Deciding the Word Problem for Ground and Strongly Shallow Identities w.r.t. Extensional Symbols

Franz Baader · Deepak Kapur

Received: 15 April 2021 / Accepted: 3 January 2022 / Published online: 7 May 2022
© The Author(s) 2022

Abstract
The word problem for a finite set of ground identities is known to be decidable in polynomial time using congruence closure, and this is also the case if some of the function symbols are assumed to be commutative or defined by certain shallow identities, called strongly shallow. We show that decidability in P is preserved if we add the assumption that certain function symbols \( f \) are extensional in the sense that \( f(s_1, \ldots, s_n) \approx f(t_1, \ldots, t_n) \) implies \( s_1 \approx t_1, \ldots, s_n \approx t_n \). In addition, we investigate a variant of extensionality that is more appropriate for commutative function symbols, but which raises the complexity of the word problem to coNP.

Keywords Word problem for ground identities · Semantic congruence closure · Interpreted function symbols · Shallow identities · Extensionality · Ordered Rewriting · Description Logics

1 Introduction

One motivation for this work stems from Description Logic (DL) [1], where constant symbols (called individual names) are used within knowledge bases to denote objects or individuals in an application domain. If such objects are composed of other objects, it makes sense to represent them as (ground) terms rather than constants. For example, the couple consisting of individual \( a \) in the first component and individual \( b \) in the second component is more reasonably represented by the term \( f(a, b) \) (where \( f \) is a binary function symbol denoting the couple constructor) than by a third constant \( c \) that is unrelated to \( a \) and \( b \). In fact, if we have two couples, one consisting of \( a \) and \( b \) and the other of \( a' \) and \( b' \), and we learn (by DL reasoning or from external sources) that \( a \) is equal to \( a' \) and \( b \) is equal to \( b' \), then this automatically implies that \( f(a, b) \) is equal to \( f(a', b') \), i.e., that this is one and the same...
couple, whereas we would not obtain such a consequence if we had introduced constants \( c \) and \( c' \) for the two couples.

If we use terms to represent objects, and can learn (e.g., by DL reasoning) that two terms are supposed to be equal, we need to be able to decide which other identities between terms can be derived from the given ones. Fortunately, this problem (usually called the word problem for ground identities) is decidable in polynomial time. The standard approach for deciding this word problem is congruence closure [2–5]. Basically, congruence closure starts with the given set of ground identities \( E \), and then extends it using closure under reflexivity, symmetry, transitivity, and congruence. The set \( CC(E) \) obtained this way is usually infinite, and the main observation that yields decidability in polynomial time is that one can restrict it to the subterms of \( E \) and the subterms of the terms for which one wants to decide the word problem. An alternative approach for deciding the word problem for ground identities is based on term rewriting. Basically, in this approach one generates an appropriate canonical term rewriting system from \( E \), and then decides whether two terms are equal modulo the theory \( E \) by computing their canonical forms and checking whether they are syntactically equal. This was implicit in [6], and made explicit in [7] (see also [8, 9] for other rewriting-based approaches).

In the motivating example from DL, but also in other settings where congruence closure is employed (such as SMT [10, 11]), it sometimes makes sense to assume that certain function symbols satisfy additional semantic properties that are not expressible by ground identities. Such symbols are then often called interpreted symbols, in contrast to the uninterpreted ones, for which no such properties are required. For example, one may want to consider couples where the order of the components is irrelevant, which means that the couple constructor function is commutative. Another interesting property for (ordered) couples is extensionality: if two couples are equal then they must have the same first and second components, i.e., the couple constructor \( f \) must satisfy the extensionality axiom \( f(x, y) ≈ f(x', y') \Rightarrow x ≈ x' ∧ y ≈ y' \). While it is well-known that adding commutativity does not increase the complexity (see, e.g., [3, 12]), extensionality has, to the best of our knowledge, not been considered in this context before. The problem with extensionality is that it allows us to derive “small” identities from larger ones. Consequently, it is conceivable that one first needs to generate such large identities using congruence and other rules, before one can get back to a smaller one through the application of extensionality. Thus, it is not obvious that also with extensionality one can restrict congruence closure to a finite set of terms determined by the input.

Here, we will tackle this problem using a rewriting-based approach. Our proofs imply that, also with extensional symbols, proofs of identities that detour through “large” terms can be replaced by proofs using only “small” terms, but it is not clear how this could be shown directly without the rewriting-based approach introduced in this paper.

The conference version of this paper [13] shows how the rewriting-based approach of [7] can be extended to also handle commutative symbols. In contrast to approaches that deal with associative-commutative (AC) symbols [14, 15] using rewriting modulo AC, commutativity is treated by introducing an additional ground rewrite system \( R(\Sigma_c) \) consisting of all appropriately ordered ground instances of commutativity. In Sect. 3, we extend this approach from commutativity to more general shallow identities [16] of the form \( h(x_1, \ldots, x_n) ≈ h(y_1, \ldots, y_n) \) where \( \{x_1, \ldots, y_n\} = \{y_1, \ldots, y_n\} \) is a set of (not necessarily distinct) variables. Although it deals with a more general setting, we were able to design a simpler congruence closure algorithm. Basically, it applies the classical rewriting-based congruence closure for uninterpreted function symbols. Shallow identities of the above form, called strongly shallow in this paper, are dealt with by reducing the left-hand sides of
the generated rules using an additional ground rewrite system induced by the strongly shallow identities. This ground system is infinite, but finitely represented by an ordered rewrite system\footnote{In [13] it was shown how to extend the commutative congruence closure algorithm developed there to deal also with (non-commutative) extensional symbols. In Sect. 4, we not only extend this approach from commutativity to strongly shallow identities. Instead, we exhibit general properties that the canonical rewrite system computed by a semantic congruence closure algorithm must satisfy such that extensionality for otherwise uninterpreted function symbols can be handled by a simple iterative algorithm that derives new identities between constants due to extensionality. As already observed in [13], it does not make sense to consider symbols that are both commutative and extensional since this makes the induced equational theory trivial. Instead, we introduce, in [13] and in Sect. 5 of the present paper, the notion of c-extensionality, which is more appropriate for commutative symbols, but increases the complexity of the word problem from polynomial to \( \text{coNP-complete}\). Section 6 gives a more detailed description of related work.}

In [13] we define our notion of strongly shallow identities, and show how such identities give rise to an ordered rewrite system that is canonical on ground terms.

2 Preliminaries

We assume that the reader is familiar with basic notions and results regarding equational theories, universal algebra, and term rewriting. In this section, we first recall the relevant notions and results regarding equational theories and term rewriting, but refer the readers to [5] for details. We will keep as close as possible to the notation introduced in [5]. In particular, we use \( \approx \) to denote identities between terms and \( = \) to denote syntactic equality. Then, we introduce congruence closure, both without and with interpreted function symbols. Finally, we define our notion of strongly shallow identities, show how such identities give rise to an ordered rewrite system that is canonical on ground terms.

2.1 Equational Theories and Term Rewriting

Terms are built as usual from variables, constants, and function symbols. A term of the form \( f(t_1, \ldots, t_n) \) for an \( n \)-ary function symbol \( f \) is said to have root symbol \( f \). An identity is a pair of terms \( (s, t) \), which we usually write as \( s \approx t \). A ground term is a term not containing variables and a ground identity is a pair of ground terms. Given a set of (not necessarily ground) identities \( F \), the equational theory induced by \( F \) is defined semantically as \( \approx_F := \{ s \approx t \mid \text{every model of } F \text{ is a model of } s \approx t \} \). The notion of model used here is the usual one from first-order logic, where the equality symbol is interpreted as the equality relation on the domain and we assume that identities are (implicitly) universally quantified. Since we consider signatures consisting only of constant and function symbols, we call first-order interpretations algebras.

Birkhoff’s theorem (see Theorem 4.5.14 in [5]) provides us with an alternative syntactic characterization of \( \approx_F \) that is based on rewriting. A given set of identities \( F \) induces a binary relation \( \rightarrow_F \) on terms. Basically, we have \( s \rightarrow_F t \) if there is an identity \( \ell \approx r \) in \( F \) such that \( s \) contains a substitution instance \( \sigma(\ell) \) of \( \ell \) as subterm, and \( t \) is obtained from \( s \) by replacing this subterm with \( \sigma(r) \). Birkhoff’s theorem says that \( \approx_F \) is identical to \( \leftrightarrow_F \), where \( \leftrightarrow_F \) denotes the reflexive, transitive, and symmetric closure of \( \rightarrow_F \).
The normal form of a term $s$ is an irreducible term $\hat{s}$ such that $s \rightarrow^* F \hat{s}$, where $\rightarrow^*_F$ denotes the reflexive and transitive closure of $\rightarrow_F$ and $\hat{s}$ is irreducible if there is no $s'$ with $\hat{s} \rightarrow_F s'$. If $\rightarrow_F$ is canonical (i.e., terminating and confluent), then every term has a unique normal form, which we then called its canonical form. In addition, $s \leftrightarrow^*_F t$ iff $s$ and $t$ have the same canonical forms. Termination ensures that the normal form exists and confluence that it is unique. The relation $\rightarrow_F$ is confluent if $s \rightarrow_F t_1$ and $s \rightarrow_F t_2$ imply that the pair of terms $(t_1, t_2)$ is joinable, i.e., there is a term $t$ such that $t_1 \rightarrow^*_F t$ and $t_2 \rightarrow^*_F t$. It is terminating if there is no infinite chain $t_0 \rightarrow_F t_1 \rightarrow_F t_2 \rightarrow_F \cdots$. Termination can be proved using a reduction order, which is a well-founded order $\prec$ on terms such that $\ell > r$ for all $\ell \approx r \in F$ implies $s > t$ for all terms $s, t$ with $s \rightarrow_F t$. Since $>$ is well-founded this then implies termination. If $\rightarrow_F$ is terminating, then confluence can be tested by checking whether all critical pairs of $F$ are joinable. Basically, critical pairs $(t_1, t_2)$ consider the most general “forks” of the form $s \rightarrow_F t_1$ and $s \rightarrow_F t_2$ that are due to overlapping left-hand sides of identities. They can be obtained by computing the most general unifiers of left-hand sides of rules with subterms of renamed variants of left-hand sides of other or the same rule. To be more precise, if $\ell \approx r$ and $g \approx d$ are identities (whose variables have been renamed apart) such that a non-variable subterm $\ell'$ of $\ell$ unifies with $g$ with most general unifier $\sigma$, then this overlap induces the critical pair $(t_1, t_2)$ where $t_1 = \sigma(r)$ and $t_2$ is obtained from $\sigma(\ell)$ by replacing the subterm $\sigma(\ell')$ with $\sigma(d)$. Such a critical pair $(t_1, t_2)$ is joinable if there is a term $t$ such that $t_1 \rightarrow^*_F t$ and $t_2 \rightarrow^*_F t$.

When considering the relation $\rightarrow_F$, one calls $F$ a term rewriting system rather than a set of identities, and writes its elements (called rewrite rules) as $\ell \rightarrow r$ instead of $\ell \approx r$. From a formal point of view, however, both rewrite rules and identities are (ordered) pairs of terms. Given a set of such pairs, we can view it as a set of identities or a term rewriting system, and thus the notions introduced above apply to both.

2.2 Congruence Closure

Let $\Sigma$ be a finite set of function symbols of arity $\geq 1$ and $C_0$ a finite set of constant symbols. We denote the set of ground terms built using symbols from $\Sigma$ and $C_0$ with $G(\Sigma, C_0)$. In the following, let $E$ be a finite set of ground identities $s \approx t$ between terms $s, t \in G(\Sigma, C_0)$, and $\approx_E$ the equational theory induced by $E$ on $G(\Sigma, C_0)$.

2.2.1 Classical Congruence Closure

It is well known (see, e.g., [5], Lemma 4.3.3) that $\approx_E$ (viewed as a subset of $G(\Sigma, C_0) \times G(\Sigma, C_0)$) can be generated using congruence closure, i.e., by exhaustively applying reflexivity, transitivity, symmetry, and congruence to $E$. To be more precise, $CC(E)$ is the smallest subset of $G(\Sigma, C_0) \times G(\Sigma, C_0)$ that contains $E$ and is closed under the following rules:

- if $s \in G(\Sigma, C_0)$, then $s \approx s \in CC(E)$ (reflexivity);
- if $s_1 \approx s_2$, $s_2 \approx s_3 \in CC(E)$, then $s_1 \approx s_3 \in CC(E)$ (transitivity);
- if $s_1 \approx s_2 \in CC(E)$, then $s_2 \approx s_1 \in CC(E)$ (symmetry);
- if $f$ is an $n$-ary function symbol and $s_1 \approx t_1, \ldots, s_n \approx t_n \in CC(E)$, then $f(s_1, \ldots, s_n) \approx f(t_1, \ldots, t_n) \in CC(E)$ (congruence).

The set $CC(E)$ is usually infinite. To obtain a decision procedure for the word problem for $E$ (i.e., for the relation $\approx_E$), one can show that it is sufficient to restrict the application of the above rules to a finite subset of $G(\Sigma, C_0)$, which consists of the subterms of terms occurring
in $E$ and of the subterms of the terms $s_0, t_0$ for which one wants to decide whether $s_0 \approx_E t_0$ holds or not (see, e.g., [5], Theorem 4.3.5).

### 2.2.2 Semantic Congruence Closure

As mentioned in the introduction, it sometimes makes sense to assume that certain function symbols are interpreted in the sense that they satisfy additional semantic properties that are not expressible by ground identities. Thus, we now consider a setting where there is a subset $\Sigma_F$ of $\Sigma$ and a set of (non-ground) identities $F$ formulated using the function symbols in $\Sigma_F$ and variables. Given a finite set $E$ of ground identities $s \approx t$ between terms $s, t \in G(\Sigma, C_0)$, we assume that also the identities in $F$ are satisfied. This means that we consider algebras $A$ that satisfy not only the ground identities in $E$, but also the universal closure of the non-ground identities in $F$, i.e., algebras $A$ that are models of $E \cup F$.

Given $s, t \in G(\Sigma, C_0)$, we say that $s \approx t$ follows from $E$ w.r.t. $F$ (written $s \approx^E_F t$) if $s^A \approx t^A$ holds in all models $A$ of $E \cup F$. Thus, $\approx^E_F$ is the restriction of $\approx_{E \cup F}$ to the ground terms in $G(\Sigma, C_0)$. By a slight abuse of notation, we will later use $\approx^E_F$ also to denote the restriction of $\approx_{E \cup F}$ to $G(\Sigma, C)$ for a set of constants $C$ containing $C_0$.

The relation $\approx^E_F \subseteq G(\Sigma, C_0) \times G(\Sigma, C_0)$ can again be generated by extending congruence closure by an additional rule that deals with the identities in $F$, i.e., $CC^F(E)$ is the smallest subset of $G(\Sigma, C_0) \times G(\Sigma, C_0)$ that contains $E$ and is closed under reflexivity, transitivity, symmetry, congruence, and the following rule:

- if $\ell \approx r \in F$ and $\sigma$ is a substitution that maps the variables in $\ell, r$ to elements of $G(\Sigma, C_0)$, then $\sigma(\ell) \approx \sigma(r) \in CC^F(E)$ (F-identities).

We call $CC^F(E)$ the congruence closure of $E$ w.r.t. $F$. Birkhoff’s theorem again shows that $CC^F(E)$ coincides with $\approx^E_F$.

