Technische Universität Dresden
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS–Report

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

Franz Baader Deepak Kapur

LTCS-Report 20-02

Postal Address:
Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
01062 Dresden

http://lat.inf.tu-dresden.de

Visiting Address:
Nöthnitzer Str. 46
Dresden

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, Germany
franz.baader@tu-dresden.de
[2] Dept. of Computer Science, University of New Mexico, USA
kapur@cs.unm.edu

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. 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.

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* [8,4,10,2]. 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 [12], and made explicit in [6] (see also [5,14] for other rewriting-based approaches).

In the motivating example from DL, but also in other settings where congruence closure is employed (such as SMT [15,11]), it sometimes makes sense to assume that certain function

symbols satisfy additional properties that are not expressible by 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 well-known that adding commutativity does not increase the complexity (see, e.g., [4,7]), 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 [6] can be extended such that it can also handle commutative symbols. In contrast to approaches that deal with associative-commutative (AC) symbols [9,3] 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 Section 3. In this section, we do not consider symbols that are both commutative and extensional since extensionality as defined until now is not appropriate for commutative symbols: if $f(a, b)$ is equal to $f(a', b')$ for an extensional and commutative symbol, then this implies that all four constants a, b, a', b' are actually equal. In Section 4, we introduce the notion of d-extensionality, which is more appropriate for commutative symbols. Whereas the approaches developed in Sections 2 and 3 yield polynomial time decision procedures for the word problem, d-extensionality makes the word problem coNP-complete.

We assume that the reader is familiar with basic notions and results regarding equational theories, universal algebra, and term rewriting, as they can, e.g., be found in [2]. We will keep as close as possible to the notation introduced in [2]. In particular, we use $\approx$ to denote identities between terms and $=$ to denote syntactic equality.

2 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 [2].

It is well know 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 $f \in \Sigma$ 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, 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., [2], Section 4.3).

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 show 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 [2]). 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 $\approx_E^{\Sigma_c}$ [4].

Here, we follow a different approach, which is based on rewriting [6,7]. 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 $s_0 \approx_{R(E)}^{\Sigma_c} t_0$.*

Proof. To show the *only-if-direction*, it is sufficient to prove that $E \subseteq \approx_{R(E)}$. Thus, consider $s \approx t \in E$. Then we have $s \approx_{R(E)} c_s$, $t \approx_{R(E)} c_t$, and $c_s \approx_{R(E)} c_t$. The former two identities hold due to Lemma 1, and the latter identity holds since $c_s \to c_t$ or $c_t \to c_s$ belongs to $R(E)$. Clearly, put together the three identities imply $s \approx_{R(E)} t$.

To show the *if-direction*, it is sufficient to prove that the following holds for all terms $u, v \in G(\Sigma, C)$:

$$u \approx_{R(E)}^{\Sigma_c} v \text{ implies } \widehat{u} \approx_E^{\Sigma_c} \widehat{v}. \tag{1}$$

In fact, (1) immediately yields the if-direction of the lemma since, for terms $s_0, t_0 \in G(\Sigma, C_0)$, we have $\widehat{s_0} \approx s_0$ and $\widehat{t_0} \approx t_0$.

Since $\approx_E^{\Sigma_c}$ is transitive and symmetric, it is sufficient to prove (1) for the case where v is obtained from u by applying one of the rules of $R(E)$ or commutativity to u.

First, assume that the function rule $f(c_{s_1}, \ldots, c_{s_n}) \to c_s \in R(E)$ is applied. Since this rule belongs to $R(E)$ only if $s = f(s_1, \ldots, s_n)$, we actually have the syntactic equality $\widehat{u} = \widehat{v}$, and thus $\widehat{u} \approx_E^{\Sigma_c} \widehat{v}$ since $\approx_E^{\Sigma_c}$ is reflexive.

