Deciding the Word Problem for Ground Identities with Commutative and Extensional Symbols

Franz Baader[1](✉) and Deepak Kapur[2]

[1] Theoretical Computer Science, TU Dresden, Dresden, Germany
franz.baader@tu-dresden.de

[2] Department of Computer Science, University of New Mexico, Albuquerque, USA
kapur@cs.unm.edu

Abstract. The word problem for a finite set of ground identities is known to be decidable in polynomial time, and this is also the case if some of the function symbols are assumed to be commutative. We show that decidability in P is preserved if we also assume 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.

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* [3,5,10,12]. Basically, congruence closure starts with the

F. Baader—Partially supported by DFG, Grant 389792660, within TRR 248.

N. Peltier and V. Sofronie-Stokkermans (Eds.): IJCAR 2020, LNAI 12166, pp. 163–180, 2020.
https://doi.org/10.1007/978-3-030-51074-9_10

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 [15], and made explicit in [7] (see also [6,16] for other rewriting-based approaches).

In the motivating example from DL, but also in other settings where congruence closure is employed (such as SMT [13,17]), it sometimes makes sense to assume that certain function symbols satisfy additional properties that are not expressible by finitely many ground identities. For example, one may want to considered 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 rule $f(x, y) \approx f(x', y') \Rightarrow x \approx x' \wedge y \approx y'$. While it is known that adding commutativity does not increase the complexity (see, e.g., [5,8]), 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 applying 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.

In the next section, we show how the rewriting-based approach of [7] can be extended such that it can also handle commutative symbols. In contrast to approaches that deal with associative-commutative (AC) symbols [4,11] using rewriting modulo AC, we treat commutativity by introducing an additional rewrite system consisting of appropriately ordered ground instances of commutativity. This sets the stage for our rewriting-based approach that works in the presence of commutative symbols and extensional symbols presented in Sect. 4. In this section, we do not consider symbols f that are both commutative and extensional since extensionality as defined until now is not appropriate for commutative symbols: for arbitrary terms s, t, commutativity yields $f(s, t) \approx f(t, s)$, and thus extensionality implies $s \approx t$, which shows that the equational theory becomes trivial. In Sect. 5, we introduce the notion of c-extensionality, which is more appropriate for commutative symbols. Whereas the approaches developed in Sects. 3 and 4 yield polynomial time decision procedures for the word problem, c-extensionality makes the word problem coNP-complete.

Due to space constraints not all proofs can be given here. Detailed proofs can be found in [2].

2 Preliminaries on Equational Theories and Term Rewriting

We assume that the reader is familiar with basic notions and results regarding equational theories, universal algebra, and term rewriting. Here, we briefly recall the most important notions and results and refer the readers to [3] for details. We will keep as close as possible to the notation introduced in [3]. In particular, we use $\approx$ to denote identities between terms and $=$ to denote syntactic equality.

Terms are built as usual from variables, constants, and function symbols. 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 identities E, the equational theory induced by E is defined (semantically) as $\approx_E := \{s \approx t \mid \text{every models of } E \text{ is a model of } s \approx t\}$. The notion of *model* used here is the usual one from first-order logic, where 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 provides us with an alternative characterization of $\approx_E$ that is based on rewriting. A given set of identities E induces a binary relation $\rightarrow_E$ on terms. Basically, we have $s \rightarrow_E t$ if there is an identity $\ell \approx r$ in E such that s contains a substitution instance $\sigma(\ell)$ of ℓ as subterm, and t is obtained from s by replacing this subterm with $\sigma(r)$. Birkhoff's theorem says that $\approx_E$ is identical to $\overset{*}{\leftrightarrow}_E$, where $\overset{*}{\leftrightarrow}_E$ denotes the reflexive, transitive, and symmetric closure of $\rightarrow_E$.

If $\rightarrow_E$ is canonical (i.e., terminating and confluent), then we have $s \overset{*}{\leftrightarrow}_E t$ iff s and t have the same canonical forms. The *canonical form* of a term s is an irreducible term $\widehat{s}$ such that $s \overset{*}{\rightarrow}_E \widehat{s}$, where $\overset{*}{\rightarrow}_E$ denotes the reflexive and transitive closure of $\rightarrow_E$ and $\widehat{s}$ is *irreducible* if there is no s' with $\widehat{s} \rightarrow_E s'$. Termination ensures that the canonical form exists and confluence that it is unique. The relation $\rightarrow_E$ is *confluent* if $s \overset{*}{\rightarrow}_E t_1$ and $s \overset{*}{\rightarrow}_E t_2$ imply that there is a term t such that $t_1 \overset{*}{\rightarrow}_E t$ and $t_2 \overset{*}{\rightarrow}_E t$. It is *terminating* if there is no infinite chain $t_0 \rightarrow_E t_1 \rightarrow_E t_2 \rightarrow_E \cdots$.

Termination can be proved using a so-called *reduction order*, which is a well-founded order $>$ on terms such that $\ell > r$ for all $\ell \approx r \in E$ implies $s > t$ for all terms s, t with $s \rightarrow_E t$. Since $>$ is well-founded this then implies termination. If $\rightarrow_E$ is terminating, then confluence can be tested by checking whether all critical pairs of E are joinable. Basically, *critical pairs* (t_1, t_2) consider the most general forks of the form $s \rightarrow_E t_1$ and $s \rightarrow_E t_2$ that are due to overlapping left-hand sides of identities. Such a pair is *joinable* if there is a term t such that $t_1 \overset{*}{\rightarrow}_E t$ and $t_2 \overset{*}{\rightarrow}_E t$.

Usually, when considering the relation $\rightarrow_E$, one calls E a *term rewriting system* rather than a set of identities, and writes its elements (called rewrite rules) as $\ell \rightarrow r$ rather than $\ell \approx r$. From a formal point of view, however, both rewrite rules and identities are 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.

3 Commutative Congruence Closure Based on Rewriting

Let Σ be a finite set of function symbols of arity ≥ 1 and C_0 a finite set of constant symbols. We denote the set of ground terms built using symbols from Σ and C_0 with $G(\Sigma, C_0)$. In the following, let E be a finite set of identities $s \approx t$ between terms $s, t \in G(\Sigma, C_0)$, and $\approx_E$ the equational theory induced by E on $G(\Sigma, C_0)$, defined either semantically using algebras or (equivalently) syntactically through rewriting [3].