Without restrictions on the set of identities $F$, the word problem for $E$ w.r.t. $F$ (i.e., the relation $\approx^E_F$) need not be decidable. For example, if $F$ axiomatizes associativity of a binary function symbol $f$, then this word problem corresponds to the word problem for finitely presented semigroups, which is known to be undecidable [18]. Thus, one needs to impose some restrictions on the set of identities $F$ to guarantee decidability of the word problem.

### 2.3 Strongly Shallow Identities and Ordered Rewrite Systems

In [13], we considered the setting where $\Sigma_F$ (denoted as $\Sigma_c$ in [13]) contains only binary function symbols and $F$ consists of the commutativity axioms $f(x, y) \approx f(y, x)$ for all $f \in \Sigma_F$. In the present paper, we extend the results of [13] from commutativity to more general kinds of semantic properties of function symbols that can be expressed by strongly shallow identities. Note that the notion of strongly shallow identities introduced in this paper is more restrictive than what is often called “shallow” in the literature [16] (see Sect. 6 for more details).

**Definition 1** Let $\Sigma_S \subseteq \Sigma$ be a set of function symbols and $h \in \Sigma_S$ a symbol in $\Sigma_S$ of arity $n$. A **strongly shallow h-identity** is of the form $h(x_1, \ldots, x_n) \approx h(y_1, \ldots, y_n)$ where $x_1, y_1, \ldots, x_n, y_n$ are not necessarily distinct variables such that $\{x_1, \ldots, x_n\} = \{y_1, \ldots, y_n\}$. The set of identities $S$ is a set of strongly shallow identities for $\Sigma_S$ if $S = \bigcup_{h \in \Sigma_S} S_h$, where $S_h$ is a non-empty set of strongly shallow $h$-identities for all $h \in \Sigma_S$. 

Sprin...
Example 1 Let $f$, $g$, $h$ respectively be a binary, an $n$-ary, and a ternary function symbol (for $n > 2$), and $a$ a constant symbol.

1. The identity $f(x, y) \approx f(y, x)$ stating that $f$ is commutative is a strongly shallow $f$-identity.
2. The identities $g(x_1, \ldots, x_i, x_{i+1}, \ldots, x_n) \approx g(x_1, \ldots, x_{i+1}, x_i, \ldots, x_n)$ for distinct variables $x_1, \ldots, x_n$ and $1 \leq i < n$ are strongly shallow $g$-identities. We call such an identity a $g$-transposition.
3. The identity $h(x, y, z) \approx h(x, y, z)$ is a strongly shallow $h$-identity.
4. The following identities are not strongly shallow according to our definition, though they are shallow according to the definition in [16]: $f(x, x) \approx x$, $f(x, y) \approx h(x, y, x)$, $f(x, y) \approx f(x, x)$, $f(x, a) \approx f(a, x)$.

The word problem for a finite set of strongly shallow identities (i.e., the relation $\approx_S$) can easily be shown to be decidable using rewriting. However, the classical notion of a canonical rewrite system, as introduced above, cannot be used for this purpose. The reason is that strongly shallow identities, like commutativity or transposition identities, usually cannot be oriented into terminating rewrite rules. However, we can orient their ground instances into terminating rules using an appropriate reduction order.

Given a finite set of constant symbols $C \supseteq C_0$, we denote the set of ground terms built using symbols from $\Sigma$ and $C$ with $G(\Sigma, C)$. Ground instances of terms are obtained by applying ground substitutions, which replace the variables in the given terms with elements of $G(\Sigma, C)$.

Definition 2 (Ordered rewrite system [17]) Given a finite set of (non-ground) identities $F$ and a reduction order $>$, we call the pair $(F, >)$ an ordered rewrite system. The ground rewrite system $R(F, >)$ induced by $(F, >)$ is defined as

$\{ \theta(s) \rightarrow \theta(t) \mid s \approx t \in F \cup F^{-1}, \theta \text{ is a ground substitution, and } \theta(s) > \theta(t) \}$,

where $F^{-1} = \{ t \approx s \mid s \approx t \in F \}$.

Since $>$ is assumed to be a reduction order, $R(F, >)$ is clearly terminating. The system $R(F, >)$ is usually infinite. However, if $F$ is finite and $>$ decidable, then one can effectively decide whether some rule in $R(F, >)$ applies to a given ground term, and compute a term resulting from the application of this rule. In particular, this implies that we can effectively compute a normal form of a given ground term w.r.t. the ground rewrite system $R(F, >)$. Note, however, that this normal form need not be unique since there is no guarantee that this system is confluent. We say that the ordered rewrite system $(F, >)$ is canonical on ground terms if $R(F, >)$ is confluent.

As a first example, we consider commutativity, i.e., the setting where $F_c$ consists of the commutativity axioms for the (binary) function symbols in $\Sigma_c \subseteq \Sigma$. Instead of $\Sigma_F$ and $F$ we use the symbols $\Sigma_c$ and $F_c$ to indicate already in the notation that we now consider commutativity. Let $>_\text{lpo}$ be the lexicographic path order\(^1\) induced by a linear order on $\Sigma \cup C$. Note that $>_\text{lpo}$ is then a linear order on $G(\Sigma, C)$ (see Exercise 5.20 in [5]). The ground rewrite system induced by $(F_c, >_\text{lpo})$ is of the form

$R(F_c, >_\text{lpo}) = \{ f(s, t) \rightarrow f(t, s) \mid f \in \Sigma_c, s, t \in G(\Sigma, C), \text{ and } s >_\text{lpo} t \}$,

and it is easy to see that it is confluent (this is an instance of Lemma 4 in [13], and also follows from what we show for strongly shallow identities below). Thus, $(F_c, >_\text{lpo})$ is canonical on ground terms.

---

\(^1\) See Sect. 5.4.2 in [5].
Now, assume $\Sigma_S \subseteq \Sigma$ and let $S$ be a set of strongly shallow identities for $\Sigma_S$. In general, the ground rewrite system $R(S, >_{lpo})$ is not confluent. For example, assume that $h \in \Sigma_S$ is ternary and $S_h = \{ h(x, y, z) \approx h(x, z, y), h(x, x, y) \approx h(x, y, y) \}$. If the linear order $>$ on $\Sigma \cup C$ that induces $>_{lpo}$ satisfies $a > b$, then the ground term $h(a, a, b)$ rewrites with $R(S, >_{lpo})$ both to $h(a, b, b)$ and $h(a, b, a)$. Since these terms are irreducible w.r.t. $R(S, >_{lpo})$, this shows that $R(S, >_{lpo})$ is not confluent.

To obtain an ordered rewrite system that is canonical on ground terms, we extend $S$ by all implied strongly shallow identities, i.e., we consider the set

$$
S^* = \{ (h(x_1, \ldots, x_n) \approx h(y_1, \ldots, y_n) \mid h \in \Sigma_S, h(x_1, \ldots, x_n) \approx S_h h(y_1, \ldots, y_n) \}.
$$

Finiteness of this set (up to renaming of variables) is an easy consequence of the fact that such identities can contain at most $n$ different variables. Thus, we can fix an arbitrary set $V_n$ of $n$ different variables and assume without loss of generality that the variables $x_1, y_1, \ldots, x_n, y_n$ belong to $V_n$. In addition, it is an easy consequence of the restricted form of strongly shallow identities that it is decidable whether an identity of this form follows from $S$: starting with $h(x_1, \ldots, x_n)$ one can simply enumerate the finitely many terms that can be obtained by iterated application of the identities in $S_h$.

Thus, $(S^*, >_{lpo})$ is a finite ordered rewrite system that can effectively be computed. Due to the finiteness of $S^*$ and decidability of $>_{lpo}$ (Proposition 5.4.16 in [5]), checking whether a rule of $R(S^*, >_{lpo})$ is applicable and actually applying it can effectively be done. The following lemma shows that $(S^*, >_{lpo})$ is canonical on ground terms.

**Lemma 1** The ground rewrite system $R(S^*, >_{lpo})$ is canonical and equivalent to $S$ on $G(\Sigma, C)$, i.e., $s \approx_S t$ iff $s \approx_{R(S^*, >_{lpo})} t$ holds for all terms $s, t \in G(\Sigma, C)$.

**Proof** Termination of $R(S^*, >_{lpo})$ is an immediate consequence of the fact that $>_{lpo}$ is a reduction order. Confluence is an easy consequence of the critical pair lemma for ordered rewrite systems in [17] (Lemma 2.1 and 2.2), but we prove it here explicitly for the sake of completeness.

Since $R(S^*, >_{lpo})$ is terminating, it is sufficient to show that all critical pairs of this ground rewrite system are joinable. There are two cases to consider: an overlap at the root or a rewrite within the substitution part. If there is an overlap at the root, then there are rules $h(\sigma(x_1), \ldots, \sigma(x_n)) \to h(\sigma(y_1), \ldots, \sigma(y_n))$ and $h(\theta(x'_1), \ldots, \theta(x'_n)) \to h(\theta(y'_1), \ldots, \theta(y'_n))$ in $R(S^*, >_{lpo})$ with identical left-hand sides, i.e., such that $h(\sigma(x_1), \ldots, \sigma(x_n)) = h(\theta(x'_1), \ldots, \theta(x'_n))$, which yields the critical pair:

$$
\langle h(\sigma(y_1), \ldots, \sigma(y_n)), h(\theta(y'_1), \ldots, \theta(y'_n)) \rangle.
$$

(1)

If we assume without loss of generality that the sets $\{ x_1, \ldots, x_n \} = \{ y_1, \ldots, y_n \}$ and $\{ x'_1, \ldots, x'_n \} = \{ y'_1, \ldots, y'_n \}$ are disjoint, then we can actually also assume that $\sigma = \theta$. Consequently, this substitution is a unifier of the terms $h(x_1, \ldots, x_n)$ and $h(x'_1, \ldots, x'_n)$, and thus is an instance of the most general unifier (mgu) $\delta$ of these two terms, i.e., there is a substitution $\lambda$ such that $\sigma = \lambda \delta = \theta$. Obviously, this mgu must map variables to variables, i.e., $\delta(x_i) = \delta(x'_i)$ is a variable for all $i = 1, \ldots, n$. But then $h(x_1, \ldots, x_n) \approx_S h(y_1, \ldots, y_n)$ and $h(x'_1, \ldots, x'_n) \approx_S h(y'_1, \ldots, y'_n)$ imply that $h(\delta(y_1), \ldots, \delta(y_n)) \approx_S h(\delta(y'_1), \ldots, \delta(y'_n))$. Since $>_{lpo}$ is total on ground terms, this implies that one of the following two rules belongs to $R(S^*, >_{lpo})$:

$$
h(\lambda(\delta(y_1)), \ldots, \lambda(\delta(y_n))) \to h(\lambda(\delta(y'_1)), \ldots, \lambda(\delta(y'_n))) \text{ or } h(\lambda(\delta(y'_1)), \ldots, \lambda(\delta(y'_n))) \to h(\lambda(\delta(y_1)), \ldots, \lambda(\delta(y_n))).
$$
Since $\sigma = \lambda.\delta = \theta$, this shows that the critical pair (1) is joinable.

A rewrite in the substitution part means that there is a rule of the form $h(\sigma(x_1), \ldots, \sigma(x_n)) \rightarrow h(\sigma(y_1), \ldots, \sigma(y_n))$ in $R(S^*, >_{\text{lpo}})$ and an $i, 1 \leq i \leq n$, such that $\sigma(x_i)$ can be rewritten to a term $r_i$ using a rule from $R(S^*, >_{\text{lpo}})$. The critical pair obtained this way is

$$(h(\sigma(x_1), \ldots, \sigma(x_{i-1}), r_i, \sigma(x_{i+1}), \ldots, \sigma(x_n)), h(\sigma(y_1), \ldots, \sigma(y_n))).$$

If we modify the substitution $\sigma$ to $\sigma'$ by setting $\sigma'(x_i) = r_i$, then both the left and the right component of the critical pair can be rewritten to the common term $h(\sigma'(y_1), \ldots, \sigma'(y_n))$ by $R(S^*, >_{\text{lpo}})$. This finishes the proof that $R(S^*, >_{\text{lpo}})$ is canonical.

By the definition of $R(S^*, >_{\text{lpo}})$, every rule $\ell \rightarrow r$ in $R(S^*, >_{\text{lpo}})$ satisfies $\ell \approx_S r$, which shows that $s \approx_{R(S^*, >_{\text{lpo}})} t$ implies $s \approx_S t$ for all terms $s, t \in G(\Sigma, C)$. The other direction is an easy consequence of the fact that $>_{\text{lpo}}$ is a linear order since this implies that every non-trivial substitution instance in $G(\Sigma, C)$ of a non-trivial identity in $S$ occurs (appropriately ordered) as a rule in $R(S^*, >_{\text{lpo}})$.

Coming back to the example illustrating that $R(S, >_{\text{lpo}})$ need not be confluent, we note that, in this example, the identity $h(x, y, y) \approx h(x, y, x)$ belongs to $S^*$, and thus the term $h(a, b, a)$ rewrites to $h(a, b, b)$ in $R(S^*, >_{\text{lpo}})$.

3 Semantic Congruence Closure Based on Rewriting

Let $\Sigma$ be a finite set of function symbols of arity $\geq 1$ and $C_0$ a finite set of constant symbols. In the following, let $E$ be a finite set of ground identities $s \approx t$ between terms $s, t \in G(\Sigma, C_0)$. In addition, assume that $\Sigma_S \subseteq \Sigma$ and let $S$ be a set of strongly shallow identities for $\Sigma_S$. Below, we show how to construct a canonical rewrite system that can be used to decide the word problem for $E$ w.r.t. $S$. This system will be the union of $R(S^*, >_{\text{lpo}})$ with a finite ground rewrite system $\bar{R}S(E)$, whose construction is described below.

3.1 Construction of the Rewrite System $\bar{R}S(E)$

Let $\text{Sub}(E)$ denote the set of subterms of the terms occurring in $E$. In a first step, we introduce a new constant $c_s$ for every term $s \in \text{Sub}(E) \setminus C_0$. To simplify notation, for a constant $a \in C_0$ we sometimes use $c_a$ to denote $a$. Let $C_1$ be the set of new constants introduced this way and $C := C_0 \cup C_1$. We fix an arbitrary linear order $>$ on $C$, which will be used to orient identities between constants into rewrite rules. Note that this order does not take into account which terms the constants correspond to, and thus we may well have $c_s > c_{f(s)}$.

The initial rewrite system $R(E)$ induced by $E$ consists of the following rules:

- If $s \in \text{Sub}(E) \setminus C_0$, then $s$ is of the form $f(s_1, \ldots, s_n)$ for an $n$-ary function symbol $f$ and terms $s_1, \ldots, s_n$ for some $n \geq 1$. For every such $s$ we add the rule $f(c_{s_1}, \ldots, c_{s_n}) \rightarrow c_s$ to $R(E)$.
- For every identity $s \approx t \in E$ we add $c_s \rightarrow c_t$ to $R(E)$ if $c_s > c_t$, and $c_t \rightarrow c_s$ if $c_t > c_s$.

Obviously, the cardinality of $C_1$ is linear in the size of $E$, and $R(E)$ can be constructed in time linear in the size of $E$. From the above construction, it follows that $R(E)$ has two types of rules: constant rules of the form $c \rightarrow d$ for $c > d$ and function rules of the form $f(c_1, \ldots, c_n) \rightarrow d$.

