dots, NF_G(t_n))$ if t matches onto Σ modulo E, and f is the Σ-symbol and $(t_1, \dots, t_n)$ the tuple of Ω-terms returned by M on input t.
2. $NF_G(t) := t$, otherwise.

LEMMA 7.7. *Under the assumptions of Definition 7.6 the function NF_G is well-defined and satisfies the requirements of Definition 4.10.*

Proof. To start with, we know from Lemma 7.3 that G is indeed a Σ-base of E. Now, to show that NF_G 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 $t_1, \dots, t_n$ are smaller than t w.r.t. this ordering.

We define this ordering using a mapping α from $T(\Omega, V)$ into the non-negative 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 not belonging 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 $t =_E f(t_1, \dots, t_n)$ for some $f \in \Sigma$, then $\alpha(t) > \alpha(t_i)$ for all $i \in \{1, \dots, n\}$. But this is an easy consequence of the fact that $t\downarrow_E^{>} = f(t_1, \dots, t_n)\downarrow_E^{>} = f(t_1\downarrow_E^{>}, \dots, t_n\downarrow_E^{>})$. In conclusion, we have shown that NF_G is well-defined.

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

Now we prove by well-founded induction on $\succ$ that $NF_G(t)$ is a normal form of t. When the second case of Definition 7.6 applies, t belongs to G by definition, which entails immediately that $NF_G(t) = t$ is in normal form. When the first case applies, we know that $NF_G(t) = f(NF_G(t_1), \dots, NF_G(t_n))$ for some Σ-symbol f and tuple $(t_1, \dots, t_n)$ such that $t =_E f(t_1, \dots, t_n)$. As we have seen above, $t \succ t_i$ for all $i \in \{1, \dots, n\}$, which entails by induction that $NF_G(t_i)$ is a normal form of t_i for each $i \in \{1, \dots, n\}$. Since $f \in \Sigma$, it is immediate that $f(NF_G(t_1), \dots, NF_G(t_n))$ is in normal form as well. To see that $NF_G(t)$ is indeed a normal form of t, it is now enough to observe that $t =_E f(t_1, \dots, t_n) =_E f(NF_G(t_1), \dots, NF_G(t_n))$, where the last equivalence is a consequence of the induction assumption that $t_i =_E NF_G(t_i)$ for each $i \in \{1, \dots, n\}$. ∎

We are now ready to show that Theorem 14 in [9] can be obtained as a corollary of our Theorem 5.13.

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

Corollary 7.8. *Let E_1, E_2 be non-trivial equational theories of respective signature Σ_1, Σ_2 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 5.13 are satisfied. By Proposition 7.4, Σ is a set of constructors according to Definition 4.5 for both E_1 and E_2. By Point 2 of Lemma 7.2, ${E_1}^{\Sigma} = {E_2}^{\Sigma}$ 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 7.7. ∎

The notion of constructors presented in this paper is considerably more general than the one introduced in [9]: it has no restrictions for E^{Σ} whereas that in [9] imposes the very strong restriction that E^{Σ} must coincide with syntactic equality on Σ-terms. Another anvantage of our notion of constructors is that it has an abstract algebraic definition whereas the definition of DKR-constructors is rather technical and depends strongly on the chosen ordering $>$.

7.3. Combination of theories constructible over a common subtheory

In this subsection, we compare our results to those published in a recent work by Fiorentini and Ghilardi. In [10], they introduce a method for combining decision procedures for the word problem that differs significantly from both the one in [9] and the one in this paper. Their declared goal is to improve on the work in [9] and our previous work in [6] by providing a method that manipulates terms using rewriting techniques, as done in [9], but at the same time has the same flexibility as our own in requiring no particular strategy in the application of the rewrite rules.

As in our work, the contributions of [10] can be decomposed in principle into three[38] parts:

1. Provide appropriate restrictions on the theories to be combined.
2. Describe a combination algorithm.
3. Prove that the combination algorithm is correct for all theories satisfying the restrictions introduced in 1.