It is well-known (see, e.g., [3], 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, 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., [3], Theorem 4.3.5).

This actually also works if one adds commutativity of some binary function symbols to the theory. To be more precise, we assume that some of the binary function symbols in Σ are commutative, i.e., there is a set of binary function symbols $\Sigma_c \subseteq \Sigma$ whose elements we call *commutative* symbols. In addition to the identities in E, we assume that the identities $f(x, y) \approx f(y, x)$ are satisfied for all function symbols $f \in \Sigma_c$. From a semantic point of view, this means that we consider algebras $\mathcal{A}$ that satisfy not only the identities in E, but also *commutativity* for the symbols in Σ_c, i.e., for all $f \in \Sigma_c$, and all elements a, b of $\mathcal{A}$ we have that $f^{\mathcal{A}}(a, b) = f^{\mathcal{A}}(b, a)$. Given $s, t \in G(\Sigma, C_0)$, we say that $s \approx t$ *follows from* E *w.r.t. the commutative symbols in* Σ_c (written $s \approx_E^{\Sigma_c} t$) if $s^{\mathcal{A}} = t^{\mathcal{A}}$ holds in all algebras that satisfy the identities in E and commutativity for the symbols in Σ_c. The relation $\approx_E^{\Sigma_c} \subseteq G(\Sigma, C_0) \times G(\Sigma, C_0)$ can also be generated by extending congruence closure by a commutativity rule.

To be more precise, $CC^{\Sigma_c}(E)$ 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 commutativity rule:

- if $f \in \Sigma_c$ and $s, t \in G(\Sigma, C_0)$, then $f(s,t) \approx f(t,s) \in CC^{\Sigma_c}(E)$ (commutativity).

We call $CC^{\Sigma_c}(E)$ the *commutative congruence closure* of E. Using Birkhoff's theorem, it is easy to see that $CC^{\Sigma_c}(E)$ coincides with $\approx_E^{\Sigma_c}$ in the sense that $s \approx t \in CC^{\Sigma_c}(E)$ iff $s \approx_E^{\Sigma_c} t$ (see Lemma 3.5.13 and Theorem 3.5.14 in [3]). Again, it is not hard to show that the restriction of the commutative congruence closure to a polynomially large set of terms determined by the input E, s_0, t_0 is complete, which yields decidability of $\approx_E^{\Sigma_c}$ [5].

Here, we follow a different approach, which is based on rewriting [7,8]. Let $S(E)$ 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 $s \in S(E) \setminus C_0$. 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 $C := C_0 \cup C_1$. Given a term $u \in G(\Sigma, C)$, we denote with $\widehat{u}$ the term in $G(\Sigma, C_0)$ obtained from u by replacing the occurrences of the constants $c_s \in C_1$ in u with the corresponding terms $s \in S(E)$.

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 $s \in S(E) \setminus C_0$, 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 $f(c_{s_1}, \ldots, c_{s_n}) \to c_s$ to $R(E)$.
- For every identity $s \approx t \in E$ we add $c_s \to c_t$ to $R(E)$ if $c_s > c_t$, and $c_t \to c_s$ 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 $c \to d$ for $c > d$ and *function rules* of the form $f(c_1, \ldots, c_n) \to d$.

Example 1. Consider $E = \{f(a, g(a)) \approx c, g(b) \approx h(a), a \approx b\}$ with $\Sigma_c = \{f\}$. It is easy to see that we have $f(h(a), b) \approx_E^{\Sigma_c} c$. Using our construction, we first introduce the new constants $C_1 = \{c_{f(a,g(a))}, c_{g(a)}, c_{g(b)}, c_{h(a)}\}$. If we fix the linear order on C as $c_{f(a,g(a))} > c_{g(a)} > c_{g(b)} > c_{h(a)} > a > b > c$, then we obtain the following rewrite system: $R(E) = \{f(a, c_{g(a)}) \to c_{f(a,g(a))}, g(a) \to c_{g(a)}, g(b) \to c_{g(b)}, h(a) \to c_{h(a)}, c_{f(a,g(a))} \to c, c_{g(b)} \to c_{h(a)}, a \to b\}$.

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

Lemma 1. *For all terms $s \in S(E)$ we have $s \approx_{R(E)} c_s$. Consequently, $u \approx_{R(E)} \widehat{u}$ and thus also $u \approx_{R(E)}^{\Sigma_c} \widehat{u}$ for all terms $u \in G(\Sigma, C)$.*

Using this lemma, we can show that the construction of $R(E)$ is correct for consequence w.r.t. commutative symbols in the following sense:

Lemma 2. *Viewed as a set of identities, $R(E)$ is a conservative extension of E w.r.t. the commutative symbols in Σ_c, i.e., for all terms $s_0, t_0 \in G(\Sigma, C_0)$ we have $s_0 \approx_E^{\Sigma_c} t_0$ iff $s_0 \approx_{R(E)}^{\Sigma_c} t_0$.*

In this lemma, we use commutativity of the elements of Σ_c as additional identities. Our goal is, however, to deal both with the ground identities in E and with commutativity by rewriting. For this reason, we consider the rewrite system[1]

$$R(\Sigma_c) := \{f(s,t) \to f(t,s) \mid f \in \Sigma_c, s,t \in G(\Sigma, C), \text{ and } s >_{lpo} t\}, \qquad (1)$$

where $>_{lpo}$ denotes the lexicographic path order (see Definition 5.4.12 in [3]) induced by a linear order on $\Sigma \cup C$ that extends $>$ on C, makes each function symbol in Σ greater than each constant symbol in C, and linearly orders the function symbols in an arbitrary way. Note that $>_{lpo}$ is then a linear order on $G(\Sigma, C)$ (see Exercise 5.20 in [3]). Consequently, for every pair of distinct terms $s, t \in G(\Sigma, C)$, we have $f(s,t) \to f(t,s) \in R(\Sigma_c)$ or $f(t,s) \to f(s,t) \in R(\Sigma_c)$.