---

2 Trivial identities are of the form $s \approx s$. 

\textcircled{S} Springer
**Example 2** Consider $E = \{ f(g(a), a) \approx c, g(b) \approx h(a), a \approx b \}$ with $\Sigma_S = \{ f \}$ and $S_f = \{ f(x, y) \approx f(y, x) \}$. It is easy to see that $f(b, h(a)) \approx^S_c$.

Using our construction, we first introduce the constants $C_1 = \{ c_f(g(a), a), c_g(a), c_g(b), c_h(a) \}$. If we fix the linear order on $C$ as $c_f(g(a), a) > c_g(a) > c_g(b) > h(a) > a > b > c$, then we obtain the following rewrite system: $R(E) = \{ f(g(a), a) \rightarrow c_f(g(a), a), g(a) \rightarrow c_g(a), g(b) \rightarrow c_g(b), h(a) \rightarrow c_h(a), c_f(g(a), a) \rightarrow c, c_g(b) \rightarrow c_h(a), a \rightarrow b \}$.

The following lemma is an easy consequence of the definition of $R(E)$. It can be shown by a simple induction on the structure of the term $s$.

**Lemma 2** For all terms $s \in \text{Sub}(E)$ it holds that $s \approx^E_{R(E)} c_s$.

Using this lemma, we can show that the construction of $R(E)$ is correct for consequence w.r.t. strongly shallow identities in the sense stated in the next lemma. Recall that $E$ and $\approx^S_E$ are binary relations in $G(\Sigma, C_0)$, whereas $R(E)$ and $\approx^S_{R(E)}$ are binary relations in $G(\Sigma, C)$ for $C = C_0 \cup C_1$.

**Lemma 3** Viewed as a set of identities, $R(E)$ w.r.t. $S$ is a conservative extension of $E$ w.r.t. $S$, i.e., $s_0 \approx^S_{R(E)} t_0$ iff $s_0 \approx^S_{E} t_0$ for all terms $s_0, t_0 \in G(\Sigma, C_0)$.

**Proof** To show the if-direction, assume that $s_0 \not\approx^S_{E} t_0$, i.e., there is an algebra $A$ over the signature $\Sigma \cup C_0$ satisfying all identities in $E \cup S$ such that $s_0^A \neq t_0^A$. We expand $A$ to the new constants in $C_1$ by interpreting $c_s$ as $s^A$ for all $s \in \text{Sub}(E) \setminus C_0$, and call the expansion obtained this way $B$. Since $s_0^B = s_0^A \neq t_0^A = t_0^B$, it is sufficient to show that $B$ satisfies the identities in $R(E) \cup S$. Since $A$ satisfies $S$, its expansion obviously satisfies $S$ as well. If $f(c_{s_1}, \ldots, c_{s_n}) \rightarrow c_s$ belongs to $R(E)$, then $s = f(s_1, \ldots, s_n)$. Consequently, $f(c_{s_1}, \ldots, c_{s_n})^B = f^B(c_{s_1}^B, \ldots, c_{s_n}^B) = f^A(s_1^A, \ldots, s_n^A) = s^A = s^B$, which shows that $B$ satisfies the identity $f(c_{s_1}, \ldots, c_{s_n}) \approx c_s$. If $c_s \rightarrow c_t$ belongs to $R(E)$, then $s \approx t \in E$. Consequently, $c_s^B = s^A = t^A = c_t^B$, which shows that $B$ satisfies the identity $c_s \approx c_t$.

To show the only-if-direction, assume that $s_0 \not\approx^S_{R(E)} t_0$, i.e., there is an algebra $B$ over the signature $\Sigma \cup C$ satisfying all identities in $R(E) \cup S$ such that $s_0^B \neq t_0^B$. Let $A$ be the reduct of $B$ to the signature $\Sigma \cup C_0$, which is obtained from $B$ by forgetting about the interpretation of the symbols in $C_1$. Since $s_0^A = s_0^B \neq t_0^B = t_0^A$, it is sufficient to show that $A$ satisfies the identities in $E \cup S$. For the identities in $S$, this is again trivially true. Thus, consider an identity $s \approx t \in E$. Then $R(E)$ contains the rule $c_s \rightarrow c_t$ or $c_t \rightarrow c_s$. In addition, Lemma 2 yields $s \approx^S_{R(E)} c_s$ and $t \approx^S_{R(E)} c_t$. Consequently, $s^A = s^B = c_s^B = c_t^B = t^A$, since $B$ satisfies the identities in $in (R(E))$.

In Lemma 3, we consider the relation $\approx^S_{R(E)}$, where the elements of $S_h$ for $h \in \Sigma_S$ are used as additional identities. Our goal is, however, to deal both with the ground identities in $E$ and with the strongly shallow identities in $S$ by rewriting. This is where the ground rewrite system $R(S^*, >_{lpo})$ comes into play. We assumed that $>_{lpo}$ is induced by a linear order on $\Sigma \cup C$ that extends $>$ on $C$, makes each function symbol in $\Sigma$ greater than each constant symbol in $C$, and linearly orders the function symbols in an arbitrary way. This implies that $\ell >_{lpo} r$ for all rules $\ell \rightarrow r \in R(E)$. Consequently, termination of the rewrite system $R(E) \cup R(S^*, >_{lpo})$ can also easily be shown using $>_{lpo}$.

However, in general $R(E) \cup R(S^*, >_{lpo})$ need not be confluent due to possible interactions of rules in $R(E)$ with each other and of rules in $R(E)$ with rules in $R(S^*, >_{lpo})$. We turn $R(E) \cup R(S^*, >_{lpo})$ into a confluent and terminating system by modifying $R(E)$ appropriately.
Algorithm 1  We start with $R_0^S(E) := R(E)$ and $i := 0$:
(a) Let $R_i^S(E)|_{con}$ consist of the constant rules in $R_i^S(E)$. For every constant $c \in C$, consider
\[ [c]_l := \{ d \in C \mid c \approx_{R_i^S(E)|_{con}} d \}, \]
and let $e$ be the least element in $[c]_l$ w.r.t. the order $>$. We call $e$ the representative of $c$ w.r.t. $R_i^S(E)$ and $>$. If $c \neq e$, then add $c \rightarrow e$ to $R_i^S(E)$.
(b) In all function rules in $R_i^S(E)$, replace each constant by its representative w.r.t. $R_i^S(E)$ and $>$, and call the resulting set of function rules $F_i^S(E)$.

(b1) Let $h$ be an $n$-ary function symbol in $\Sigma_S$. For each function rule of the form $h(c_1, \ldots, c_n) \rightarrow d$ in $F_i^S(E)$, replace this rule with the rule $\ell \rightarrow d$, where $\ell$ is the canonical form of $h(c_1, \ldots, c_n)$ w.r.t. the canonical rewrite system $R(S^*, >_{lpo})$. Then continue with (b2).

(b2) Let $f$ be an $n$-ary function symbol in $\Sigma$. For every term $f(c_1, \ldots, c_n)$ occurring as the left-hand side of a rule in $F_i^S(E)$, consider all the rules $f(c_1, \ldots, c_n) \rightarrow d_1, \ldots, f(c_1, \ldots, c_n) \rightarrow d_k$ in $F_i^S(E)$ with this left-hand side. Let $d$ be the least element w.r.t. $>$ in $\{ d_1, \ldots, d_k \}$. Add $f(c_1, \ldots, c_n) \rightarrow d$ and $d_j \rightarrow d$ for all $j$ with $d_j \neq d$ to $R_i^S(E)$.

If at least one constant rule has been added in step (b), then set $i := i + 1$ and continue with step (a). Otherwise, terminate with output $\hat{R}^S(E) := R_{i+1}^S(E)$.

Let us illustrate the construction of $\hat{R}^S(E)$ using Example 2. In step (a), the non-trivial equivalence classes are $[a]_0 = \{ a, b \}$ with representative $b$, $[c f(g(a), a)] = \{ c f(g(a), a), c \}$ with representative $c$, and $[c g(b)] = \{ c g(b), c h(a) \}$ with representative $c h(a)$. Thus, $a \rightarrow b$, $c f(g(a), a) \rightarrow c g(b) \rightarrow c h(a)$ are the constant rules added to $R_1^S(E)$. The function rules in $F_0^S(E)$ are then $f(c g(a), b) \rightarrow c$, $g(b) \rightarrow c h(a)$, $h(a)$, $h(b)$, $c h(a)$, $h(b)$, $c h(a)$. Since $c g(a) > b$, the left-hand side of the rule $f(c g(a), b) \rightarrow c$ is reduced with $R(S^*, >_{lpo})$ to $f(b, c g(a))$, and thus in step (b1) this rule is replaced with $f(b, c g(a)) \rightarrow c$. In step (b2), there are two rules with left-hand side $g(b)$. For these rules, we add $c g(a) \rightarrow c h(a)$ and $g(b) \rightarrow c h(a)$ to $R_1^S(E)$. The rules with left-hand sides different from $g(b)$ are moved unchanged from $F_0^S(E)$ to $R_1^S(E)$ since their left-hand sides are unique. Thus, $R_1^S(E) = \{ a \rightarrow b, c f(g(a), a) \rightarrow c, c g(b) \rightarrow c h(a), c g(a) \rightarrow c h(a), f(b, c g(a)) \rightarrow c, g(b) \rightarrow c h(a), h(b) \rightarrow c h(a) \}$.

In the second iteration step, we now have the new non-trivial equivalence class $[c g(b)]_l = \{ c g(b), c h(a), c g(a) \}$ with representative $c h(a)$. The net effect of step (a) is, however, that the constant rules are moved unchanged from $R_1^S(E)$ to $R_2^S(E)$. The function rules in $F_1^S(E)$ are then $f(b, c h(a)) \rightarrow c$, $g(b) \rightarrow c h(a)$, $h(b) \rightarrow c h(a)$. Consequently, no reduction is possible in step (b1) and no constant rules are added in step (b2). The construction thus terminates with output $\hat{R}^S(E) = \{ a \rightarrow b, c f(g(a), a) \rightarrow c, c g(b) \rightarrow c h(a), c g(a) \rightarrow c h(a), f(b, c h(a)) \rightarrow c, g(b) \rightarrow c h(a), h(b) \rightarrow c h(a) \}$.

3.2 Proof of Correctness and Tractability

Our goal is now to show that $\hat{R}^S(E) \cup R(S^*, >_{lpo})$ provides us with a polynomial-time decision procedure for the word problem in $E$ w.r.t. $S$. When looking at the complexity of the word problem, one needs to clarify what counts as input and what is assumed to be fixed. In our setting, where we want to know whether $s_0 \approx^S_E t_0$ holds or not, one could on the one hand assume that $S$ and $E$ are fixed, and only the terms $s_0, t_0$ are part of the input. On the other hand, one could assume that the input consists of $S$, $E$, and $s_0, t_0$. In this paper,
we make the latter assumption, which makes our polynomiality result considerably stronger than if we had used the former one.

However, to obtain this result, we make the additional assumption that the maximal arity of the function symbols in $\Sigma_S$ is bounded by a constant. In this case we say that $\Sigma_S$ is arity-bounded. For the case of commutativity, this restriction is satisfied since then all the symbols in $\Sigma_S$ are binary. If we consider $g$-transpositions, but do not bound the arity $n$ of the symbol $g$ (i.e., the arities of the symbols satisfying transpositions is determined by the input rather than fixed from the outset), then the arity-boundedness condition is not satisfied.

If $S_g$ consists of all $g$-transitions, then the set of identities \({g(x_1, \ldots, x_n) \approx g(y_1, \ldots, y_n)}\) contains for distinct variables $x_1, \ldots, x_n$ all identities of the form $g(x_1, \ldots, x_n) \approx g(x_{\pi(1)}, \ldots, x_{\pi(n)})$, where $\pi$ is a permutation of the set $\{1, \ldots, n\}$, and is thus exponential in $n$. If $n$ is assumed to be a constant, then this set of identities is actually of constant size.

In the following, we assume for our complexity results that $\Sigma_S$ is arity bounded. Note that, even if the maximal arity of the symbols in $\Sigma_S$ is bounded in this way, how many symbols of a certain allowed arity are present in $\Sigma_S$ and which strongly shallow identities they satisfy, can still be determined by the input. First, we prove that, under this assumption, the construction of $\tilde{R}^S(E)$ takes only polynomial time.

**Lemma 4** If $\Sigma_S$ is arity-bounded, then the system $\tilde{R}^S(E)$ can be computed from $R(E)$ in polynomial time.

**Proof** First, note that step (a) can clearly be performed in polynomial time since deciding $\approx_{R_i^S(E)}$ amounts to performing reachability tests in an undirected graph. Producing the original rules in $F_i^S(E)$ can then clearly also be done in polynomial time.

The canonical forms of left-hand sides of the form $h(c_1, \ldots, c_n)$ for $h \in \Sigma_S$ required in step (b1) can also be computed in polynomial time. In fact, the terms obtained by a rewrite sequence in $R(S^{*}, >_{lpo})$ that starts with $h(c_1, \ldots, c_n)$ are all of the form $h(d_1, \ldots, d_n)$ with $\{d_1, \ldots, d_n\} = \{c_1, \ldots, c_n\}$. The number of such terms is bounded by $n^n$ (which is a constant) and since $R(S^{*}, >_{lpo})$ is terminating, there cannot be a repetition in this sequence. Consequently, at most $n^n$ (i.e., constantly many) rewrite steps are applied.

Grouping the rules in $F_i^S(E)$ according to their left-hand sides in step (b2) is clearly also possible in polynomial time, as is adding the new rules to $R_{i+1}^S(E)$. In case the procedure does not terminate after step (b), the number of different equivalence classes of constants decreases by at least one. Thus, the iteration must terminate after at most $|C|$ steps.

Next, we show that the construction of $\tilde{R}^S(E)$ is correct in the following sense.

**Lemma 5** Viewed as a set of identities, $\tilde{R}^S(E) \cup R(S^{*}, >_{lpo})$ is equivalent to $R(E)$ w.r.t. $S$, i.e., $\approx_{R_E} \approx_{\tilde{R}^S(E) \cup R(S^{*}, >_{lpo})}$ for all terms $s, t \in G(\Sigma, C)$.

**Proof** Note that (on $G(\Sigma, C)$) we have $\approx_{R_E} = \approx_{R_{0}^S(E)} = \approx_{R^S(E) \cup R(S^{*}, >_{lpo})}$ since $R(E) = R_{0}^S(E)$ and $R(S^{*}, >_{lpo})$ is equivalent to $S$ on $G(\Sigma, C)$ by Lemma 1. It is thus sufficient to show that the modifications performed when going from $R_{i}^S(E)$ to $R_{i+1}^S(E)$ do not change the induced equational theory, i.e., $\approx_{R_{i}^S(E) \cup R(S^{*}, >_{lpo})} = \approx_{R_{i+1}^S(E) \cup R(S^{*}, >_{lpo})}$ for all $i \geq 0$ for which $R_{i+1}^S(E)$ is defined.