Both the combination algorithm and the proof of correctness given in [10] differ considerably from ours. The algorithm is based on rewriting techniques and its correctness is proved within a categorical framework. The restrictions on the theories are also introduced within the categorical framework. However, the authors do provide an algebraic version of these restrictions and show that they are in fact more general than those we presented in [6]—which already subsumed those in [9].

[38] Actually, our own work contains a fourth contribution: the proof that the restrictions on the theories are themselves modular (Section 6). Fiorentini and Ghilardi do not explicitly provide such a modularity result in [10]. In principle, however, it should be possible to produce it in the framework of [10] as well.

It can be shown, however, that they are just as general as the results presented here.[39]

The main restriction introduced in [10] is that the component theories E_1 and E_2 are *constructible* over a common subtheory in the shared signature. This notion of constructibility is intimately related to our notion of constructors, as they point out in [10] and we are going to illustrate below. The actual definition of constructibility given in [10] involves category theory concepts, such as factorization systems and left extensions, which are out of the scope of this paper but are essential to prove the confluence and termination of the rewrite system used in the combination algorithm. Fortunately, [10] also contains a characterization of constructibility in algebraic terms (Proposition 10.4), which will be good enough for this paper. For comparison's sake, we paraphrase it here in the terminology of this paper, and use it as an algebraic definition of constructibility. With a slight abuse of notation, we will write $E = E'$ for two equational theories E, E' if the two theories entail exactly the same equations.

DEFINITION 7.9 (Constructibility). Let E_0 be a non-trivial equational theory of signature Σ, and E an equational theory of signature $\Omega \supseteq \Sigma$ such that $E^{\Sigma} = E_0$. Then, E is *constructible over* E_0 iff the Σ-reduct of every free model $\mathcal{A}$ of E over some set X of generators is a free model of E_0 over a set of generators Y such that

- $X \subseteq Y$,
- Y is invariant under all Ω-automorphisms of $\mathcal{A}$ that are an extension of a bijection of X onto itself.

It is shown in [10] that the notion above strictly subsumes the notion of constructors we used in [6] (where we required the restriction of the theory to the constructor signature to be collapse-free). However, this is not true anymore for the more general notion of constructors we already had in [5, 7], and also use in this paper, as one can easily see by comparing the definition above with Definition 4.5.

Fiorentini and Ghilardi provide a syntactical characterization of constructibility as well in [10]. As it turns out, this characterization is substantially equivalent to our own syntactical characterization of constructors in Theorem 4.7. On the surface, their syntactical conditions seem more restrictive than ours. First, the sets that in [10] correspond to our Σ-bases[40] are all closed under renaming of variables. Second, the normal forms of terms over these sets must satisfy more conditions than we have in Definition 4.6(2). As the authors themselves show, however, these conditions are just technical restrictions that simplify proofs; they can be assumed without loss of generality. As for the closure under renaming, although we do not embed it into our definition of a Σ-base, we do need it anyway for our combination results. In conclusion, Fiorentini and Ghilardi's constructibility can be characterized in terms of our Σ-bases as follows.

PROPOSITION 7.10. *Let E_0 be a non-trivial equational theory of signature Σ, and E an equational theory of signature $\Omega \supseteq \Sigma$ such that $E^{\Sigma} = E_0$. Then, E is* constructible over E_0 *iff E admits a Σ-base closed under bijective renaming of V.*

For their decidability results, Fiorentini and Ghilardi use the notion of *effective constructibility*. In our terms, the theory E above is effectively constructible over

[39] Reportedly, at the time of their writing of [10], the authors were not aware of our own more general results, which we first reported in [5] and then published in [7].

[40] Namely, the sets denoted by E' in Proposition 10.1 of [10].

the theory E_0 iff it is constructible over E_0 and admits a Σ-base G (closed under renaming) such that, for every term $t \in T(\Omega, V)$, one can effectively compute a term $s(\bar{v}) \in T(\Sigma, V)$ and a tuple $\bar{r}$ of terms in G such that $t =_E s(\bar{r})$. It is not difficult to show that, for theories E with decidable word problem, effective constructibility corresponds exactly to computability of normal forms in our sense with respect to recursive Σ-bases closed under renaming.

Effective constructibility of component theories over the same subtheory yields the following main combination result in [10].