Second, assume that the constant rule $c_s \to c_t \in R(E)$ is applied. Since this rule belongs to $R(E)$ only if $s \approx t \in E$, the replacement of c_s with c_t when going from u to v can be mirrored by replacing s with t when going from $\widehat{u}$ to $\widehat{v}$, which shows $\widehat{u} \approx_E^{\Sigma_c} \widehat{v}$.

Third, assume that commutativity for a function symbol in Σ_c is applied. Since commutativity of this symbol is available both in $\approx_E^{\Sigma_c}$ and in $\approx_{R(E)}^{\Sigma_c}$, its application when going from u to v can be mirrored by an application of commutativity that transforms $\widehat{u}$ into $\widehat{v}$. To be more precise, assume that $f \in \Sigma_c$ and that u is transformed into v by replacing a subterm of the form $f(g, h)$ with $f(h, g)$. Then $\widehat{u}$ contains the subterm $f(\widehat{g}, \widehat{h})$, which can be replaced by an application of commutativity with $f(\widehat{h}, \widehat{g})$, thus yielding $\widehat{v}$. □

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

$$R(\Sigma_c) := \{f(s,t) \to f(t,s) \mid s,t \in G(\Sigma, C) \text{ and } s >_{lpo} t\},^3 \tag{2}$$

where $>_{lpo}$ denotes the lexicographic path order (see Section 5.4.2 in [2]) 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 [2]). 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. 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 $R_{i+1}^{\Sigma_c}(E)$.

(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 add $f(c_2, c_1) \to d$.

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

[3] 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 $>_{lpo}$.

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. First, we prove that the construction of this system takes only polynomial time.

Lemma 3. *The system $\widehat{R}^{\Sigma_c}(E)$ can be computed from $R(E)$ in polynomial time.*

Proof. First, note that step (a) can clearly be performed in polynomial time since deciding $\approx_{R_i^{\Sigma_c}(E)|_{con}}$ amounts to performing reachability tests in an undirected graph (more efficiently, one can maintain the equivalence classes $[c]_i$ and the representatives w.r.t. $R_i^{\Sigma_c}(E)$ and $>$ by building and updating a union-find data structure, as e.g. described in Section 4.4 of [2]). Producing the rules in $F_i^{\Sigma_c}(E)$ and grouping them according to their left-hand sides in step (b) is clearly also possible in polynomial time, as is adding the new rules to $R_{i+1}^{\Sigma_c}(E)$. In case the procedure does not terminate in this step, 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 $\widehat{R}^{\Sigma_c}(E)$ is correct in the following sense:

Lemma 4. *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 $s \approx^{\Sigma_c}_{R(E)} t$ iff $s \approx_{\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)} t$.*

Proof. First, note that $\approx^{\Sigma_c}_{R(E)} = \approx^{\Sigma_c}_{R_0^{\Sigma_c}(E)} = \approx_{R_0^{\Sigma_c}(E) \cup R(\Sigma_c)}$ since $R(E) = R_0^{\Sigma_c}(E)$ and $R(\Sigma_c)$ realizes commutativity of the symbols in Σ_c. It is thus sufficient to show that the modifications performed when going from $R_i^{\Sigma_c}(E)$ to $R_{i+1}^{\Sigma_c}(E)$ do not change the induced equational theory, i.e., $\approx_{R_i^{\Sigma_c}(E) \cup R(\Sigma_c)} = \approx_{R_{i+1}^{\Sigma_c}(E) \cup R(\Sigma_c)}$ for all $i \geq 0$ for which $R_{i+1}^{\Sigma_c}(E)$ is defined.

To show the inclusion from left to right, first consider step (a). If $R_i^{\Sigma_c}(E)$ contains the constant rule $c_1 \to c_2$, then c_1 and c_2 belong to the same equivalence class w.r.t. $R_i^{\Sigma_c}(E)|_{con}$ and $c_1 > c_2$. In case c_2 is the least element in this class, then $R_{i+1}^{\Sigma_c}(E)$ still contains the rule $c_1 \to c_2$. Otherwise, since we know that $c_1 > c_2$, the least element e in the class is different from these two constants, and thus $R_{i+1}^{\Sigma_c}(E)$ contains the rules $c_1 \to e, c_2 \to e$, which shows $c_1 \approx_{R_{i+1}^{\Sigma_c}(E)} c_2$, and thus $c_1 \approx_{R_{i+1}^{\Sigma_c}(E) \cup R(\Sigma_c)} c_2$.