---

3 More efficiently, one can maintain the equivalence classes $[c]_i$ and the representatives w.r.t. $R_{i}^S(E)$ and $>$ by building and updating a union-find data structure, as e.g. described in Sect. 4.4 of [5].
To show the inclusion from left to right, first consider step (a). If \( R^S_i(E) \) contains the constant rule \( c_1 \rightarrow c_2 \), then \( c_1 \) and \( c_2 \) belong to the same equivalence class w.r.t. \( R^S_i(E) |_{\text{con}} \) and \( c_1 \succ c_2 \). In case \( c_2 \) is the least element in this class, then \( R^S_{i+1}(E) \) still contains the rule \( c_1 \rightarrow c_2 \). Otherwise, since we know that \( c_1 \succ c_2 \), the least element \( e \) in the class is different from these two constants, and thus \( R^S_{i+1}(E) \) contains the rules \( c_1 \rightarrow e, c_2 \rightarrow e \), which shows \( c_1 \approx_{R^S_{i+1}(E)} c_2 \), and thus \( c_1 \approx_{R^S_{i+1}(E) \cup R(S^*, >_{\text{ipo}})} c_2 \).

Regarding step (b), note that the replacements performed in the construction of \( F^S_i(E) \) replace constants \( c \) occurring in function rules by constants \( e \) that are equivalent to \( c \) both w.r.t. \( R^S_i(E) \) and w.r.t. \( R^S_{i+1}(E) \). Thus, these replacements do not change the overall equational theory. In step (b1), rules of the form \( h(c_1, \ldots, c_n) \rightarrow d \) are replaced with rules \( \ell \rightarrow d \) such that \( h(c_1, \ldots, c_n) \approx_{R(S^*, >_{\text{ipo}})} \ell \), which thus also does not change the overall equational theory.

For step (b2), consider a function rule \( f(c_1, \ldots, c_n) \rightarrow d_j \) in \( F^S_i(E) \). Either this rule also belongs to \( R^S_{i+1}(E) \), or \( R^S_{i+1}(E) \) contains rules \( f(c_1, \ldots, c_n) \rightarrow d, d_j \rightarrow d \). In both cases, \( f(c_1, \ldots, c_n) \approx_{R^S_{i+1}(E)} d_j \), and thus \( f(c_1, \ldots, c_n) \approx_{R^S_{i+1}(E) \cup R(S^*, >_{\text{ipo}})} d_j \) is preserved.

The inclusion from right to left can be shown similarly to the one from left to right. \( \square \)

**Lemma 6** Viewed as a term rewriting system, \( \tilde{R}^S(E) \cup R(S^*, >_{\text{ipo}}) \) is canonical, i.e., terminating and confluent.

**Proof** Termination of the term rewriting system \( \tilde{R}^S(E) \cup R(S^*, >_{\text{ipo}}) \) can be shown as for \( R(E) \cup R(S^*, >_{\text{ipo}}) \), by using the reduction order \( >_{\text{ipo}} \).

Regarding confluence, we first claim that there are no non-trivial critical pairs (see Sect. 6.2 in [5]) between the rules in \( \tilde{R}^S(E) \). To see this, note that two function rules from \( \tilde{R}^S(E) \) cannot overlap due to the fact that in step (b2) no more constants are identified, and thus all left-hand sides of function rules in \( \tilde{R}^S(E) \) are unique. In addition, any constant can occur as left-hand side of at most one rule due to step (a). A rule of the form \( f(c_1, \ldots, c_n) \rightarrow d \) cannot overlap with a rule of the form \( e \rightarrow e' \) since the \( c_i \) are representatives, whereas \( e \) is not a representative.

Second, we have already shown in the proof of Lemma 1 that all critical pairs between rules in \( R(S^*, >_{\text{ipo}}) \) are joinable.

Thus, consider an overlap of a rule \( h(s_1, \ldots, s_n) \rightarrow r \) in \( R(S^*, >_{\text{ipo}}) \) with a rule in \( \tilde{R}^S(E) \). If this overlap occurs at the root position, then the rule in \( \tilde{R}^S(E) \) is a function rule of the form \( h(c_1, \ldots, c_n) \rightarrow d \) in \( \tilde{R}^S(E) \). However, due to step (b1), the left-hand sides of such function rules are irreducible w.r.t. \( R(S^*, >_{\text{ipo}}) \), which contradicts our assumption that \( h(s_1, \ldots, s_n) = h(c_1, \ldots, c_n) \) is the left-hand side of a rule in \( R(S^*, >_{\text{ipo}}) \).

Finally, assume that the rule from \( \tilde{R}^S(E) \) is applied below the root position in the left-hand side of the rule in \( R(S^*, >_{\text{ipo}}) \). Joinability of the critical pair obtained this way can be shown as in the case of a rewrite in the substitution part in the proof of Lemma 1.

Summing up, we have shown that all non-trivial critical pairs of \( \tilde{R}^S(E) \cup R(S^*, >_{\text{ipo}}) \) can be joined, which proves confluence of \( \tilde{R}^S(E) \cup R(S^*, >_{\text{ipo}}) \). \( \square \)

Since \( \tilde{R}^S(E) \cup R(S^*, >_{\text{ipo}}) \) is canonical, each term \( s \in G(\Sigma, C) \) has a unique canonical form (i.e., irreducible term reachable from \( s \)) w.r.t. \( \tilde{R}^S(E) \cup R(S^*, >_{\text{ipo}}) \). We can thus use the system \( \tilde{R}^S(E) \cup R(S^*, >_{\text{ipo}}) \) to decide whether terms \( s_0, t_0 \) are equivalent in \( E \) w.r.t. \( S \), i.e., whether \( s_0 \approx t_0 \in CC^S(E) \), by computing the canonical forms of the terms \( s_0 \) and \( t_0 \).
Theorem 1 If \( s_0, t_0 \in G(\Sigma, C_0) \), then \( s_0 \approx t_0 \in CC^S(E) \) iff \( s_0 \) and \( t_0 \) have the same canonical form w.r.t. \( \hat{R}^S(E) \cup R(S^*, >_{lpo}) \).

Proof If \( s_0 \approx t_0 \in CC^S(E) \), then \( s_0 \approx^S_{E} t_0 \), and thus \( s_0 \approx^S_{\hat{R}^S(E) \cup R(S^*, >_{lpo})} t_0 \) by Lemma 3 and Lemma 5. Since \( \hat{R}^S(E) \cup R(S^*, >_{lpo}) \) is canonical, this implies that \( s_0 \) and \( t_0 \) have the same canonical form w.r.t. \( \hat{R}^S(E) \cup R(S^*, >_{lpo}) \).

Conversely, if \( s_0 \) and \( t_0 \) have the same canonical form w.r.t. \( \hat{R}^S(E) \cup R(S^*, >_{lpo}) \), then \( s_0 \approx^S_{\hat{R}^S(E) \cup R(S^*, >_{lpo})} t_0 \). Lemma 5 yields \( s_0 \approx^S_{R(E)} t_0 \). Since \( s_0, t_0 \in G(\Sigma, C_0) \), we can now apply Lemma 3 to obtain \( s_0 \approx^S_{E} t_0 \), which is equivalent to \( s_0 \approx t_0 \in CC^S(E) \). \( \square \)

As an example, consider the rewrite system \( \hat{R}^S(E) \) that we have computed (above Lemma 4) from the set of ground identities \( E \) in Example 2, and recall that \( f(b, h(a)) \approx^S_E c \). The canonical form of \( c \) is clearly \( c \), and the canonical form of \( f(b, h(a)) \) can be computed by the following rewrite sequence: \( f(b, h(a)) \rightarrow^* \hat{R}^S(E) f(b, h(b)) \rightarrow^* \hat{R}^S(E) f(b, c h(a)) \rightarrow^* \hat{R}^S(E) c \). In this example, no rule of \( R(S^*, >_{lpo}) \) is used in the computation of the canonical forms, which is of course not the case in general. Also note that we have used \( R(S^*, >_{lpo}) \) in the construction of \( \hat{R}^S(E) \).

The construction of \( \hat{R}^S(E) \) is actually independent of the terms \( s_0, t_0 \) for which we want to decide the word problem in \( E \) w.r.t. \( S \). This is in contrast to approaches that restrict the construction of the congruence closure to the subterms of \( E \) and the subterms of the terms \( s_0, t_0 \) for which one wants to decide the word problem. This fact will turn out to be useful in the next section.

It remains to show that the decision procedure obtained by applying Theorem 1 requires only polynomial time if \( \Sigma_S \) is assumed to be arity-bounded.

Corollary 1 If \( \Sigma_S \) is arity-bounded, then the word problem for finite sets of ground identities w.r.t. \( S \) is decidable in polynomial time, i.e., given a finite set of ground identities \( E \subseteq G(\Sigma, C_0) \times G(\Sigma, C_0) \) and terms \( s_0, t_0 \in G(\Sigma, C_0) \), we can decide in polynomial time, whether \( s_0 \approx^S_{E} t_0 \) holds or not if we assume that the maximal arity of the symbols in \( \Sigma_S \) is bounded by a constant.

Proof Since we already know that \( \hat{R}^S(E) \) can be constructed in time polynomial in the size of \( E \), and thus also has polynomial size, it is sufficient to show that the canonical form of a term \( s_0 \in G(\Sigma, C_0) \) w.r.t. the rewrite system \( \hat{R}^S(E) \cup R(S^*, >_{lpo}) \) can be computed in time polynomial in the size of \( s_0 \) and \( \hat{R}^S(E) \). This is clearly the case if only a polynomial number of rewrite steps are needed to produce the canonical form of \( s_0 \), and each application of a rewrite step also takes only polynomial time. The latter is an easy consequence of the facts that the relevant part of the set of identities

\[
S^* = \{ h(x_1, \ldots, x_n) \approx h(y_1, \ldots, y_n) \mid h \in \Sigma_S, h(x_1, \ldots, x_n) \approx_{S_0} h(y_1, \ldots, y_n) \}
\]

is of linear size if \( n \) is assumed to be a constant and \( >_{lpo} \) is decidable in polynomial time (see Proposition 5.4.16 in [5]).

We show the former by induction on the structure of \( s_0 \). If \( s_0 \) is a constant, then there are at most \( |C| - 1 \) rewrite steps starting with \( s_0 \) possible. This yields the desired polynomial bound since the cardinality of \( C \) is linear in the size of \( E \).

Now assume that \( s_0 = h(s_1, \ldots, s_n) \) for a function symbol \( h \in \Sigma \). First, we compute the canonical forms \( t_1, \ldots, t_n \) of \( s_1, \ldots, s_n \). By induction, we can assume that the number of rewrite steps required for these computations is bounded by polynomials \( p_1, \ldots, p_n \) in the sizes of \( E \) and \( s_1, \ldots, s_n \), respectively. If \( h \in \Sigma_S \), then we rewrite \( h(t_1, \ldots, t_n) \) with
the rules in $R(S^*, >_{lpo})$. Note that the terms obtained by this rewrite sequence are all of
the form $h(u_1, \ldots, u_n)$ with $\{t_1, \ldots, t_n\} = \{u_1, \ldots, u_n\}$. Since the number of such terms
is bounded by $n^n$ (which is a constant) and $R(S^*, >_{lpo})$ is terminating, there cannot be a
repetition in this sequence. Consequently, at most $n^n$ rewrite steps are applied. If the term
$h(v_1, \ldots, v_n)$ obtained by this rewrite sequence is the left-hand side of a rule in $\tilde{R}^c(E)$, then
it rewrites to a constant $d$, from which at most $|C| - 1$ rewrite steps are possible. Overall,
we thus need at most $p_1 + \ldots + p_n + n^n + |C|$ rewrite steps to compute the canonical form
of $s_0 = h(s_1, \ldots, s_n)$, which is clearly a polynomial bound in the size of $s_0$ and $E$.

In the case where $h \in \Sigma \setminus \Sigma_S$, the bound is $p_1 + \ldots + p_n + |C|$ instead. ☐

Commutative function symbols all have arity 2, and thus $\Sigma_S = \Sigma_c$ is clearly arity-
bounded. Given $\Sigma_c$, we denote the set of commutativity axioms for the elements of $\Sigma_c$ as $F_c$.

**Corollary 2** The commutative word problem for finite sets of ground identities is decidable
in polynomial time, i.e., given a finite set of ground identities $E \subseteq G(\Sigma, C_0) \times G(\Sigma, C_0)$,
commutativity axioms $F_c$ for the symbols in $\Sigma_c \subseteq \Sigma$, and terms $s_0, t_0 \in G(\Sigma, C_0)$, we can
decide in polynomial time whether $s_0 \approx_{E}^{F_c} t_0$ holds or not.

Even if $\Sigma_S$ is not arity-bounded, we may still obtain polynomially decidable word
problems. As an example, consider for all $n > 2$ function symbols $g_n$ that are assumed to satisfy
the transposition identities

$$S_{g_n} = \{g_n(x_1, \ldots, x_i, x_{i+1}, \ldots, x_n) \approx g_n(x_1, \ldots, x_{i+1}, x_i, \ldots, x_n) \mid 1 \leq i \leq n - 1\}.$$

In this case, it is not necessary to consider the (exponentially many) derived identities between
strongly shallow $g_n$-terms in $S^*_{g_n}$ to construct the ordered rewrite system. Using $(S_{g_n}, >_{lpo})$
instead is sufficient since this system is canonical on ground terms. In fact, using the rules in
$R(S_{g_n}, >_{lpo})$ one can compute the canonical form of a ground term in polynomial time since
this corresponds to applying bubble-sort. This observation can be used to show that the word
problem for finite sets of ground identities w.r.t. function symbols satisfying transposition
identities is decidable in polynomial time even without assuming arity-boundedness.

## 4 Semantic Congruence Closure with Extensionality

Here, we additionally assume that some of the function symbols are extensional, i.e., there is
a set of function symbols $\Sigma^e \subseteq \Sigma$ whose elements we call extensional symbols. An
extensional symbol $f \in \Sigma^e$ must satisfy the conditional identities

$$f(x_1, \ldots, x_n) \approx f(y_1, \ldots, y_n) \Rightarrow x_i \approx y_i \text{ for all } i, 1 \leq i \leq n. \quad (2)$$

In the present section, we assume that extensional symbols do not satisfy any other
semantic properties. The reason for this restriction will be explained in the next section.

We have shown in [13] how to integrate extensional symbols into the rewriting-based
commutative congruence closure algorithm developed there. It is not hard to see that this
approach can be adapted to the integration of extensionality into congruence closure w.r.t.
strongly shallow identities as developed in the previous section. Below, we describe this
integration in the more general setting of a generic rewriting-based semantic congruence
closure. In addition to making our result more general, this also has the advantage that it
clarifies which properties of our rewriting-based congruence closure algorithm are vital for integrating extensional symbols.

### 4.1 Generic Semantic Congruence Closure

Abstracting from the specific form of the semantic properties, we now consider a setting where there is a subset $\Sigma_F$ of $\Sigma$ and a set of identities $F$ formulated using the function symbols in $\Sigma_F$ and variables (see Sect. 2.2.2). We assume that one can construct a canonical ground rewrite system $\hat{R}^F$ that satisfies properties similar to the system $\hat{R}^S(E) \cup R(S^*, >_{lpo})$ considered in the previous section.

We call a ground rewrite system $R \subseteq G(\Sigma, C) \times G(\Sigma, C)$ flat if it consists of constant rules of the form $c \rightarrow d$ and function rules of the form $f(c_1, \ldots, c_n) \rightarrow d$, where $c, c_1, \ldots, c_n, d \in C$ and $f \in \Sigma$ is an $n$-ary function symbol.