The term rewriting system $R(E) \cup R(\Sigma_c)$ can easily be shown to terminate using this order. In fact, $>_{lpo}$ is a reduction order, and we have $\ell >_{lpo} r$ for all rules $\ell \to r \in R(E) \cup R(\Sigma_c)$. However, in general $R(E) \cup R(\Sigma_c)$ need not be confluent. For instance, in Example 1 we have the two rewrite sequences $g(a) \to g(b) \to c_{g(b)} \to c_{h(a)}$ and $g(a) \to c_{g(a)}$ w.r.t. $R(E) \cup R(\Sigma_c)$, and the two constants $c_{h(a)}$ and $c_{g(a)}$ are irreducible w.r.t. $R(E) \cup R(\Sigma_c)$, but not equal.

We turn $R(E) \cup R(\Sigma_c)$ into a confluent and terminating system by modifying $R(E)$ appropriately. We start with $R_0^{\Sigma_c}(E) := R(E)$ and $i := 0$:

(a) Let $R_i^{\Sigma_c}(E)|_{con}$ consist of the constant rules in $R_i^{\Sigma_c}(E)$. For every constant $c \in C$, consider

$$[c]_i := \{d \in C \mid c \approx_{R_i^{\Sigma_c}(E)|_{con}} d\},$$

and let e be the least element in $[c]_i$ w.r.t. the order $>$. We call e the *representative* of c w.r.t. $R_i^{\Sigma_c}(E)$ and $>$. If $c \neq e$, then add $c \to e$ to $R_{i+1}^{\Sigma_c}(E)$.

(b) In all function rules in $R_i^{\Sigma_c}(E)$, replace each constant by its representative w.r.t. $R_i^{\Sigma_c}(E)$ and $>$, and call the resulting set of function rules $F_i^{\Sigma_c}(E)$. Then, we distinguish two cases, depending on whether the function symbol occurring in the rule is commutative or not.

(b1) Let f be an n-ary function symbol not belonging to Σ_c. For every term $f(c_1, \ldots, c_n)$ occurring as the left-hand side of a rule in $F_i^{\Sigma_c}(E)$, consider all the rules $f(c_1, \ldots, c_n) \to d_1, \ldots, f(c_1, \ldots, c_n) \to d_k$ in $F_i^{\Sigma_c}(E)$ with this left-hand side. Let d be the least element w.r.t. $>$ in $\{d_1, \ldots, d_k\}$. Add $f(c_1, \ldots, c_n) \to d$ and $d_j \to d$ for all j with $d_j \neq d$ to $R_{i+1}^{\Sigma_c}(E)$.

[1] Since this system is in general infinite, we do not generate it explicitly. But we can nevertheless apply the appropriate rule when encountering a commutative symbol during rewriting by just ordering its arguments according to $>_{lpo}$.

(b2) Let f be a binary function symbol belonging to Σ_c. For all pairs of constant symbols c_1, c_2 such that $f(c_1, c_2)$ or $f(c_2, c_1)$ is the left-hand side of a rule in $F_i^{\Sigma_c}(E)$, consider the set of constant symbols $\{d_1, \ldots, d_k\}$ occurring as right-hand sides of such rules, and let d be the least element w.r.t. $>$ in this set. Add $d_j \to d$ for all j with $d_j \neq d$ to $R_{i+1}^{\Sigma_c}(E)$. In addition, if $c_2 >_{lpo} c_1$, then add $f(c_1, c_2) \to d$ to $R_{i+1}^{\Sigma_c}(E)$, and otherwise $f(c_2, c_1) \to d$.

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

Let us illustrate the construction of $\widehat{R}^{\Sigma_c}(E)$ using Example 1. In step (a), the non-trivial equivalence classes are $[a]_0 = \{a, b\}$ with representative b, $[c_{f(a,g(a))}] = \{c_{f(a,g(a))}, c\}$ with representative c, and $[c_{g(b)}] = \{c_{g(b)}, c_{h(a)}\}$ with representative $c_{h(a)}$. Thus, $a \to b, c_{f(a,g(a))} \to c, c_{g(b)} \to c_{h(a)}$ are the constant rule added to $R_1^{\Sigma_c}(E)$. The function rules in $F_0^{\Sigma_c}(E)$ are then $f(b, c_{g(a)}) \to c, g(b) \to c_{g(a)}, g(b) \to c_{h(a)}, h(b) \to c_{h(a)}$. For the two rules with left-hand side $g(b)$, we add $c_{g(a)} \to c_{h(a)}$ and $g(b) \to c_{h(a)}$ to $R_1^{\Sigma_c}(E)$. The rules with left-hand sides different from $g(b)$ are moved unchanged from $F_0^{\Sigma_c}(E)$ to $R_1^{\Sigma_c}(E)$ since their left-hand sides are unique. Thus, $R_1^{\Sigma_c}(E) = \{a \to b, c_{f(a,g(a))} \to c, c_{g(b)} \to c_{h(a)}, c_{g(a)} \to c_{h(a)}, f(b, c_{g(a)}) \to c, g(b) \to c_{h(a)}, h(b) \to c_{h(a)}\}$.

In the second iteration step, we now have the new non-trivial equivalence class $[c_{g(b)}]_1 = \{c_{g(b)}, c_{h(a)}, c_{g(a)}\}$ with representative $c_{h(a)}$. The net effect of step (a) is, however, that the constant rules are moved unchanged from $R_1^{\Sigma_c}(E)$ to $R_2^{\Sigma_c}(E)$. The function rules in $F_1^{\Sigma_c}(E)$ are then $f(b, c_{h(a)}) \to c, g(b) \to c_{h(a)}, h(b) \to c_{h(a)}$. Consequently, no constant rules are added in step (b), and the construction terminates with output $\widehat{R}^{\Sigma_c}(E) = \{a \to b, c_{f(a,g(a))} \to c, c_{g(b)} \to c_{h(a)}, c_{g(a)} \to c_{h(a)}, f(b, c_{h(a)}) \to c, g(b) \to c_{h(a)}, h(b) \to c_{h(a)}\}$.

Our goal is now to show that $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$ provides us with a polynomial-time decision procedure for the commutative word problem in E.