THEOREM 7.11. *Let Σ_1, Σ_2 be two signatures and let $\Sigma := \Sigma_1 \cap \Sigma_2$. Let E_0 be a non-trivial equational theory of signature Σ and, for $i = 1, 2$, let E_i be equational theories with signature Σ_i and decidable word problem such that ${E_i}^\Sigma = E_0$. If both E_1 and E_2 are effectively constructible over E_0, then $E_1 \cup E_2$ has a decidable word problem.*

Now, this results has exactly the same scope as our corresponding result in Theorem 5.13. In fact, consider two equational theories E_1, E_2 of signature Σ_1, Σ_2, respectively, both with decidable word problem. Let $\Sigma := \Sigma_1 \cap \Sigma_2$.

First assume that E_1, E_2 are equational theories satisfying the assumptions of Theorem 7.11. We show that the assumptions of Theorem 5.13 are satisfied as well.

Clearly, E_i (for $i \in \{1, 2\}$) is non-trivial since $E_0 = {E_i}^\Sigma$ was assumed to be non-trivial. By assumption, the word problem for E_i is decidable. By Proposition 7.10, E_i admits a Σ-base G_i closed under bijective renaming of V. From what we observed earlier, we can assume that G_i is recursive and G_i-normal forms are computable. Finally, ${E_1}^\Sigma = E_0 = {E_2}^\Sigma$, which shows that all the prerequisites for Theorem 5.13 are satisfied.

Conversely, assume that, for $i = 1, 2$, Σ is a set of constructors for E_i, E_i is non-trivial and admits a recursive Σ-base G_i closed under bijective renaming of V, and G_i-normal forms are computable. Furthermore, assume that ${E_1}^\Sigma = {E_2}^\Sigma$. It follows that Theorem 5.13 applies. We show that Theorem 7.11 also applies.

Let $i \in \{1, 2\}$ and $E_0 := {E_1}^\Sigma = {E_2}^\Sigma$. Clearly, E_0 is non-trivial as well. With Proposition 7.10 and Proposition 4.11 we can now conclude that E_i is effectively constructible over E_0.

In conclusion, we can say that the approach employed in [10], although based on completely different techniques and proofs, produces the same modularity result as ours on the decidability of the word problem in the combination of two equational theories with (possibly) non-disjoint signatures.

At the moment, it is not clear which approach to prefer. Both yield the same results, and also with about the same effort (like our paper, [10] is also fairly long). For the readers from the automated reasoning community, our approach (based on universal algebra) may be more accessible than the categorical approach used in [10], but probably this is a matter of taste. The main test case for both approaches will be whether they can be extended to more general combination problems, such as the combination of unification algorithms.

8. CONCLUSION AND OPEN QUESTIONS

In this paper, we have described a 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 [17], 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.

We have first presented (in Section 3) a procedure that can deal with equational theories over disjoint signatures, and then extended this procedure (in Section 5) to treat theories sharing symbols that we called constructors. This extension was achieved simply by adding three new rules for handling the constructor symbols. The reasons for a two-step presentation of the procedure were mainly didactic. 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 of the decidability of the word problem in the *disjoint case* has been known for quite some time [21, 24, 23, 19, 13]. Our main goal in Section 3 was the development of a *rule-based* combination procedure, which we believe is simpler and more flexible than the known ones because it uses rules that can be applied in arbitrary order.

This not only provides for more transparent proofs, as we think we have demonstrated, but it also leads to a rather general extension of the procedure to the *non-disjoint case*.

To our knowledge, the only other combination results for the word problem in the case of component theories with symbols in common are those presented in [9] and [10]. We have argued that our method is more flexible than the one presented in [9], and shown that, in addition, it applies to a more general class of theories than those considered in [9]. Furthermore, we believe that our algebraic approach yields a less technical, and hence more transparent, definition of this class. The approach followed in [10] is very flexible too, and applies to the same class of theories as ours, as we have shown. Most likely then, preferring one approach over the other should be a matter of personal background and taste: our combination method is based on (what is essentially) a derivation calculus, whereas that in [10] is based on a rewrite system; our semantical arguments are drawn from universal algebra, whereas those in [10] are drawn from category theory.