**Definition 3** We say that rewrite-based congruence closure is effective for $\Sigma_F$ and $F$ if, for every finite flat ground rewrite system $R \subseteq G(\Sigma, C) \times G(\Sigma, C)$, there is a (possibly infinite) ground rewrite system $\hat{R}^F \subseteq G(\Sigma, C) \times G(\Sigma, C)$ that satisfies the following properties:

1. All rules containing a symbol in $\Sigma_F$ have left-hand sides whose root symbols belong to $\Sigma_F$,
2. All rules containing a symbol $f \in \Sigma \setminus \Sigma_F$ as root symbol of the left-hand side are flat function rules of the form $f(c_1, \ldots, c_n) \rightarrow d$, and the subset of these rules can effectively be computed,
3. $\hat{R}^F$ is equivalent to $R$ w.r.t. $F$ on terms in $G(\Sigma, C)$,
4. $\hat{R}^F$ is canonical and canonical forms of terms in $G(\Sigma, C)$ w.r.t. this term rewriting system can effectively be computed.

Note that we do not require that the canonical ground rewrite system $\hat{R}^F$ itself is computable, but only that there is an algorithm for computing canonical forms w.r.t. this system. The reason is that $\hat{R}^F$ may actually be infinite, and thus may not be computable in finite time. One may wonder how one can get a procedure for computing canonical forms w.r.t. $\hat{R}^F$ if this system is infinite. One possibility is that one can compute a finite representation of this system that allows one to produce these normal forms. In the case of strongly shallow identities, this finite representation consists of the finite ground rewrite system $\hat{R}^S(E)$ together with the ordered rewrite system $(S^*, >_{lpo})$. It is easy to see that the system $\hat{R}^S := \hat{R}^S(E) \cup R(S^*, >_{lpo})$ induced by these two components satisfies the conditions required by Definition 3.

Under the assumptions formulated in this definition, given two terms $s_0, t_0$ in $G(\Sigma, C_0)$, we can decide whether $s_0 \approx^F_R t_0$ holds or not by computing their canonical forms w.r.t. $\hat{R}^F$ and checking whether they are syntactically equal. In order to obtain a polynomial-time decision procedure, it is sufficient to assume that all the computations mentioned above can be performed in polynomial time. In this case we say that rewrite-based congruence closure is polynomial for $\Sigma_F$ and $F$. If $S$ is a strongly shallow set of identities over an arity-bounded signature $\Sigma_S$, then $\hat{R}^S = \hat{R}^S(E) \cup R(S^*, >_{lpo})$ satisfies the conditions required for a polynomial rewrite-based congruence closure. However, as mentioned at the end of Sect. 3, polynomiality may be achieved even if $\Sigma_S$ is not arity-bounded.
4.2 Generic Semantic Congruence Closure with Extensionality

Let $\Sigma_F$ and $F$ be as above, and assume that $\Sigma^e \subseteq \Sigma \setminus \Sigma_F$. In addition to the identities in $E$ and the identities in $F$ for the symbols in $\Sigma_F$, we now assume that also the conditional identities in (2) are satisfied for the symbols $f \in \Sigma^e$. From a semantic point of view, this means that we now consider algebras $\mathcal{A}$ that satisfy not only the identities in $E \cup F$, but also extensionality for the symbols in $\Sigma^e$, i.e., for all $f \in \Sigma^e$, all $i$, $1 \leq i \leq n$, and all elements $a_1, \ldots, a_n, b_1, \ldots, b_n$ of $\mathcal{A}$ it holds that $f^\mathcal{A}(a_1, \ldots, a_n) = f^\mathcal{A}(b_1, \ldots, b_n)$ implies $a_i = b_i$ for all $i$, $1 \leq i \leq n$. We say that $s \approx t$ follows from $E$ w.r.t. $F$ and the extensional symbols in $\Sigma^e$ (written $s \approx_{E, \Sigma^e} t$) if $s^\mathcal{A} = t^\mathcal{A}$ holds in all algebras $\mathcal{A}$ that satisfy the identities in $E \cup F$ and extensionality for the symbols in $\Sigma^e$.

The relation $\approx_{E, \Sigma^e}$ is the smallest subset of $G(\Sigma, \mathcal{C}_0) \times G(\Sigma, \mathcal{C}_0)$ that contains $E$ and is closed under reflexivity, transitivity, symmetry, congruence, $F$-identities, and the following extensionality rule:

- if $f \in \Sigma^e$ is an $n$-ary function symbol, $1 \leq i \leq n$, and $f(s_1, \ldots, s_n) \approx f(t_1, \ldots, t_n) \in CC^F, \Sigma^e(\mathcal{E})$, then $s_i \approx t_i \in CC^F, \Sigma^e(\mathcal{E})$ (extensionality).

**Proposition 1** For all terms $s, t \in G(\Sigma, \mathcal{C}_0)$ it holds that $s \approx_{E, \Sigma^e} t$ iff $s \approx t \in CC^F, \Sigma^e(\mathcal{E})$.

**Proof** This proposition is an easy consequence of Theorem 54 in [19]. Adapted to our setting, this theorem says that $\approx_{E, \Sigma^e}$ is the least congruence containing $E$ that is invariant under applying $F$-identities and extensionality. Clearly, this is exactly $CC^F, \Sigma^e(\mathcal{E})$. $\square$

To obtain a decision procedure for $\approx_{E, \Sigma^e}$, we assume that rewrite-based congruence closure is effective for $\Sigma_F$ and $F$. To apply this assumption, we first need to transform $E$ into a flat rewrite system. Thus, let the flat term rewriting system $R(E)$ be defined as in Sect. 3.

**Example 3** Consider $E' = \{f(g(a), a) \approx c, g(b) \approx h(a), g(a) \approx g(b)\}$ with $\Sigma_F = \{f\}$ where $F = \{f(x, y) \approx f(y, x)\}$ and $\Sigma^e = \{g\}$. Then $f(b, h(a)) \approx_{E', \Sigma^e} c$.

Let the set $C_1$ of new constants and the linear order on all constants be defined as in Example 2. Then we obtain the following flat system:

$$R(E') = \{f(c_{f(g(a), a)}, a) \rightarrow c_{f(g(a), a)}, g(a) \rightarrow c_{g(b)}, g(b) \rightarrow c_{g(b)}, h(a) \rightarrow c_{h(a)}, c_{f(g(a), a)} \rightarrow c, c_{g(b)} \rightarrow c_{h(a)}, c_{g(b)} \rightarrow c_{g(b)}\}.$$

**Lemma 7** The system $R(E)$ is a conservative extension of $E$ also w.r.t. $F$ and the extensional symbols in $\Sigma^e$, i.e., $s_0 \approx_{E, \Sigma^e} t_0$ iff $s_0 \approx_{R(E), \Sigma^e} t_0$ for all terms $s_0, t_0 \in G(\Sigma, \mathcal{C}_0)$.

**Proof** The proof of this lemma is basically identical to the proof of Lemma 3. One only needs to note, additionally, that going to the expansion and the reduct constructed in this proof does not change satisfaction of the extensionality axioms. $\square$

The confluent and terminating rewrite system for deciding $\approx_{E, \Sigma^e}$ is now constructed by first applying the generic semantic congruence closure to $R(E)$, then applying extensionality to derive new identities between constants, and iterating these two steps until no new identities are derived in the second step.
Algorithm 2  The rewrite system $\hat{R}^F,\Sigma^e(\hat{E})$ is constructed by performing the following steps, starting with $R_0 := R(\hat{E})$ and $i := 0$:

(g) Apply the generic semantic congruence closure for $\Sigma_F$ and $F$ to $R_i$ to obtain the canonical rewrite system $\hat{R}_i^F$.

(e) Let $R_{i+1} := R_i$. For all $f \in \Sigma^e$, all pairs of distinct flat function rules $f(c_1, \ldots, c_n) \rightarrow d, f(c'_1, \ldots, c'_n) \rightarrow d$ in $\hat{R}_i^F$, and all $j, 1 \leq j \leq n$, such that $c_j \neq c'_j$, add $c_j \rightarrow c'_j$ to $R_{i+1}$ if $c_j \succ c'_j$ and otherwise add $c'_j \rightarrow c_j$ to $R_{i+1}$. If at least one constant rule has been added in this step, then set $i := i + 1$ and continue with step (g). Otherwise, terminate with output $\hat{R}^F,\Sigma^e(E) := \hat{R}_i^F$.

Lemma 8  The construction of $\hat{R}^F,\Sigma^e(E)$ terminates after a linear number of iterations and produces a canonical rewriting system.

Proof  Termination after linearly many iterations is an immediate consequence of the fact that, in each iteration of the construction (except the last one), the number of distinct equivalence classes of constants decreases, and the cardinality of the set of constants $C$ is linear in the size of the input. The system $\hat{R}^F,\Sigma^e(E)$ is canonical since it is produced by an application of rewrite-based congruence closure to a finite flat ground rewrite system $R_i$. □

We illustrate the construction using Example 3. In step (g), we apply Algorithm 1 from the previous section. In step (a) of that algorithm, the non-trivial equivalence classes of constants are $[c_f(g(a),a)] = \{c_f(g(a),a), c\}$ with representative $c$ and $[c_g(b)] = \{c_g(b), c_h(a)\}$ with representative $c_h(a)$. Thus, $c_f(g(a),a) \rightarrow c, c_g(b) \rightarrow c_h(a)$, are the constant rules added. The function rules are then $f(c г(a), a) \rightarrow c, g(a) \rightarrow c_h(a), g(b) \rightarrow c_h(a), h(a) \rightarrow c_h(a)$. In step (b1), the first rule is rewritten to $f(a, c_h(a)) \rightarrow c$. Since the function rules obtained this way have unique left-hand sides, no constant rule is added in step (b2). Thus, the construction in step (g) terminates and yields the system that is the union of the generated flat rules with the ones in $R(F, >_{\text{lp}})$ (with the strongly shallow identities $F$ being commutativity of the symbols in $\Sigma_F$). Note that, for commutativity, $F^* = F$.

Consequently, we now proceed with step (e). Since $g \in \Sigma^e$, the presence of the rules $g(a) \rightarrow c_h(a)$ and $g(b) \rightarrow c_h(a)$ triggers the addition of $a \rightarrow b$ to $R_1$.

In the second iteration step, we now have in step (a) of the construction applied in (g) the additional non-trivial equivalence class $[a] = \{a, b\}$ with representative $b$. The net effect of step (a) is then to generate the constant rules $c_f(g(a),a) \rightarrow c, c_g(b) \rightarrow c_h(a)$, and $a \rightarrow b$. The obtained function rules after step (b1) are then $f(b, c_h(a)) \rightarrow c, g(b) \rightarrow c_h(a), h(b) \rightarrow c_h(a)$. No new constant rules are added in step (b2). In step (e), the construction terminates with the set of flat rules $R_1 = \{a \rightarrow b, c_f(g(a),a) \rightarrow c, c_g(b) \rightarrow c_h(a), f(b, c_h(a)) \rightarrow c, g(b) \rightarrow c_h(a), h(b) \rightarrow c_h(a)\}$, which is identical to the system $\hat{R}^F(E)$ computed for the set of identity $E$ of Example 2. The system $\hat{R}^F,\Sigma^e(E)$ produced by this run of Algorithm 2 is $R_1 \cup R(F, >_{\text{lp}})$.

Our goal is now to show, for the general case, that $\hat{R}^F,\Sigma^e(E)$ provides us with a decision procedure for the extensional word problem for $E \cup F$, i.e., it allows us to decide the relation $\approx^F,\Sigma^e_E$. This is an easy consequence of the following theorem.

Theorem 2  If $s_0, t_0 \in G(\Sigma, C_0)$, then $s_0 \approx t_0 \in CC^F,\Sigma^e(E)$ iff $s_0$ and $t_0$ have the same canonical form w.r.t. $\hat{R}^F,\Sigma^e(E)$.

Proof  Assume that Algorithm 2 has terminated with output $\hat{R}^F,\Sigma^e(E) = \hat{R}_n^F$.

To show the if-direction, assume that $s_0$ and $t_0$ have the same canonical form w.r.t. $\hat{R}^F,\Sigma^e(E)$. Then, $s_0 \approx t_0 \in R_n$, and thus $s_0 \approx t_0$ in $\hat{R}_n^F$ is equivalent to $R_n$ w.r.t. $F$. Springer
Consequently, it is sufficient to prove that \( \cong_{R_{i+1}} \subseteq \cong_{R_i} \) holds for all \( i, 0 \leq i < n \). In fact, then \( s_0 \cong_{R_0} t_0 \) implies \( s_0 \cong_{R_i} t_0 \), which yields \( s_0 \cong_{E} t_0 \) by Lemma 7 since \( R_0 = R(E) \).

To see the inclusion \( \cong_{R_{i+1}} \subseteq \cong_{R_i} \), note that the presence of the rules \( f(c_1, \ldots, c_n) \rightarrow d, f(c'_1, \ldots, c'_n) \rightarrow d \) in \( \hat{R}_i \) implies that \( f(c_1, \ldots, c_n) \cong_{R_i} f(c'_1, \ldots, c'_n) \), and thus \( c_j \cong_{R_i} c'_j \). Since these are the only identities added when going from \( R_i \) to \( R_{i+1} \), this completes the proof of the if-direction.

We show the only-if-direction by contraposition. Thus, assume that \( s_0, t_0 \) have different canonical forms w.r.t. \( \hat{R}^{F}.\Sigma^e(E) \). Let \( A \) be the initial algebra (i.e., the free algebra over the empty set of generators) for \( \hat{R}^{F}.\Sigma^e(E) \) viewed as a set of identities over the signature \( \Sigma \cup C \). Recall that this algebra has the equivalence classes of terms in \( G(\Sigma, C) \) w.r.t. \( \cong_{\hat{R}^{F}.\Sigma^e(E)} \) as its elements, and any term \( s \in G(\Sigma, C) \) is interpreted in \( A \) as the class of \( s \). Since \( \hat{R}^{F}.\Sigma^e(E) \) is canonical, we can represent such a class by the unique canonical form of its elements. Obviously, the fact that \( s_0, t_0 \) have different canonical forms w.r.t. \( \hat{R}^{F}.\Sigma^e(E) \) implies \( s_0^A \neq t_0^A \). Thus, if we can show that \( A \) satisfies \( E \), the identities in \( F \), and extensionality for every \( f \in \Sigma^e \), then \( s_0^A \neq t_0^A \) implies that \( s_0 \cong_{E} t_0 \), and thus \( s_0 \cong_{R(E)} t_0 \).

If \( s \cong t \in E \), then \( s \cong_{R(E)} t \) (see the proof of Lemma 3). Since \( R(E) = R_0 \subseteq \cdots \subseteq R_n \), this implies \( s \cong_{R_i} t \), and thus \( s \cong_{R_i} t \). Consequently, these two terms are evaluated to the same element of \( A \), which shows that \( A \) satisfies the identities in \( E \).

Let \( u \approx v \in F \) and \( \theta(u), \theta(v) \) be ground instances of \( u, v \). Then we know that \( \theta(u) \approx_{R(E)} \theta(v) \). As above, this implies that the two terms \( \theta(u) \) and \( \theta(v) \) evaluated to the same element of \( A \), which shows that \( A \) satisfies the identities in \( F \).