Lemma 3. *The system $\widehat{R}^{\Sigma_c}(E)$ can be computed from $R(E)$ in polynomial time, and its construction is correct in the following sense: viewed as a set of identities, $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$ is equivalent to $R(E)$ with commutativity, i.e., for all terms $s, t \in G(\Sigma, C)$ we have $s \approx_{R(E)}^{\Sigma_c} t$ iff $s \approx_{\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)} t$.*

If we view $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$ as a term rewriting system, then we obtain the following result.

Lemma 4. *$\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$ is canonical, i.e., terminating and confluent.*

Proof. Termination of the term rewriting system $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$ can be shown as for $R(E) \cup R(\Sigma_c)$, using the reduction order $>_{lpo}$ introduced in the definition of $R(\Sigma_c)$. Confluence can thus be proved by showing that all non-trivial critical pairs of this system can be joined (see [2] for details). □

Since $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$ is canonical, each term $s \in G(\Sigma, C)$ has a unique normal form (i.e., irreducible term reachable from s) w.r.t. $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$, which we call the *canonical form* of s. We can thus use the system $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$ to decide whether terms s, t are equivalent w.r.t. E and commutativity of the symbols in Σ_c, i.e., whether $s \approx t \in CC^{\Sigma_c}(E)$, by computing the canonical forms of the terms s and t.

Theorem 1. *Let $s_0, t_0 \in G(\Sigma, C_0)$. Then we have $s_0 \approx t_0 \in CC^{\Sigma_c}(E)$ iff s_0 and t_0 have the same canonical form w.r.t. $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$.*

Consider the rewrite system $\widehat{R}^{\Sigma_c}(E)$ that we have computed (above Lemma 3) from the set of ground identities E in Example 1, and recall that $f(h(a), b) \approx_E^{\Sigma_c} c$. The canonical form of c is clearly c, and the canonical form of $f(h(a), b)$ can be computed by the following rewrite sequence:

$$f(h(a), b) \rightarrow_{R(\Sigma_c)} f(b, h(a)) \rightarrow_{\widehat{R}^{\Sigma_c}(E)} f(b, h(b)) \rightarrow_{\widehat{R}^{\Sigma_c}(E)} f(b, c_{h(a)}) \rightarrow_{\widehat{R}^{\Sigma_c}(E)} c.$$

Note that the construction of $\widehat{R}^{\Sigma_c}(E)$ is actually independent of the terms s_0, t_0 for which we want to decide the word problem in E. 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.

Since it is easy to show that reduction to canonical forms requires only a polynomial number of rewrite steps, Theorem 1 thus yields the following complexity result.

Corollary 1. *The commutative word problem for finite sets of ground identities is decidable in polynomial time, i.e., given a finite set of ground identities $E \subseteq G(\Sigma, C_0) \times G(\Sigma, C_0)$, a set $\Sigma_c \subseteq \Sigma$ of commutative symbols, and terms $s_0, t_0 \in G(\Sigma, C_0)$, we can decide in polynomial time whether $s_0 \approx_E^{\Sigma_c} t_0$ holds or not.*

This complexity result has been shown before in [5] and [8], but note that, in these papers, detailed proofs are given for the case without commutativity, and then it is only sketched how the respective approach can be extended to accommodate commutativity. Like the approach in this paper, the one employed in [8] is rewriting-based, but in contrast to ours it does not explicitly use the rewrite system $R(\Sigma_c)$.

4 Commutative Congruence Closure with Extensionality

Here, we additionally assume that some of the *non-commutative*[2] function symbols are extensional, i.e., there is a set of function symbols $\Sigma^e \subseteq \Sigma \setminus \Sigma_c$ whose elements we call *extensional* symbols. In addition to the identities in E and

[2] We will explain in the next section why the notion of extensionality introduced in (2) below is not appropriate for commutative symbols.

commutativity for the symbols in Σ_c, we now assume that also the following conditional identities are satisfied for every n-ary function symbol $f \in \Sigma^e$:

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

From a semantic point of view, this means that we now consider algebras $\mathcal{A}$ that satisfy not only the identities in E and commutativity for the symbols in Σ_c, but also *extensionality* for the symbols in Σ^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 $\mathcal{A}$ we have that $f^{\mathcal{A}}(a_1, \ldots, a_n) = f^{\mathcal{A}}(b_1, \ldots, b_n)$ implies $a_i = b_i$ for all $i, 1 \leq i \leq n$. Let $\Sigma_c^e = (\Sigma_c, \Sigma^e)$ and $s, t \in G(\Sigma, C_0)$. We say that $s \approx t$ follows from E w.r.t. the commutative symbols in Σ_c and the extensional symbols in Σ^e (written $s \approx_E^{\Sigma_c^e} t$) if $s^{\mathcal{A}} = t^{\mathcal{A}}$ holds in all algebras that satisfy the identities in E, commutativity for the symbols in Σ_c, and extensionality for the symbols in Σ^e.

The relation $\approx_E^{\Sigma_c^e} \subseteq G(\Sigma, C_0) \times G(\Sigma, C_0)$ can also be generated using the following extension of congruence closure by an extensionality rule. To be more precise, $CC^{\Sigma_c^e}(E)$ 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, commutativity, and the following extensionality rule:

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

Proposition 1. *For all terms $s, t \in G(\Sigma, C_0)$ we have $s \approx_E^{\Sigma_c^e} t$ iff $s \approx t \in CC^{\Sigma_c^e}(E)$.*

Proof. This proposition is an easy consequence of Theorem 54 in [18], which (adapted to our setting) says that $\approx_E^{\Sigma_c^e}$ is the least congruence containing E that is invariant under applying commutativity and extensionality. Clearly, this is exactly $CC^{\Sigma_c^e}(E)$. □

To obtain a decision procedure for $\approx_E^{\Sigma_c^e}$, we extend the rewriting-based approach from the previous section. Let the term rewriting system $R(E)$ be defined as in Sect. 3.

Example 2. Consider $E' = \{f(a, g(a)) \approx c, g(b) \approx h(a), g(a) \approx g(b)\}$ with $\Sigma_c = \{f\}$ and $\Sigma^e = \{g\}$. It is easy to see that we have $f(h(a), b) \approx_{E'}^{\Sigma_c^e} c$. Let the set C_1 of new constants and the linear order on all constants be defined as in Example 1. Now, we obtain the following rewrite system: $R(E') = \{f(a, c_{g(a)}) \to c_{f(a,g(a))}, g(a) \to c_{g(a)}, g(b) \to c_{g(b)}, h(a) \to c_{h(a)}, c_{f(a,g(a))} \to c, c_{g(b)} \to c_{h(a)}, c_{g(a)} \to c_{g(b)}\}$.