Regarding step (b1), note that the replacements performed in the construction of $F_i^{\Sigma_c}(E)$ replace constants c occurring in function rules by constants e that are equivalent to c both w.r.t. $R_i^{\Sigma_c}(E)$ and w.r.t. $R_{i+1}^{\Sigma_c}(E)$. Thus, these replacements do not change the overall equational theory. Now, consider a function rule $f(c_1, \ldots, c_n) \to d_j$ in $F_i^{\Sigma_c}(E)$. Either this rule also belongs to $R_{i+1}^{\Sigma_c}(E)$, or $R_{i+1}^{\Sigma_c}(E)$ contains rules $f(c_1, \ldots, c_n) \to d, d_j \to d$. In both cases, we have $f(c_1, \ldots, c_n) \approx_{R_{i+1}^{\Sigma_c}(E)} d_j$, and thus $f(c_1, \ldots, c_n) \approx_{R_{i+1}^{\Sigma_c}(E) \cup R(\Sigma_c)} d_j$.

In step (b2), a function rule of the form $f(c_1, c_2) \to d_j$ for $f \in \Sigma_c$ may be removed from $R_i^{\Sigma_c}(E)$, but then $d_j \to d$ and either $f(c_1, c_2) \to d$ or $f(c_2, c_1) \to d$ belong to $R_{i+1}^{\Sigma_c}(E)$. Clearly, this implies $f(c_1, c_2) \approx_{R_{i+1}^{\Sigma_c}(E) \cup R(\Sigma_c)} d_j$ since $f(c_1, c_2) \approx_{R(\Sigma_c)} f(c_2, c_1)$.

The inclusion from right to left can be shown similarly to the one from left to right. □

Lemma 5. *Viewed as a term rewriting system, $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$ 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 $R(E) \cup R(\Sigma_c)$, by using the reduction order $>_{lpo}$ introduced in the definition of $R(\Sigma_c)$.

Regarding confluence, first note that there are no non-trivial critical pairs (see Section 6.2 in [2]) between the rules in $\widehat{R}^{\Sigma_c}(E)$. To see this, first note that two function rules from $\widehat{R}^{\Sigma_c}(E)$ cannot overlap due to the fact that in step (b) no more constants are identified, and thus all left-hand sides of function rules in $\widehat{R}^{\Sigma_c}(E)$ 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 $f(c_1, \ldots, c_n) \to d$ cannot overlap with a rule of the form $e \to e'$ since the c_i are representatives, whereas e is not a representative.

Second, consider an overlap of a rule $f(s,t) \to f(t,s)$ in $R(\Sigma_c)$ with a rule in $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$. If this overlap occurs at the root position, then the other rule is a rule of the form $f(c_1, c_2) \to d$ in $\widehat{R}^{\Sigma_c}(E)$. But then we must have $s = c_1$ and $t = c_2$. This cannot be the case since we know that $s >_{lpo} t$ by the definition of $R(\Sigma_c)$, but also $c_2 >_{lpo} c_1$ or $c_1 = c_2$ by the construction of $\widehat{R}^{\Sigma_c}(E)$.

Finally, assume that the rule from $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$ is applied within s or t. In the latter case, we have $f(s,t) \to_{R(\Sigma_c)} f(t,s)$ and $f(s,t) \to_{\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)} f(s,t')$. Since $s >_{lpo} t >_{lpo} t'$, we can close this fork by the following rewrite steps: $f(s,t') \to_{R(\Sigma_c)} f(t',s)$ and $f(t,s) \to_{\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)} f(t',s)$.