Let \( f \in \Sigma^e \) be an \( n \)-ary function symbol and \( s_1, t_1, \ldots, s_n, t_n \in G(\Sigma, C) \) be terms such that the terms \( f(s_1, \ldots, s_n) \) and \( f(t_1, \ldots, t_n) \) evaluate to the same element of \( A \). Then \( s_1, \ldots, s_n \) and \( t_1, \ldots, t_n \) are evaluated to the same element of \( \hat{R}^{F}.\Sigma^e(E) \), and thus these terms have the same canonical forms w.r.t. \( \hat{R}^{F}.\Sigma^e(E) \). Let \( s_1', t_1', \ldots, s_n', t_n' \) be the canonical forms of the terms \( s_1, t_1, \ldots, s_n, t_n \), respectively. Then the canonical forms of \( f(s_1, \ldots, s_n) \) and \( f(t_1, \ldots, t_n) \) are respectively identical to the canonical forms of \( f(s_1', \ldots, s_n') \) and \( f(t_1', \ldots, t_n') \). Since \( f \neq \Sigma^F \), no rule with root symbol from \( \Sigma^F \) on the left-hand side is applicable to the latter terms. Thus, the only possible rules applicable to these terms are flat function rules for \( f \). Consequently, there are two possible cases for how the (identical) canonical forms of \( f(s_1, \ldots, s_n) \) and \( f(t_1, \ldots, t_n) \) can look like:

1. The terms \( f(s_1, \ldots, s_n) \) and \( f(t_1, \ldots, t_n) \) respectively have the canonical forms \( f(s'_1, \ldots, s'_n) \) and \( f(t'_1, \ldots, t'_n) \), and the corresponding arguments are syntactically equal, i.e., \( s'_j = t'_j \) for \( j = 1, \ldots, n \). In this case, \( s_j \cong_{\hat{R}^{F}.\Sigma^e(E)} s'_j = t'_j \cong_{\hat{R}^{F}.\Sigma^e(E)} t_j \) for \( j = 1, \ldots, n \), and thus, for \( j = 1, \ldots, n \), the terms \( s_j \) and \( t_j \) evaluate to the same element of \( A \).

2. The terms \( f(s_1, \ldots, s_n) \) and \( f(t_1, \ldots, t_n) \) reduce to the same constant \( d \). Then \( \hat{R}^{F}.\Sigma^e(E) \) must contain the rules \( f(s'_1, \ldots, s'_n) \rightarrow d \) and \( f(t'_1, \ldots, t'_n) \rightarrow d \). By the construction of \( \hat{R}^{F}.\Sigma^e(E) \), we thus have \( s'_j = t'_j \) for all \( j = 1, \ldots, n \) since otherwise new constant rules would have been added in step (e) and the construction would not yet have terminated. This yields \( s_j \cong_{\hat{R}^{F}.\Sigma^e(E)} s'_j = t'_j \cong_{\hat{R}^{F}.\Sigma^e(E)} t_j \) for \( j = 1, \ldots, n \), and thus, for \( j = 1, \ldots, n \), the terms \( s_j \) and \( t_j \) evaluate to the same element of \( A \).

Summing up, we have thus shown that \( A \) also satisfies extensionality for the symbols in \( \Sigma^e \), which completes the proof of the only-if-direction. \( \square \)
Recall that $f(b, h(a)) \approx_{E'}^{F, \Sigma^e} c$ for the set of identities $E'$ of Example 3. We have already seen that these two terms rewrite to the same canonical form w.r.t. $\hat{R}^S(E) \cup R(F, \succ_{lpo}) = \hat{R}^{F, \Sigma^e}(E')$.

The following corollary is an easy consequence of the above theorem, the conditions satisfied if rewrite-based congruence closure is effective (polynomial), and Lemma 8.

**Corollary 3** Let $\Sigma_F \subseteq \Sigma$ and $\Sigma^e \subseteq \Sigma \setminus \Sigma_F$, $F$ be a set of identities formulated using the function symbols in $\Sigma_F$ and variables, and assume that rewrite-based congruence closure is effective (polynomial) for $\Sigma_F$ and $F$. Then the word problem for finite sets $E$ of ground identities w.r.t. $F$ and the extensional symbols in $\Sigma^e$ is decidable (in polynomial time), i.e., given terms $s_0, t_0 \in G(\Sigma, C_0)$, we can decide (in polynomial time) whether $s_0 \approx_{E, F}^{\Sigma^e} t_0$ holds or not.

In the case of a set of strongly shallow identities $S$ for a signature $\Sigma_S$ that is arity bounded, the results shown in the previous section imply that rewrite-based congruence closure is polynomial for $\Sigma_S$ and $S$. Thus, we obtain the following instance of Corollary 3.

**Corollary 4** Let $\Sigma_S \subseteq \Sigma$ and $\Sigma^e \subseteq \Sigma \setminus \Sigma_S$, $S$ be a set of strongly shallow identities for $\Sigma_S$, and assumed that $\Sigma_S$ is arity-bounded. Then the word problem for finite sets of ground identities $S$ and the extensional symbols in $\Sigma^e$ is decidable in polynomial time, i.e., given a finite set of ground identities $E \subseteq G(\Sigma, C_0) \times G(\Sigma, C_0)$ and terms $s_0, t_0 \in G(\Sigma, C_0)$, we can decide in polynomial time whether $s_0 \approx_{E}^{S, \Sigma^e} t_0$ holds or not if we assume that the maximal arity of the symbols in $\Sigma_S$ is bounded by a constant.

We have mentioned in the introduction that it is unclear how this polynomiality result could be obtained by a simple adaptation of the usual approach that restricts congruence closure to a polynomially large set of subterms determined by the input (informally called “small” terms in the following). The main problem is that one might have to generate identities between “large” terms before one can get back to a desired identity between “small” terms using extensionality. The question is now where our proof actually deals with this problem. The answer is: in Case 1 of the case distinction in the proof of the only-if-direction of Theorem 2. In fact, there we consider a derived identity $f(s_1, \ldots, s_n) \approx f(t_1, \ldots, t_n)$ such that the (syntactically identical) canonical forms of $f(s_1, \ldots, s_n)$ and $f(t_1, \ldots, t_n)$ are not a constant from $C$, but of the form $f(s'_1, \ldots, s'_n) = f(t'_1, \ldots, t'_n)$. Basically, this means that $f(s_1, \ldots, s_n)$ and $f(t_1, \ldots, t_n)$ are terms that are not equivalent modulo $E$ to subterms of terms occurring in $E$, since the latter terms have a constant representing them. Thus, these terms are “large” terms that potentially could cause a problem: an identity between them has been derived, and now extensionality applied to this identity yields new identities $s_j \approx t_j$ between smaller terms. Our proof shows that these identities can nevertheless be derived from $\hat{R}^{F, \Sigma^e}(E)$, and thus do not cause a problem. For the proof of Theorem 2 to go through, it is important that the canonical rewrite system $\hat{R}^{F, \Sigma^e}(E)$ works for all ground terms in $G(\Sigma, C)$, and not just for a finite set of “small” terms.

### 5 Symbols that are Commutative and Extensional

In the previous section, we have made the assumptions that the sets $\Sigma_F$ and $\Sigma^e$ are disjoint, i.e., we did not consider extensionality for symbols satisfying the identities in $F$. The reason is that, without this restriction, we might obtain more consequences than we bargained for. For example, it is easy to see that the presence of a commutative and extensional symbol would...
trivialize the equational theory. In fact, if \( f \) is assumed to be commutative and extensional, then commutativity yields \( f(s,t) \approx f(t,s) \) for all terms \( s, t \in G(\Sigma, C_0) \), and extensionality then yields \( s \approx t \). This shows that, in this case, the commutative and extensional congruence closure would be \( G(\Sigma, C_0) \times G(\Sigma, C_0) \), independently of \( E \), and thus even for \( E = \emptyset \).

To avoid this problem, one can, however, look at appropriate variants of extensionality. In this section, we restrict the attention to commutativity, and consider the following variant of extensionality for commutative function symbols \( f \), which we call \( c \)-extensionality:

\[
f(x_1, x_2) \approx f(y_1, y_2) \Rightarrow (x_1 \approx y_1 \wedge x_2 \approx y_2) \vee (x_1 \approx y_2 \wedge x_2 \approx y_1).
\]  

(3)

For example, if \( f \) is a commutative couple constructor, and two couples turn out to be equal, then we want to infer that they consist of the same two persons, independently of the order in which they were put into the constructor.

Unfortunately, adding such a rule makes the word problem coNP-hard, which can be shown by a reduction from validity of propositional formulae.

**Proposition 2** In the presence of at least one commutative and \( c \)-extensional symbol, the word problem for finite sets of ground identities is coNP-hard.

We prove this proposition by a reduction from validity of propositional formulae. Thus, consider a propositional formula \( \phi \), and let \( p_1, \ldots, p_n \) be the propositional variables occurring in \( \phi \). We take the constants 0 and 1, and for every \( i, 1 \leq i \leq n \), we view \( p_i \) as a constant symbol, and add a second constant symbol \( \overline{p_i} \). In addition, we consider the function symbols \( \lor, \land, \neg, f \), and assume that \( f \) is commutative and satisfies (3). We then consider ground identities that axiomatize the truth tables for \( \lor, \land, \neg \), i.e.,

\[
\begin{align*}
f_\lor(0,0) &\approx 0, & f_\lor(1,0) &\approx 1, & f_\lor(0,1) &\approx 1, & f_\lor(1,1) &\approx 1, \\
f_\land(0,0) &\approx 0, & f_\land(1,0) &\approx 0, & f_\land(0,1) &\approx 0, & f_\land(1,1) &\approx 1, \\
f_\neg(0) &\approx 1, & f_\neg(1) &\approx 0.
\end{align*}
\]  

(4)

In addition, we consider, for every \( i, 1 \leq i \leq n \), the identities

\[
f(p_i, \overline{p_i}) \approx f(0, 1).
\]

Let \( E_\phi \) be the set of these ground identities, and let \( t_\phi \) be the term obtained from \( \phi \) by replacing the Boolean operations \( \lor, \land, \neg \) by the corresponding function symbols \( f_\lor, f_\land, f_\neg \). Proposition 2 is now an immediate consequence of the following lemma.

**Lemma 9** The identity \( t_\phi \approx 1 \) holds in every algebra satisfying \( E_\phi \) together with (3) and commutativity of \( f \) iff \( \phi \) is valid.

**Proof** First assume that \( \phi \) is valid, and let \( A \) be an algebra satisfying \( E_\phi \) together with (3) and commutativity of \( f \). Since \( A \) satisfies the identities \( f(p_i, \overline{p_i}) \approx f(0, 1) \) for \( i = 1, \ldots, n \), as well as (3), we can deduce that \( p_i^A \in \{0^A, 1^A \} \) for all \( i = 1, \ldots, n \). Let \( v \) be the propositional valuation that satisfies \( v(p_i) = 1 \) iff \( p_i^A = 1^A \). Since \( \phi \) is valid, we know that \( v(\phi) = 1 \). Using this fact and the identities axiomatizing the truth tables for \( \lor, \land, \neg \), we obtain that \( t_\phi^A = 1^A \). Thus, the identity \( t_\phi \approx 1 \) holds in \( A \).

Conversely, assume that \( \phi \) is not valid, and let \( v \) be a propositional valuation such that \( v(\phi) = 0 \). Let \( B \) be the free commutative algebra with generators 0, 1 and the binary function symbol \( f \) as binary operation. Thus, the domain \( B \) of \( B \) consists of the equivalence classes of ground terms in \( G(\{f\}, \{0, 1\}) \) modulo commutativity. We expand \( B \) to an algebra \( A \) that also interprets the symbols \( p_1, \ldots, p_n, \overline{p_1}, \ldots, \overline{p_n}, f_\lor, f_\land, f_\neg \) as follows:
Example 4 We illustrate the nondeterministic construction using the identities 
\[ \Sigma \] identities axiomatize commutativity of the symbols in 
\[ f \] already syntactically equal.

\[ \hat{f} \] i.e., \[ f \hat{A} \] behaves like disjunction, \[ f_\lor^A \] behaves like conjunction, and \[ f_\lnot^A \] behaves like negation, and on tuples not in \[ \{0^B, 1^B\} \times \{0^B, 1^B\} \] (for \[ f_\lor \] and \[ f_\lnot \]) or \[ \{0^B, 1^B\} \] (for \[ f_- \]) they yield an arbitrary value, say \[ 0^B \];

\[ f^A \] is \[ f^B \].

It is easy to see that \( \mathcal{A} \) satisfies the identities in \( E_\phi \). In addition, commutativity of \( f \) and
\( (3) \) are satisfied since these properties hold in the free commutative algebra \( [20] \). It is now
easy to see that \( r^A_\phi = v(\phi)^B = 0^B = 0^A \), and thus \( \mathcal{A} \) does not satisfy the identity \( t_\phi \approx 1 \)
since in \( B \) the class of the generator \( 0 \) is different from the class of the generator \( 1 \).

To prove a complexity upper bound that matches the lower bound stated in Proposition 2,
we consider a finite signature \( \Sigma \), a finite set of ground identities \( E \subseteq G(\Sigma, C_0) \times G(\Sigma, C_0) \)
as well as sets \( \Sigma_c \subseteq \Sigma \) and \( \Sigma^e \subseteq \Sigma \) of commutative and extensional symbols, respectively,
and assume that the non-commutative extensional symbols in \( \Sigma^e \setminus \Sigma_c \) satisfy extensionality
\( (2) \), whereas the commutative extensional symbols in \( \Sigma^e \cap \Sigma_c \) satisfy e-extensionality \( (3) \). We
want to show that, in this setting, the problem of deciding, for given terms \( s_0, t_0 \in G(\Sigma, C_0) \),
whether \( s_0 \) is not equivalent to \( t_0 \) is in NP.

For this purpose, we employ a nondeterministic variant of Algorithm 2. In step (g) this
procedure applies Algorithm 1 for the special case where the set \( F_c \) of strongly shallow
identities axiomatize commutativity of the symbols in \( \Sigma_c \). We denote the canonical rewrite
system produced by an application of this algorithm to the current flat system \( R_i \) by \( \hat{R}^{F_c}_i \),
i.e., \( \hat{R}^{F_c}_i = \hat{R}^{F_c}(R_i) \cup R(F_c, >_{lpo}) \).

For extensional symbols \( f \in \Sigma^e \setminus \Sigma_c \), step (e) is also performed as in Algorithm 2. For
extensional symbols \( f \in \Sigma^e \cap \Sigma_c \), step (e) is modified as follows: for all pairs of distinct flat
function rules \( f(c_1, c_2) \rightarrow d, f(c'_1, c'_2) \rightarrow d \) in \( \hat{R}^{F_c}_i \), nondeterministically choose whether

\[ c_1 \text{ and } c'_1 \text{ as well as } c_2 \text{ and } c'_2 \] are to be identified, or

\[ c_1 \text{ and } c'_2 \text{ as well as } c_2 \text{ and } c'_1 \] are to be identified,

and then add the corresponding constant rules to \( R_{i+1} \) unless the respective constants are
already syntactically equal.

This nondeterministic algorithm has different runs, depending on the choices made in
the nondeterministic part of step (e). But each run \( r \) produces a rewrite system \( \hat{R}^{F_c \cdot \Sigma^e}(E) \).