Lemma 5. *The system $R(E)$ is a conservative extension of E also w.r.t. the commutative symbols in Σ_c and the extensional symbols in Σ^e, i.e., for all terms $s_0, t_0 \in G(\Sigma, C_0)$ we have $s_0 \approx_E^{\Sigma_c^e} t_0$ iff $s_0 \approx_{R(E)}^{\Sigma_c^e} t_0$.*

We extend the construction of the confluent and terminating rewrite system corresponding to $R(E)$ by adding a third step that takes care of extensionality. To be more precise, $\widehat{R}^{\Sigma_c^e}(E)$ is constructed by performing the following steps, starting with $R_0^{\Sigma_c^e}(E) := R(E)$ and $i := 0$:

(a) Let $R_i^{\Sigma_c^e}(E)|_{con}$ consist of the constant rules in $R_i^{\Sigma_c^e}(E)$. For every constant $c \in C$, consider
$$[c]_i := \{d \in C \mid c \approx_{R_i^{\Sigma_c^e}(E)|_{con}} d\},$$
and let e be the least element in $[c]_i$ w.r.t. the order $>$. We call e the *representative* of c w.r.t. $R_i^{\Sigma_c^e}(E)$ and $>$. If $c \neq e$, then add $c \to e$ to $R_{i+1}^{\Sigma_c^e}(E)$.

(b) In all function rules in $R_i^{\Sigma_c^e}(E)$, replace each constant by its representative w.r.t. $R_i^{\Sigma_c^e}(E)$ and $>$, and call the resulting set of function rules $F_i^{\Sigma_c^e}(E)$. Then, we distinguish two cases, depending on whether the function symbol occurring in the rule is commutative or not.

(b1) Let f be an n-ary function symbol not belonging to Σ_c. For every term $f(c_1, \ldots, c_n)$ occurring as the left-hand side of a rule in $F_i^{\Sigma_c^e}(E)$, consider all the rules $f(c_1, \ldots, c_n) \to d_1, \ldots, f(c_1, \ldots, c_n) \to d_k$ in $F_i^{\Sigma_c^e}(E)$ with this left-hand side. Let d be the least element w.r.t. $>$ in $\{d_1, \ldots, d_k\}$. Add $f(c_1, \ldots, c_n) \to d$ and $d_j \to d$ for all j with $d_j \neq d$ to $R_{i+1}^{\Sigma_c^e}(E)$.

(b2) Let f be a binary function symbol belonging to Σ_c. For all pairs of constant symbols c_1, c_2 such that $f(c_1, c_2)$ or $f(c_2, c_1)$ is the left-hand side of a rule in $F_i^{\Sigma_c^e}(E)$, consider the set of constant symbols $\{d_1, \ldots, d_k\}$ occurring as right-hand sides of such rules, and let d be the least element w.r.t. $>$ in this set. Add $d_j \to d$ for all j with $d_j \neq d$ to $R_{i+1}^{\Sigma_c^e}(E)$. In addition, if $c_2 >_{lpo} c_1$, then add $f(c_1, c_2) \to d$ to $R_{i+1}^{\Sigma_c^e}(E)$, and otherwise $f(c_2, c_1) \to d$.

If at least one constant rule has been added in this step, then set $i := i + 1$ and continue with step (a). Otherwise, continue with step (c).

(c) For all $f \in \Sigma^e$, all pairs of distinct rules $f(c_1, \ldots, c_n) \to d, f(c_1', \ldots, c_n') \to d$ in $F_i^{\Sigma_c^e}(E)$, and all $i, 1 \leq i \leq n$ such that $c_i \neq c_i'$, add $c_i \to c_i'$ to $R_{i+1}^{\Sigma_c^e}(E)$ if $c_i > c_i'$ and otherwise add $c_i' \to c_i$ to $R_{i+1}^{\Sigma_c^e}(E)$. If at least one constant rule has been added in this step, then set $i := i + 1$ and continue with step (a). Otherwise, terminate with output $\widehat{R}^{\Sigma_c^e}(E) := R_{i+1}^{\Sigma_c^e}(E)$.

We illustrate the above construction using Example 2. In step (a), the non-trivial equivalence classes are $[c_{f(a,g(a))}] = \{c_{f(a,g(a))}, c\}$ with representative c and $[c_{g(b)}] = \{c_{g(a)}, c_{g(b)}, c_{h(a)}\}$ with representative $c_{h(a)}$. Thus, $c_{f(a,g(a))} \to c, c_{g(a)} \to c_{h(a)}, c_{g(b)} \to c_{h(a)}$ are the constant rules added to $R_1^{\Sigma_c^e}(E')$. The function rules in $F_0^{\Sigma_c^e}(E')$ are then $f(a, c_{h(a)}) \to c, g(a) \to c_{h(a)}, g(b) \to c_{h(a)}, h(a) \to c_{h(a)}$. Since these rules have unique left-hand sides, no constant rule is added in step (b). Consequently, we proceed with step (c). Since $g \in \Sigma^e$, the presence of the rules $g(a) \to c_{h(a)}$ and $g(b) \to c_{h(a)}$ triggers the addition of $a \to b$ to $R_1^{\Sigma_c^e}(E')$. The function rules in $R_1^{\Sigma_c^e}(E')$ are the ones in $F_0^{\Sigma_c^e}(E')$.

In the second iteration step, we now have the new non-trivial equivalence class $[a]_1 = \{a, b\}$ with representative b. The net effect of step (a) is, again, that the constant rules are moved unchanged from $R_1^{\Sigma_c^e}(E')$ to $R_2^{\Sigma_c^e}(E')$. The function rules in $F_1^{\Sigma_c^e}(E')$ are then $f(b, c_{h(a)}) \to c, g(b) \to c_{h(a)}, h(b) \to c_{h(a)}$. Consequently, no new constant rules are added in steps (b) and (c), and the construction terminates with output $\widehat{R}^{\Sigma_c^e}(E') = \{a \to b, c_{f(a,g(a))} \to c, c_{g(a)} \to c_{h(a)}, c_{g(b)} \to c_{h(a)}, f(b, c_{h(a)}) \to c, g(b) \to c_{h(a)}, h(b) \to c_{h(a)}\}$, which is identical to the system $\widehat{R}^{\Sigma_c}(E)$ computed for the set of identity E of Example 1.