It should be noted that while the present paper (like [10]) is concerned only with the word problem, [9] also contains combination results for unification and matching, Thus, one direction for future research would be to extend our approach to the combination of decision procedures for the matching and the unification problem as well. Whether and how easily this is possible may be one of the main criteria for deciding whether to prefer our approach or the one in [10].

A further generalization would come from lifting our results to the case of many-sorted equational logic. This should not be very difficult, 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.

Finally, we would like to point out that the results presented here depend on two technical requirements. One is the requirement in the definition of constructors (Definition 4.5) that the set X of generators of the free algebra $\mathcal{A}$ be included in the set of generators of the reduct $\mathcal{A}^{\Sigma}$, and the other is the requirement in the (extended) combination procedure that the Σ-bases of the component theories be closed under renaming. In all the examples we have found so far, these requirements are either immediately satisfied or can be assumed to be satisfied with no loss of generality. Nonetheless, the question of whether they can be removed altogether is still open. To this regard it is interesting to notice that the authors of [10], who arrived at their results independently from us and through a completely different approach, need both requirements as well. This seems to indicate that there is indeed a fundamental (non-technical) reason for them.

ACKNOWLEDGMENTS

The second author would like to thank Christophe Ringeissen and Silvio Ghilardi for a number of discussions and clarifications about their own work on the subject of this paper and its relationship with the one presented here.

REFERENCES

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

[2] Franz Baader and Klaus U. Schulz. Combination of constraint solvers for free and quasi-free structures. *Theoretical Computer Science*, 192:107–161, 1998.

[3] 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.

[4] Franz Baader and Cesare Tinelli. Deciding the word problem in the union of equational theories. Technical Report UIUCDCS-R-98-2073, Department of Computer Science, University of Illinois at Urbana-Champaign, October 1998.

[5] Franz Baader and Cesare Tinelli. Combining equational theories sharing non-collapse-free constructors. Technical Report 99-13, Department of Computer Science, University of Iowa, October 1999.

[6] Franz Baader and Cesare Tinelli. Deciding the word problem in the union of equational theories sharing constructors. In P. Narendran and M. Rusinowitch, editors, *Proceedings of the 10th International Conference on Rewriting Techniques and Applications (Trento, Italy)*, volume 1631 of *Lecture Notes in Computer Science*, pages 175–189. Springer-Verlag, 1999.

[7] Franz Baader and Cesare Tinelli. Combining equational theories sharing non-collapse-free constructors. In H. Kirchner and Ch. Ringeissen, editors, *Proceedings of the 3rd International Workshop on Frontiers of Combining Systems, FroCoS'2000, Nancy (France)*, volume 1794 of *Lecture Notes in Artificial Intelligence*, pages 260–274. Springer-Verlag, March 2000.

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

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

[10] Camillo Fiorentini and Silvio Ghilardi. Path rewriting and combined word problems. Technical Report 250-00, Department of Computer Science, Università degli Studi di Milano, Milan, Italy, May 2000. (Revised version to appear in Theoretical Computer Science).

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

[12] J.-P. Jouannaud and H. Kirchner. Completion of a set of rules modulo a set of equations. *SIAM J. Computing*, 15(4):1155–1194, 1986.

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

[14] 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.

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

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

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

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

[19] 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.

[20] Enno Ohlebusch. Modular properties of composable term rewriting systems. *Journal of Symbolic Computation*, 20(1):1–41, 1995.

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

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

[23] Manfred Schmidt-Schauß. Unification in a combination of arbitrary disjoint equational theories. *Journal of Symbolic Computation*, 8(1–2):51–100, July/August 1989. Special issue on unification. Part II.

[24] Erik Tidén. *First-Order Unification in Combinations of Equational Theories.* PhD dissertation, The Royal Institute of Technology, Stockholm, 1986.

[25] 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 Academic Publishers, March 1996.

[26] 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).

[27] Cesare Tinelli and Christophe Ringeissen. Unions of non-disjoint theories and combinations of satisfiability procedures. *Theoretical Computer Science*, 2001. (to appear).

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

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