Example 4 We illustrate the nondeterministic construction using the identities \( E_\phi \) for \( \phi = p \lor \lnot p \) from our coNP-hardness proof. Then \( E_\phi \) consists of the identities in \( (4) \) together
with the identity \( f(p, \overline{p}) \approx f(0, 1) \). Assuming an appropriate order on the constants, the
system \( R(E_\phi) \) contains, among others, the rules

\[ f_\lor(1, 0) \rightarrow c_{f_\lor(1,0)}, \quad c_{f_\lor(1,0)} \rightarrow 1, \quad f_\lor(0, 1) \rightarrow c_{f_\lor(0,1)}, \quad c_{f_\lor(0,1)} \rightarrow 1 \]

\[ f_-(-0) \rightarrow c_{f_-(-0)}, \quad c_{f_-(-0)} \rightarrow 1, \quad f_-(-1) \rightarrow c_{f_-(-1)}, \quad c_{f_-(-1)} \rightarrow 0 \]

\[ f(p, \overline{p}) \rightarrow c_{f(p,\overline{p})}, \quad f(1, 0) \rightarrow c_{f(1,0)}, \quad c_{f(p,\overline{p})} \rightarrow f(1,0). \]

In steps (a) and (b) of Algorithm 1, these rules are transformed into the form

\[ f_\lor(1, 0) \rightarrow 1, \quad c_{f_\lor(1,0)} \rightarrow 1, \quad f_\lor(0, 1) \rightarrow 1, \quad c_{f_\lor(0,1)} \rightarrow 1 \]

\[ f_-(-0) \rightarrow 1, \quad c_{f_-(-0)} \rightarrow 1, \quad f_-(-1) \rightarrow 0, \quad c_{f_-(-1)} \rightarrow 0 \]

\[ f(p, \overline{p}) \rightarrow c_{f(p,\overline{p})} \rightarrow f(1,0), \quad f(1,0) \rightarrow c_{f(1,0)}, \quad c_{f(p,\overline{p})} \rightarrow c_{f(1,0)}. \]
Since no new constant rule is added, the construction proceeds with step (e). Due to the presence of the rules \( f(p, \overline{p}) \rightarrow c_{f(1,0)} \) and \( f(1,0) \rightarrow c_{f(1,0)} \) for \( f \in \Sigma_e \cap \Sigma_e \), it now nondeterministically chooses between identifying \( p \) with 1 or with 0. In the first case, the constant rules \( p \rightarrow 1, \overline{p} \rightarrow 0 \) are added, and in the second \( p \rightarrow 0, \overline{p} \rightarrow 1 \) are added. In the next iteration, no new constant rules are added, and thus the construction terminates. It has two runs \( r_1 \) and \( r_2 \). The generated rewrite systems \( \hat{R}_{r_1}^{F_c, \Sigma_e}(E) \) and \( \hat{R}_{r_2}^{F_c, \Sigma_e}(E) \) share the rules in (5), but the first contains \( p \rightarrow 1 \) whereas the second contains \( p \rightarrow 0 \).

**Lemma 10** Every run \( r \) of the construction terminates after a linear number of iterations and the system \( \hat{R}_{r}^{F_c, \Sigma_e}(E) \) this run produces is canonical.

**Proof** Termination after a linear number of iterations can be shown as in the proof of Lemma 8. By Lemma 6, \( \hat{R}_{r}^{F_c, \Sigma_e}(E) \) is canonical since it is produced by an application of Algorithm 1 to a flat system \( R_n \).

Using the canonical rewrite systems \( \hat{R}_{r}^{F_c, \Sigma_e}(E) \), we can now characterize when an identity follows from \( E \) w.r.t. commutativity of the symbols in \( \Sigma_e \), extensionality of the symbols in \( \Sigma_e \setminus \Sigma_c \), and c-extensionality of the symbols in \( \Sigma_e \cap \Sigma_c \) as follows.

**Theorem 3** Let \( s_0, t_0 \in G(\Sigma, C_0) \). The identity \( s_0 \approx t_0 \) holds in every algebra that satisfies \( E \), commutativity for every \( f \in \Sigma_c \), extensionality for every \( f \in \Sigma_e \setminus \Sigma_c \), and c-extensionality for every \( f \in \Sigma_e \cap \Sigma_c \) iff \( s_0, t_0 \) have the same canonical forms w.r.t. \( \hat{R}_{r}^{F_c, \Sigma_e}(E) \) for every run \( r \) of the nondeterministic construction.

**Proof** First, we show the if-direction. Thus, assume that \( s_0, t_0 \) have the same canonical forms w.r.t. \( \hat{R}_{r}^{F_c, \Sigma_e}(E) \) for every run \( r \). Let \( A \) be an algebra that satisfies \( E \), commutativity for every \( f \in \Sigma_c \), extensionality for every \( f \in \Sigma_e \setminus \Sigma_c \), and c-extensionality for every \( f \in \Sigma_e \cap \Sigma_c \). We must show that \( s_0^A = t_0^A \).

To this purpose, we expand \( A \) to the new constants in \( C_1 \) by setting \( c_s^A := s^A \), and call the expansion obtained this way still \( A \). Since \( A \) is a model of \( E \), it is easy to see that it is also a model of \( R(E) \). In addition, it satisfies the rewrite rules (viewed as identities) added in steps (a) and (b) of Algorithm 1. The same is true for the rules added in step (e) of the modified version of Algorithm 2 when treating symbols in \( \Sigma_e \setminus \Sigma_c \). For symbols \( f \in \Sigma_e \cap \Sigma_c \), we let \( A \) decide which option to take. To be more precise, for all pairs of distinct rules \( f(c_1, c_2) \rightarrow d, f(c_1', c_2') \rightarrow d \) in \( \hat{R}_{r}^{F_c, \Sigma_e}(E) \), we can assume by induction that \( f(c_1, c_2)^A = f^A(c_1^A, c_2^A) = d^A = f^A(c_1'^A, c_2'^A) = f(c_1', c_2')^A \) holds. Since \( A \) satisfies c-extensionality for \( f \), this implies that \( c_1^A = c_1'^A, c_2^A = c_2'^A \) or \( c_1^A = c_2'^A, c_2^A = c_1'^A \). If the former is the case, then we take the first option in the nondeterministic choice, and otherwise we take the second option. Overall, this yields a run \( r \) of the nondeterministic construction such that \( A \) is a model of \( \hat{R}_{r}^{F_c, \Sigma_e}(E) \). Note that \( A \) is a model of the rules in \( R(F_c, >_{IPO}) \) since it satisfies commutativity for every symbol in \( \Sigma_c \). Since \( s_0, t_0 \) have the same canonical forms w.r.t. \( \hat{R}_{r}^{F_c, \Sigma_e}(E) \), we know that \( s_0 \approx_{\hat{R}_{r}^{F_c, \Sigma_e}(E)} t_0 \). This yields \( s_0^A = t_0^A \) since \( A \) is a model of \( \hat{R}_{r}^{F_c, \Sigma_e}(E) \).

Second, we show the only-if-direction by contraposition. Thus, assume that there is a run \( r \) of the algorithm such that \( s_0, t_0 \) have different canonical forms w.r.t. \( \hat{R}_{r}^{F_c, \Sigma_e}(E) \). Since \( \hat{R}_{r}^{F_c, \Sigma_e}(E) \) is canonical, this implies that \( s_0 \not\approx_{\hat{R}_{r}^{F_c, \Sigma_e}(E)} t_0 \). As in the proof of Theorem 2, let \( A \) be the initial algebra for \( \hat{R}_{r}^{F_c, \Sigma_e}(E) \) viewed as a set of identities over the signature \( \Sigma \cup C \). Then \( s_0 \not\approx_{\hat{R}_{r}^{F_c, \Sigma_e}(E)} t_0 \) implies \( s_0^A \neq t_0^A \). Thus, it is sufficient to show that \( A \) satisfies \( E \),
commutativity for every \( f \in \Sigma_c \), extensionality for every \( f \in \Sigma^e \setminus \Sigma_c \), and c-extensionality for every \( f \in \Sigma^e \cap \Sigma_c \). Since the former three properties can be shown as in the proof of Theorem 2, we concentrate on the last one.

Thus, let \( f \in \Sigma^e \cap \Sigma_c \) be a commutative and c-extensional function symbol and \( s_1, t_1, s_2, t_2 \in G(\Sigma, C) \) be terms such that the terms \( f(s_1, s_2) \) and \( f(t_1, t_2) \) evaluate to the same element of \( A \). Again, this implies that these terms have the same canonical forms w.r.t. \( \widehat{R}_{E, f, \Sigma^e} (E) \). Let \( s'_1, t'_1, s'_2, t'_2 \) be the canonical forms of the terms \( s_1, t_1, s_2, t_2 \), respectively. As before, we distinguish several cases, but since \( f \) is commutative we now also need to take the rules in \( R(F_c, >_{lpo}) \), which are part of \( \widehat{R}_{E, f, \Sigma^e} (E) \), into account:

- First, assume that the canonical forms of \( f(s_1, s_2) \) and \( f(t_1, t_2) \) still have root symbol \( f \).
  We distinguish several subcases, depending on how the rules in \( R(F_c, >_{lpo}) \) have been applied:
  - The terms \( f(s_1, s_2) \) and \( f(t_1, t_2) \) have the canonical forms \( f(s'_1, s'_2) \) and \( f(t'_1, t'_2) \), respectively, and the corresponding arguments are syntactically equal, i.e., \( s'_1 = t'_1 \) and \( s'_2 = t'_2 \). As in the proof of Theorem 2, this implies that \( s_1 \) and \( t_1 \) as well as \( s_2 \) and \( t_2 \) respectively evaluate to the same elements of \( A \).
  - The case where the terms \( f(s_1, s_2) \) and \( f(t_1, t_2) \) have the canonical forms \( f(s'_2, s'_1) \) and \( f(t'_2, t'_1) \), respectively, can be handled in the same way since then again \( s'_1 = t'_1 \) and \( s'_2 = t'_2 \).
  - The terms \( f(s_1, s_2) \) and \( f(t_1, t_2) \) have the canonical forms \( f(s'_1, s'_2) \) and \( f(t'_1, t'_2) \), respectively, and the corresponding arguments are syntactically equal, i.e., \( s'_2 = t'_1 \) and \( s'_1 = t'_2 \). In this case we can derive that \( s_2 \) and \( t_1 \) as well as \( s_1 \) and \( t_2 \) respectively evaluate to the same elements of \( A \).
  - The case where the terms \( f(s_1, s_2) \) and \( f(t_1, t_2) \) have the canonical forms \( f(s'_1, s'_2) \) and \( f(t'_1, t'_2) \), respectively, can be handled in the same way since then again \( s'_2 = t'_1 \) and \( s'_1 = t'_2 \).

- Second, assume that the canonical forms of \( f(s_1, s_2) \) and \( f(t_1, t_2) \) are the same constant \( d \). Again, we distinguish several subcases, depending on how the rules in \( R(F_c, >_{lpo}) \) have been applied before the reduction to the constant \( d \):
  - The term rewriting system \( \widehat{R}_{E, f, \Sigma^e} (E) \) contains the rules \( f(s'_1, s'_2) \rightarrow d \) and \( f(t'_1, t'_2) \rightarrow d \). By the construction of \( \widehat{R}_{E, f, \Sigma^e} (E) \), we thus have \( s'_1 = t'_1 \) and \( s'_2 = t'_2 \), or \( s'_1 = t'_2 \) and \( s'_2 = t'_1 \) since no new constant rules have been added. This implies that \( s_1 \) and \( t_1 \) as well as \( s_2 \) and \( t_2 \) respectively evaluate to the same elements of \( A \), or \( s_2 \) and \( t_2 \) as well as \( s_1 \) and \( t_1 \) respectively evaluate to the same elements of \( A \).
  - The cases where the function rules reducing to \( d \) contain the arguments of \( f \) in other permutations can be treated in the same way.

Summing up, we have thus shown that \( A \) satisfies c-extensionality for the symbols in \( \Sigma^e \cap \Sigma_c \), which completes the proof of the theorem. \( \square \)

Coming back to Example 4, we note that \( \phi = p \lor \neg p \) is valid, and thus (by Lemma 9), the identity \( f_{\lor}(p, f_{\neg}(p)) \approx 1 \) holds in all algebras that satisfy \( E_\phi \) and interpret \( f \) as a commutative and c-extensional symbol. Using the rewrite system generated by the run \( r_1 \), we obtain the following rewrite sequence:

\[
 f_{\lor}(p, f_{\neg}(p)) \rightarrow f_{\lor}(1, f_{\neg}(p)) \rightarrow f_{\lor}(1, f_{\neg}(1)) \rightarrow f_{\lor}(1, 0) \rightarrow 1.
\]
For the run \( r_2 \), we obtain the sequence
\[
\begin{align*}
&f\vee(p, f\neg(p)) \rightarrow f\vee(0, f\neg(p)) \rightarrow f\vee(0, f\neg(0)) \rightarrow f\vee(0, 1) \rightarrow 1.
\end{align*}
\]

Thus, for both runs the terms \( f\vee(p, f\neg(p)) \) and 1 have the same canonical form 1.

Together with Proposition 2, Theorem 3 yields the following complexity results.

**Corollary 5** Consider a finite set of ground identities \( E \subseteq G(\Sigma, C_0) \times G(\Sigma, C_0) \) as well as sets \( \Sigma_c \subseteq \Sigma \) and \( \Sigma^e \subseteq \Sigma \) of commutative and extensional symbols, respectively, and two terms \( s_0, t_0 \in G(\Sigma, C_0) \). The problem of deciding whether the identity \( s_0 \approx t_0 \) holds in every algebra that satisfies \( E \), commutativity for every \( f \in \Sigma_c \), extensionality for every \( f \in \Sigma^e \setminus \Sigma_c \), and \( c \)-extensionality for every \( f \in \Sigma^e \cap \Sigma_c \) is coNP-complete.

**Proof** Since Proposition 2 yields coNP-hardness, it is sufficient to show that the complement problem is in NP. This is an easy consequence of Theorem 3. In fact, to show that \( s_0 \approx t_0 \) does not hold in all such algebras, it is sufficient to generate one run \( r \) of our nondeterministic construction, and then test whether \( s_0 \) and \( t_0 \) have different canonical forms w.r.t. \( \hat{RF}_{c, \Sigma^e}(E) \). The system \( \hat{RF}_{c, \Sigma^e}(E) \) can be generated in nondeterministic polynomial time, and the canonical forms can then be computed in polynomial time. \( \square \)

### 6 Related Work

As already mentioned in the introduction, several approaches for how to solve the word problem for finite sets of ground identities by constructing a canonical rewrite system have been developed in the literature [7–9]. Restricted to the case of uninterpreted function symbols, the approach employed in the present paper is similar to the rewriting-based congruence closure algorithm developed in [7]. In the presence of interpreted function symbols, i.e., ones that are must satisfy certain non-ground identities, our approach differs from previous ones [12, 14, 15] by the use of an ordered rewrite system. Basically, in our approach the ordered rewrite system deals with the semantic properties of the interpreted function symbols, and the only interaction with the construction of the rewrite system for the ground identities is through computing canonical forms of the left-hand sides of function rules using the ordered rewrite system. Otherwise, the construction is the same as for the case of uninterpreted function symbols. The papers [14, 15], which consider symbols that are associativity and commutativity (AC), deal with the problem that AC does not allow for a terminating non-deterministic construction, and then test whether \( s_0 \) and \( t_0 \) have different canonical forms w.r.t. \( \hat{R}_{F, \Sigma^c}(E) \). The system \( \hat{R}_{F, \Sigma^c}(E) \) can be generated in nondeterministic polynomial time, and the canonical forms can then be computed in polynomial time. \( \square \)
the present paper. According to [16], an identity is shallow if every variable occurs at depth at most one in the left- and right-hand side of the identity. In particular, this notion of shallow also encompasses ground identities and identities like idempotency [i.e. \( f(x, x) = x \)]. It is shown in [16] that the word problem for shallow theories is decidable. The approach basically applies a completion procedure akin to unfailing completion [22] to obtain an ordered rewrite system whose ground instantiation is canonical. This yields decidability of the word problem as considered in our Corollary 1, but no complexity result is stated in [16]. Extensionality is not considered in that paper. Our restriction to sets of strongly shallow identities \( S \) allows us to obtain an ordered rewrite system \((S^*, >_{lpo})\) whose ground instantiation \( R(S^*, >_{lpo})\) is canonical in a very simple way. For more general shallow identities, this simple construction would not work (e.g., idempotency would destroy finiteness of \( S^* \)). In particular, our polynomiality results depend of the fact that \( S^* \) is of polynomial size if the signature \( \Sigma_S \) is arity-bounded.