Our goal is now to show that $\widehat{R}^{\Sigma_c^e}(E)$ provides us with a polynomial-time decision procedure for the extensional word problem in E, i.e., it allows us to decide the relation $\approx_E^{\Sigma_c^e}$. Let $R(\Sigma_c)$ and $>_{lpo}$ be defined as in (1).

Lemma 6. *The system $\widehat{R}^{\Sigma_c^e}(E)$ can be computed from $R(E)$ in polynomial time. Viewed as a set of identities, $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ is*

- *sound for commutative and extensional reasoning, i.e., for all rules $s \to t$ in $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ we have $s \approx_{R(E)}^{\Sigma_c^e} t$, and*
- *complete for commutative reasoning, i.e., or all terms $s, t \in G(\Sigma, C)$ we have that $s \approx_{R(E)}^{\Sigma_c} t$ implies $s \approx_{\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)} t$.*

Lemma 7. *Viewed as a term rewriting system, $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ is canonical, i.e., terminating and confluent.*

Intuitively, $\widehat{R}^{\Sigma_c^e}(E)$ extends $\widehat{R}^{\Sigma_c}(E)$ by additional rules relating constants that are equated due to extensionality. However, to keep the system confluent, we need to re-apply the other steps once two constants have been equated.

Lemma 8. *If $s, t \in G(\Sigma, C)$ have the same canonical forms w.r.t. $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$, then they also have the same canonical forms w.r.t. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$.*

We are now ready to prove our main technical result, from which decidability of the commutative and extensional word problem immediately follows.

Theorem 2. *Let $s, t \in G(\Sigma, C_0)$. Then we have $s \approx t \in CC^{\Sigma_c^e}(E)$ iff s and t have the same canonical form w.r.t. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$.*

Proof. Since the if-direction is easy to show, we concentrate here on the only-if-direction. If $s, t \in G(\Sigma, C_0)$ are such that $s \approx t \in CC^{\Sigma_c^e}(E)$, then there is a sequence of identities $s_1 \approx t_1, s_2 \approx t_2, \ldots, s_k \approx t_k$ such that $s_k = s, t_k = t$, and for all $i, 1 \leq i \leq k$, the identity $s_i \approx t_i$ belongs to E or can be derived from some of the identities $s_j \approx t_j$ with $j < i$ by apply reflexivity, transitivity, symmetry, congruence, commutativity, or extensionality. We prove that s and t

have the same canonical form w.r.t. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ by induction on the number of applications of the extensionality rule used when creating this sequence.

In the *base case*, no extensionality rule is used, and thus $s \approx t \in CC^{\Sigma_c}(E)$. By Theorem 1, s and t have the same canonical form w.r.t. $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$, and thus by Lemma 8 also w.r.t. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$.

In the *step case*, we consider the last identity $s_m \approx t_m$ obtained by an application of the extensionality rule. Then, by induction, we know that, for each $i, 1 \leq i < m$, the terms s_i and t_i have the same canonical form w.r.t. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$.

Now, consider the application of extensionality to an identity $s_\ell \approx t_\ell$ ($\ell < m$) that produced $s_m \approx t_m$. Thus, we have $s_\ell = f(g_1, \dots, g_n)$ and $t_\ell = f(h_1, \dots, h_n)$ for some n-ary function symbol $f \in \Sigma^e$, and extensionality generates the new identity $g_\mu \approx h_\mu$ for some $\mu, 1 \leq \mu \leq n$, such that $s_m = g_\mu$ and $t_m = h_\mu$. For $\nu = 1, \dots, n$, let g'_ν be the canonical form of g_ν w.r.t. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$, and h'_ν the canonical form of h_ν w.r.t. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$. We know that the canonical forms of s_ℓ and t_ℓ w.r.t. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ are identical, and these canonical forms can be obtained by normalizing $f(g'_1, \dots, g'_n)$ and $f(h'_1, \dots, h'_n)$. Since the rules of $R(\Sigma_c)$ are not applicable to these terms due to the fact that $f \notin \Sigma_c$, there are two possible cases for how the canonical forms of s_ℓ and t_ℓ can look like:

1. s_ℓ and t_ℓ respectively have the canonical forms $f(g'_1, \dots, g'_n)$ and $f(h'_1, \dots, h'_n)$, and thus the corresponding arguments are syntactically equal, i.e., $g'_\nu = h'_\nu$ for $\nu = 1, \dots, n$. In this case, the identity $s_m \approx t_m$ added by the application of the extensionality rule satisfies $s_m \approx_{\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)} t_m$ since we have $s_m = g_\mu \approx_{\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)} g'_\mu = h'_\mu \approx_{\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)} h_\mu = t_m$.
2. s_ℓ and t_ℓ reduce to the same constant d. Then $\widehat{R}^{\Sigma_c^e}(E)$ must contain the rules $f(g'_1, \dots, g'_n) \rightarrow d$ and $f(h'_1, \dots, h'_n) \rightarrow d$. By the construction of $\widehat{R}^{\Sigma_c^e}(E)$, we again have that $g'_\mu = h'_\mu$, i.e., the two terms are syntactically equal. In fact, otherwise a new constant rule $g'_\mu \rightarrow h'_\mu$ or $h'_\mu \rightarrow g'_\mu$ would have been added, and the construction would not have terminated yet. We thus have again $s_m = g_\mu \approx_{\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)} g'_\mu = h'_\mu \approx_{\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)} h_\mu = t_m$.

Summing up, we have seen that we have $s_i \approx_{\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)} t_i$ for all $i, 1 \leq i \leq m$. Since the identities $s_j \approx t_j$ for $m < j \leq k$ are generated from the identities $s_i \approx t_i$ for $i = 1, \dots, m$ and E using only reflexivity, transitivity, symmetry, commutativity, and congruence, this implies that also these identities satisfy $s_j \approx_{\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)} t_j$. In particular, we thus have $s_k \approx_{\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)} t_k$. Since $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ is canonical, this implies that $s_k = s$ and $t_k = t$ have the same canonical form w.r.t. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$. □

Recall that we have $f(h(a), b) \approx_{E'}^{\Sigma_c^e} c$ for the set of identities E' of Example 2. We have already seen that these two terms rewrite to the same canonical form w.r.t. $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c) = \widehat{R}^{\Sigma_c^e}(E') \cup R(\Sigma_c)$.

Again, it is easy to show that the decision procedure obtained by applying Theorem 2 requires only polynomial time.

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

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 rewriting-based approach actually deals with this problem. The answer is: in Case 1 of the case distinction in the proof of Theorem 2. In fact, there we consider a derived identity $s_\ell \approx t_\ell$ such that the (syntactically identical) canonical forms of $s_\ell = f(g_1, \ldots, g_n)$ and $t_\ell = f(h_1, \ldots, h_n)$ are not a constant from C, but of the form $f(g_1', \ldots, g_n') = f(h_1', \ldots, h_n')$. Basically, this means that s_ℓ and t_ℓ 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, s_ℓ, t_ℓ are "large" terms that potentially could cause a problem: an identity between them has been derived, and now extensionality applied to this identity yields a new identity $g_\mu \approx h_\mu$ between smaller terms. Our induction proof shows that this identity can nevertheless be derived from $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$, and thus does not cause a problem.

5 Symbols that Are Commutative and Extensional

In the previous section, we have made the assumptions that the sets Σ_c and Σ^e are disjoint, i.e., we did not consider extensionality for commutative symbols. The reason is that the presence of a commutative *and* extensional symbol would trivialize the equational theory. In fact, as already mentioned in the introduction, 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 $s \approx t$. 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 $E = \emptyset$.

In this section, we consider the following variant of extensionality for commutative function symbols f, which we call *c-extensionality*:

$$f(x_1, x_2) \approx f(y_1, y_2) \Rightarrow (x_1 \approx y_1 \wedge x_2 \approx y_2) \vee (x_1 \approx y_2 \wedge x_2 \approx y_1). \qquad (3)$$

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 ϕ, and let $p_1, \ldots, p_n$ be the propositional variables occurring in ϕ. 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 $f_\vee, f_\wedge, f_\neg, f$, and assume that f is commutative and satisfies (3). We then consider ground identities that axiomatize the truth tables for $\vee, \wedge, \neg$, i.e.,

$$\begin{array}{llll} f_\vee(0,0) \approx 0, & f_\vee(1,0) \approx 1, & f_\vee(0,1) \approx 1, & f_\vee(1,1) \approx 1, \\ f_\wedge(0,0) \approx 0, & f_\wedge(1,0) \approx 0, & f_\wedge(0,1) \approx 0, & f_\wedge(1,1) \approx 1, \\ f_\neg(0) \approx 1, & f_\neg(1) \approx 0. & & \end{array} \tag{4}$$

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_ϕ be the set of these ground identities, and let t_ϕ be the term obtained from ϕ by replacing the Boolean operations $\vee, \wedge$, and $\neg$ by the corresponding function symbols $f_\vee, f_\wedge$, and $f_\neg$.

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_ϕ together with* (3) *and commutativity of f iff ϕ is valid.*

To prove a *complexity upper bound* that matches the lower bound stated in Proposition 2, we consider a finite signature Σ, 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 c-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 our construction of $\widehat{R}^{\Sigma_c^e}(E)$. In steps (a) and (b), this procedure works as described in the previous section. For extensional symbols $f \in \Sigma^e \setminus \Sigma_c$, step (c) is also performed as in the previous section. For an extensional symbol $f \in \Sigma^e \cap \Sigma_c$, step (c) is modified as follows: for all pairs of distinct rules $f(c_1, c_2) \to d, f(c_1', c_2') \to d$ in $F_i^{\Sigma_c^e}(E)$, nondeterministically choose whether

- c_1 and c_1' as well as c_2 and c_2' are to be identified, or
- c_1 and c_2' as well as c_2 and c_1' are to be identified,

and then add the corresponding constant rules to $R_{i+1}^{\Sigma_c^e}(E)$ 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 (c). But each run r produces a rewrite system $\widehat{R}_r^{\Sigma_c^e}(E)$.

Example 3. We illustrate the nondeterministic construction using the identities E_ϕ for $\phi = p \vee \neg p$ from our coNP-hardness proof. Then E_ϕ 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

$$\begin{array}{llll} f_\vee(1,0) \rightarrow c_{f_\vee(1,0)}, & c_{f_\vee(1,0)} \rightarrow 1, & f_\vee(0,1) \rightarrow c_{f_\vee(0,1)}, & c_{f_\vee(0,1)} \rightarrow 1 \\ f_\neg(0) \rightarrow c_{f_\neg(0)}, & c_{f_\neg(0)} \rightarrow 1, & f_\neg(1) \rightarrow c_{f_\neg(1)}, & c_{f_\neg(1)} \rightarrow 0 \\ f(p,\overline{p}) \rightarrow c_{f(p,\overline{p})}, & f(1,0) \rightarrow c_{f(1,0)}, & c_{f(p,\overline{p})} \rightarrow c_{f(1,0)}. & \end{array}$$

In step (a) and (b) of the construction, these rules are transformed into the form

$$\begin{array}{llll} f_\vee(1,0) \rightarrow 1, & c_{f_\vee(1,0)} \rightarrow 1, & f_\vee(0,1) \rightarrow 1, & c_{f_\vee(0,1)} \rightarrow 1 \\ f_\neg(0) \rightarrow 1, & c_{f_\neg(0)} \rightarrow 1, & f_\neg(1) \rightarrow 0, & c_{f_\neg(1)} \rightarrow 0 \\ f(p,\overline{p}) \rightarrow c_{f(1,0)}, & f(1,0) \rightarrow c_{f(1,0)}, & c_{f(p,\overline{p})} \rightarrow c_{f(1,0)}. & \end{array} \tag{5}$$

Since no new constant rule is added, the construction proceeds with step (c). 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 $f \in \Sigma_c \cap \Sigma^e$, 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 $\widehat{R}^{\Sigma_c^e}_{r_1}(E)$ and $\widehat{R}^{\Sigma_c^e}_{r_2}(E)$ share the rules in (5), but the first contains $p \rightarrow 1$ whereas the second contains $p \rightarrow 0$.