Lynch et al. [23, 24] introduce a method called schematic saturation for automatically generating a ground satisfiability procedure for a given theory \( T \) by combining the axioms of \( T \) with ground clauses using paramodulation. Similar to Knuth-Bendix completion, schematic saturation need not terminate. However, if it does terminate, then the word problem for ground identities w.r.t. the theory \( T \) is decidable in polynomial time by superposition because superposition only generates polynomially many clauses. Similar to our complexity result for strongly shallow identities stated in Corollary 1, this polynomiality result needs the assumption that the signature used by the theory \( T \) is arity-bounded. If applied to commutativity and clauses specifying extensionality for non-commutative symbols, schematic saturation terminates,\(^4\) which means that superposition can be used to decides the word problem for ground identities w.r.t. that theory in polynomial time. Schematic saturation does not terminate in the presence of commutative symbols that are c-extensional. But it can be extended by one more deletion rule, which then leads to termination.\(^5\) If schematic saturation with the deletion rule halts on the clauses representing some theory, then the word problem for ground identities w.r.t. that theory is again decidable by superposition, but now it may generate exponentially many clauses. This appears to yield an ExpTime decision procedure for the coNP-complete word problem considered in Corollary 5. The problem with applying the results in [23, 24] is that there are no criteria known that are both easy to check and that guarantee that schematic saturation (with or without the additional deletion rule) terminates. One must run the procedure and see what happens. Unfortunately, there is currently no publicly accessible implementation of this procedure available that could be used for doing this.

The paper [25] develops a procedure that can combine decision procedures for the word problem for equational theories that share only constants. To be more precise, the following result is shown there.

**Theorem 4** (Baader and Tinelli [25]) Let \( E := E_1 \cup E_2 \), where, for \( i = 1, 2 \), \( E_i \) is a non-trivial equational theory over \( \Sigma_i \). Furthermore, assume that \( \Sigma := \Sigma_1 \cap \Sigma_2 \) is a finite set of constant symbols, and that \( E_1 \) and \( E_2 \) agree on \( \Sigma \), that is, \( c =_{E_1} d \) iff \( c =_{E_2} d \) for all \( c, d \in \Sigma \).

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

Though it is not explicitly stated in [25], the combination procedure preserves polynomiality for the word problem, i.e., if the word problem is decidable in polynomial time for \( E_1 \)

---

\(^4\) Christoper Lynch, personal communication, July 15, 2020.

\(^5\) Christoper Lynch, personal communication, July 15, 2020.
and $E_2$, then it is also decidable in polynomial time for $E$ (see the termination proof for the procedure in [26]).

This result can be applied in the setting of ground identities with interpreted function symbols considered in the present paper. Indeed, assume that $F$ and $G$ are equational theories over the disjoint signatures $\Sigma_F$ and $\Sigma_G$ such that the word problem for finite set of ground identities w.r.t. $F$ is decidable, and the same is true for $G$. Then the word problem for finite sets of ground identities w.r.t. $F \cup G$ is decidable as well. In fact, while the original ground identities may then contain symbols from both $\Sigma_F$ and $\Sigma_G$, flattening as in our construction of the rewrite system $R(E)$ can be used to obtain a conservative extension of the input theory in which only constants are shared between the $F$ and the $G$ part. These theories may not yet agree on the shared constants, but this can be achieved by iteratedly adding identities between the shared constants derived from one theory to the other. This way, we can, for example, combine the decision procedure for the case of ground identities w.r.t. strongly shallow identities described in Sect. 3 with a decision procedure for ground identities w.r.t. AC symbols [14, 15, 27, 28]. This kind of combination also works in the presence of extensional symbols. In fact, while extensionality is axiomatized by conditional identities rather than identities, our results in Sect. 4 show that we can represent its effect using the equational theory axiomatized by the computed canonical rewrite system, which allows us to apply the above combination result.

7 Conclusion

We have shown, using a rewriting-based approach, that adding strongly shallow identities (like commutativity) and extensionality for certain function symbols to a finite set of ground identities leaves the complexity of the word problem in P. In contrast, adding c-extensionality for commutative function symbols raises the complexity to coNP. For classical congruence closure, it is well-known that it can actually be computed in $O(n \log n)$ [4, 11]. Since this complexity upper bound can also be achieved using a rewriting-based approach [9, 12], we believe that the approach developed here can also be used to obtain an $O(n \log n)$ upper bound for the word problem for ground identities in the presence of commutativity (and maybe other kinds of strongly shallow identities) and extensionality, as in Sect. 4, but this question was not in the focus of the present paper.

Our result on adding extensionality is actually not restricted to the case of interpreted function symbols whose semantic properties are defined by strongly shallow identities. Instead, we exhibited general properties that the canonical rewrite system computed by semantic congruence closure for the equational theory under consideration must satisfy such that extensionality can be treated in the simple way described in Sect. 4. An important condition is that the system is canonical for all ground terms, and not just a finite set of “small” terms determined by the input. It would be interesting to see whether rewriting-based approaches to congruence closure for other theories produced rewrite systems that satisfy the conditions stated in Sect. 4. For the rewriting-based approaches that deal with associative-commutative symbols proposed in [14, 15], this is not directly the case since they use rewriting modulo AC. However, our approach for adding extensional symbols in Sect. 4 can probably be extended to deal with rewriting modulo AC since the AC symbols are not extensional. We have mentioned in the previous section that extensionality in the presence of non-extensional AC symbols can be dealt with by the combination results in [25]. However, while this approach yields a decision procedure for the word problem, it does not provide us with a canonical ground
rewrite system. Another, and more challenging question would be to come up with an appropriate notion of extensionality for AC symbols, and to show that adding it leaves the word problem decidable.

In our treatment of strongly shallow identities, we have used a rather brute-force way to obtain an ordered rewrite system that is canonical on ground terms, which basically adds all implied identities between strongly shallow \( h \)-terms (see the definition of the set \( S^* \) in Sect. 2.3). Instead, one could use unfailing completion \([22]\) to add only those consequences that are needed to achieve ground confluence. The results in \([16]\) show that this always terminates on shallow (and thus also strongly shallow) identities. The interesting question would then be under what conditions on the strongly shallow identities this actually terminates in polynomial time without requiring arity-boundedness. As mentioned in the previous section, the approach used in \([16]\) can actually deal with more general kinds of shallow identities than the strongly shallow ones considered here. It produces an ordered rewrite system that is canonical for ground terms, but it is not clear whether, with the more general notion of shallow identities employed in \([16]\), this system satisfies the properties required in Sect. 4 for adding extensionality. For strongly shallow identities other than commutativity, it would also be interesting to come up with appropriate variants of \( c \)-extensionality.

Regarding the application motivation from DL, it should be easy to extend tableau-based algorithms for DLs to deal with individuals named by ground terms and identities between these terms. Basically, the tableau algorithm then works with the canonical forms of such terms, and if it identifies two terms (e.g., when applying a tableau-rule dealing with number restrictions), then the rewrite system and the canonical forms need to be updated. More challenging would be a setting where rules are added to the knowledge base that generate new terms if they find a certain constellation in the knowledge base (e.g., a married couple, for which the rule introduces a ground term denoting the couple and assertions that link the couple with its components). In the context of first-order logic and modal logics, the combination of tableau-based reasoning and congruence closure has respectively been investigated in \([29]\) and \([30]\).

**Acknowledgements** The authors would like to thank the reviewers (of the original IJCAR 2020 paper \([13]\) and of this journal version) for their careful reading and their useful comments, which helped to improve the presentation of this article. We also thank Barbara Morawska and Christopher Lynch for helpful discussions. Franz Baader was partially supported by the AI competence center ScADSI.AI Dresden/Leipzig and the Deutsche Forschungsgemeinschaft (DFG), Grant 389792660, within TRR 248. Deepak Kapur was partially supported by the National Science Foundation (NSF) CCF 1908804.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

**References**

1. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An introduction to description logic. Cambridge University Press, Cambridge (2017)
2. Kozen, D.: Complexity of finitely presented algebras. In: Proceedings of the 9th ACM symposium on theory of computing, pp. 164–177. ACM (1977)
3. Downey, P.J.; Sethi, R.; Tarjan, R.E.: Variations on the common subexpression problem. J. ACM 27(4), 758–771 (1980)
4. Nelson, G.; Oppen, D.: Fast decision procedures based on congruence closure. J. ACM 27(2), 356–364 (1980)
5. Baader, F.; Nipkow, T.: Term rewriting and all that. Cambridge University Press, Cambridge (1998)
6. Shostak, R.E.: An algorithm for reasoning about equality. Commun. ACM 21(7), 583–585 (1978)
7. Kapur, D.: Shostak’s congruence closure as completion. In: Proceedings of the 8th international conference on rewriting techniques and applications (RTA 1997), lecture notes in computer science, vol. 1232, pp. 23–37. Springer, Berlin (1997)
8. Gallier, J.H.; Narendran, P.; Plaisted, D.A.; Raatz, S.; Snyder, W.: An algorithm for finding canonical sets of ground rewrite rules in polynomial time. J. ACM 40(1), 1–16 (1993)
9. Snyder, W.: A fast algorithm for generating reduced ground rewriting systems from a set of ground equations. J. Symb. Comput. 15(4), 415–450 (1993)
10. Stump, A.; Barrett, C.W.; Dill, D.L.; Levitt, J.R.: A decision procedure for an extensional theory of arrays. In: Proceedings of 16th annual IEEE symposium on logic in computer science (LICS 2001), pp. 29–37. IEEE Computer Society (2001)
11. Nieuwenhuis, R.; Oliveras, A.: Fast congruence closure and extensions. Inf. Comput. 205(4), 557–580 (2007)
12. Kapur, D.: Conditional congruence closure over uninterpreted and interpreted symbols. J. Syst. Sci. Complex. 32(1), 317–355 (2019)
13. Baader, F.; Kapur, D.: Deciding the word problem for ground identities with commutative and extensional symbols. In: Peltier, N.; Sofronie-Stokkermans, V. (eds.) Proceedings of the 10th international joint conference on automated reasoning IJCAR 2020. Part I, lecture notes in computer science, vol. 12166, pp. 163–180. Springer, Cham (2020)
14. Narendran, P.; Rusinowitch, M.: Any ground associative-commutative theory has a finite canonical system. J. Autom. Reason. 17(1), 131–143 (1996)
15. Bachmair, L.; Ramakrishnan, I.V.; Tiwari, A.; Vigneron, L.: Congruence closure modulo associativity and commutativity. In: Kirchner, H.; Ringeissen, C. (eds.) Proceedings of the third international workshop on frontiers of combining systems (FroCoS 2000), lecture notes in computer science, vol. 1794, pp. 245–259. Springer, Berlin (2000)
16. Comon, H.; Haberstra, M.; Jouannaud, J.-P.: Syntacticness, cycle-syntacticness, and shallow theories. Inf. Comput. 111(1), 154–191 (1994)
17. Martin, U.; Nipkow, T.: Ordered rewriting and confluence. In: M.E. Stickel (ed.) Proceedings of the 10th international conference on automated deduction (CADE 1990), lecture notes in computer science, vol. 449, pp. 366–380. Springer (1990)
18. Matijasevich, Y.: Simple examples of undecidable associative calculi. Sov. Math. (Doklady) 8(2), 555–557 (1967)
19. Wechler, W.: Universal algebra for computer scientists, EATCS monographs on theoretical computer science, vol. 25. Springer, New York (1992)
20. Siekmann, J.H.: Unification of commutative terms. In: Proceedings of the international symposium on symbolic and algebraic manipulation, EUROCAM’79, lecture notes in computer science, vol. 72, pp. 531–545. Springer, Marseille, France (1979)
21. Mayr, E.W.; Meyer, A.R.: The complexity of the word problems for commutative semigroups and polynomial ideals. Adv. Math. 46(3), 305–329 (1982)
22. Bachmair, L.: Canonical equational proofs. Birkhäuser, Boston (1991)
23. Lynch, C.; Morawska, B.: Automatic decidability. In: Proceedings of the 17th IEEE symposium on logic in computer science (LICS 2002), pp. 7–16. IEEE Computer Society (2002)
24. Lynch, C.; Ranise, S.; Ringeissen, C.; Tran, D.: Automatic decidability and combinability. Inf. Comput. 209(7), 1026–1047 (2011)
25. Baader, F.; Tinelli, C.: A new approach for combining decision procedure for the word problem, and its connection to the Nelson-Oppen combination method. In: McCune, W. (ed.) Proceedings of the 14th international conference on automated deduction (CADE 1997), lecture notes in computer science, vol. 1249, pp. 19–33. Springer, Berlin (1997)
26. Baader, F.; Tinelli, C.: A new approach for combining decision procedures for the word problem, and its connection to the Nelson-Oppen combination method. LTCS-report LTCS-96-01, LuFg theoretical computer science, RWTH Aachen (1996). https://tu-dresden.de/ing/informatik/thi/lat/forschung/technische-berichte
27. Kandri-Rody, A., Kapur, D., Narendran, P.: An ideal-theoretic approach to work problems and unification problems over finitely presented commutative algebras. In: Jouannaud, J. (ed.) Proceedings of the first international conference on rewriting techniques and applications (RTA 1985), lecture notes in computer science, vol. 202, pp. 345–364. Springer, Berlin (1985)

28. Kapur, D.: A modular associative commutative (AC) congruence closure algorithm. In: Kobayashi, N. (ed.) 6th international conference on formal structures for computation and deduction (FSCD 2021), LIPIcs, vol. 195, p. 15:1-15:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Wadern (2021)

29. Käufl, T., Zabel, N.: The theorem prover of the program verifier Tatzelwurm. In: Stickel, M.E. (ed.) Proceedings of the 10th international conference on automated deduction (CADE’90), lecture notes in computer science, vol. 449, pp. 657–658. Springer, Berlin (1990)

30. Schmidt, R.A., Waldmann, U.: Modal tableau systems with blocking and congruence closure. In: H. de Nivelle (ed.) Proceedings of the 24th international conference on automated reasoning with analytic tableaux and related methods (TABLEAUX 2015), lecture notes in computer science, vol. 9323, pp. 38–53. Springer (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.