Coming back to the general case, as in the proofs of Lemma 6 and Lemma 7, we can show the following for the rewrite systems $\widehat{R}^{\Sigma_c^e}_r(E)$.

Lemma 10. *For every run r, the term rewriting system $\widehat{R}^{\Sigma_c^e}_r(E)$ is produced in polynomial time, and the system $\widehat{R}^{\Sigma_c^e}_r(E) \cup R(\Sigma_c)$ is canonical.*

Using the canonical rewrite systems $\widehat{R}^{\Sigma_c^e}_r(E) \cup R(\Sigma_c)$, we can now characterize when an identity follows from E w.r.t. commutativity of the symbols in Σ_c, 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. $\widehat{R}^{\Sigma_c^e}_r(E) \cup R(\Sigma_c)$ for every run r of the nondeterministic construction.*

The main ideas for how to deal with extensionality and c-extensionality in the proof of this theorem are very similar to how extensionality was dealt with in the proof of Theorem 2. As for all the other results stated without proof here, a detailed proof can be found in [2]. Together with Proposition 2, Theorem 3 yields the following complexity results.

Corollary 3. *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.*

Coming back to Example 3, we note that $\phi = p \vee \neg p$ is valid, and thus (by Lemma 9), the identity $f_\vee(p, f_\neg(p)) \approx 1$ holds in all algebra that satisfy E_ϕ 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: $f_\vee(p, f_\neg(p)) \to f_\vee(1, f_\neg(p)) \to f_\vee(1, f_\neg(1)) \to f_\vee(1, 0) \to 1$. For the run r_2, we obtain the sequence $f_\vee(p, f_\neg(p)) \to f_\vee(0, f_\neg(p)) \to f_\vee(0, f_\neg(0)) \to f_\vee(0, 1) \to 1$. Thus, for both runs the terms $f_\vee(p, f_\neg(p))$ and 1 have the same canonical form 1.

6 Conclusion

We have shown, using a rewriting-based approach, that adding commutativity and extensionality of 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 $O(n \log n)$ [12,13]. Since this complexity upper bound can also be achieved using a rewriting-based approach [8,16], 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 extensionality, as in Sect. 4, but this question was not in the focus here.

The rules specifying extensionality are simple kinds of Horn rules whose atoms are identities. The question arises which other such Horn rules can be added without increasing the complexity of the word problem. It is known that allowing for associative-commutative (AC) symbols leaves the word problem for finite sets of ground identities decidable [4,11]. It would be interesting to see what happens if additionally (non-AC) extensional symbols are added. The approaches employed in [4,11] are rewriting-based, but in contrast to our treatment of commutativity, they use rewriting modulo AC. It is thus not clear whether the approach developed in the present paper can be adapted to deal with AC symbols.

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 [9] and [14].

Acknowledgements. The authors would like to thank the reviewers for their careful reading of the paper and their useful comments, which helped to improve the presentation of the paper. The first author thanks Barbara Morawska for helpful discussions.

References

1. Baader, F., et al.: An Introduction to Description Logic. Cambridge University Press, Cambridge (2017)
2. Baader, F., Kapur, D.: Deciding the word problem for ground identities with commutative and extensional symbols. LTCS-Report 20-02, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Dresden (2020). https://tu-dresden.de/inf/lat/reports#BaKa-LTCS-20-02
3. Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge University Press, Cambridge (1998)
4. Bachmair, L., Ramakrishnan, I.V., Tiwari, A., Vigneron, L.: Congruence closure modulo associativity and commutativity. In: Kirchner, H., Ringeissen, C. (eds.) FroCoS 2000. LNCS (LNAI), vol. 1794, pp. 245–259. Springer, Heidelberg (2000). https://doi.org/10.1007/10720084_16
5. Downey, P., Sethi, R., Tarjan, R.E.: Variations on the common subexpression problem. J. ACM **27**(4), 758–771 (1980)
6. Gallier, J.: An algorithm for finding canonical sets of ground rewrite rules in polynomial time. J. ACM **40**(1), 1–16 (1993)
7. Kapur, D.: Shostak's congruence closure as completion. In: Comon, H. (ed.) RTA 1997. LNCS, vol. 1232, pp. 23–37. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-62950-5_59
8. Kapur, D.: Conditional congruence closure over uninterpreted and interpreted symbols. J. Syst. Sci. Complex. **32**(1), 317–355 (2019). https://doi.org/10.1007/s11424-019-8377-8
9. Käufl, T., Zabel, N.: The theorem prover of the program verifier *Tatzelwurm*. In: Stickel, M.E. (ed.) CADE 1990. LNCS, vol. 449, pp. 657–658. Springer, Heidelberg (1990). https://doi.org/10.1007/3-540-52885-7_128
10. Kozen, D.: Complexity of finitely presented algebras. In: Proceedings of the 9th ACM Symposium on Theory of Computing, pp. 164–177. ACM (1977)
11. Narendran, P., Rusinowitch, M.: Any ground associative-commutative theory has a finite canonical system. In: Book, R.V. (ed.) RTA 1991. LNCS, vol. 488, pp. 423–434. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-53904-2_115
12. Nelson, G., Oppen, D.C.: Fast decision procedures based on congruence closure. J. ACM **27**(4), 356–364 (1980)
13. Nieuwenhuis, R., Oliveras, A.: Fast congruence closure and extensions. Inf. Comput. **205**(4), 557–580 (2007)
14. Schmidt, R.A., Waldmann, U.: Modal Tableau systems with blocking and congruence closure. In: De Nivelle, H. (ed.) TABLEAUX 2015. LNCS (LNAI), vol. 9323, pp. 38–53. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24312-2_4
15. Shostak, R.E.: An algorithm for reasoning about equality. Commun. ACM **21**(7), 583–585 (1978)

16. 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)
17. 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)
18. Wechler, W.: Universal Algebra for Computer Scientists. EATCS Monographs on Theoretical Computer Science, vol. 25. Springer, Cham (1992). https://doi.org/10.1007/978-3-642-76771-5