Now, assume that the rewriting is performed within s. Then we have the fork $f(s,t) \to_{R(\Sigma_c)} f(t,s)$ and $f(s,t) \to_{\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)} f(s',t)$.

- If $s' >_{lpo} t$, then we can close this fork by the following rewrite steps: $f(s',t) \to_{R(\Sigma_c)} f(t,s')$ and $f(t,s) \to_{\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)} f(t,s')$.
- If $s' = t$, then we have $f(s',t) = f(t,t) = f(t,s')$ and $f(t,s) \to_{\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)} f(t,s')$.
- If $t >_{lpo} s'$, then we can close the fork by leaving $f(s',t)$ as it is, and rewriting

$$f(t,s) \to_{\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)} f(t,s') \to_{R(\Sigma_c)} f(s',t).$$

This shows that all non-trivial critical pairs of $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$ can be joined, which proves confluence of $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$. □

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. $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$.*

Proof. If $s_0 \approx t_0 \in CC^{\Sigma_c}(E)$, then $s_0 \approx_E^{\Sigma_c} t_0$, and thus $s_0 \approx_{\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)} t_0$ by Lemma 2 and Lemma 4. Since $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$ is canonical, this implies that s_0 and t_0 have the same canonical form w.r.t. $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$.

Conversely, if s_0 and t_0 have the same canonical form w.r.t. $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$, then we have $s_0 \approx_{\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)} t_0$. Lemma 4 yields $s_0 \approx_{R(E)}^{\Sigma_c} t_0$. Since $s_0, t_0 \in G(\Sigma, C_0)$, we can now apply Lemma 2 to obtain $s_0 \approx_E^{\Sigma_c} t_0$, which is equivalent to $s_0 \approx t_0 \in CC^{\Sigma_c}(E)$. □

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

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.

In this section, it remains to show that the decision procedure obtained by applying Theorem 1 requires only polynomial time.

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.*

Proof. Since we already know that $\widehat{R}^{\Sigma_c}(E)$ 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 $s_0 \in G(\Sigma, C_0)$ can be computed in time polynomial in the size of s_0 and $\widehat{R}^{\Sigma_c}(E)$. This is clearly the case if only a polynomial number of rewrite steps are needed to produce the canonical form of s_0.

For every constant $c \in C$, we define its *rank* to be the cardinality of the set $\{d \in C \mid c > d\}$. Since the cardinality of C is linearly bounded by the size of E and C_0, this is also true for the rank of each constant. In fact, the rank of a constant in C is at most $|C| - 1$. We define the *rank of a function symbol* in Σ to be $|C|$. Given a term $s \in G(\Sigma, C)$, its *rank* is the sum of the ranks of the symbols occurring in s. Clearly, this rank is polynomial in the size of s and E.

We claim that the rank of terms is decreased by an application of a rule in $\widehat{R}^{\Sigma_c}(E)$. In fact, if a rule of the form $f(c_1, \ldots, c_n) \to d$ is applied, then at least $|C[$ is subtracted from the rank due to the removal of f, and at most $|C| - 1$ is added due to the addition of d. If a rule of the form $d \to d'$ is applied, then we know that $d > d'$, and thus the rank of d is larger than the rank of d'. Application of a rule from $R(\Sigma_c)$ does not change the rank of a term. In addition, if we use innermost rewriting to compute the canonical form, the rules from $R(\Sigma_c)$ are applied at most once to every occurrence of a commutative function symbol. □

3 Commutative congruence closure with extensionality

Here, we additionally assume that some of the *non-commutative*[4] 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 commutativity for the symbols in Σ_c, we now assume that also the following conditional identities are satisfied for every n-ary function symbol $f \in \Sigma^e$:

$$f(x_1, \ldots, x_n) \approx f(y_1, \ldots, y_n) \Rightarrow x_i \approx y_i \quad \text{for all } i, 1 \leq i \leq n. \tag{3}$$

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

[4] We will explain in the next section why the notion of extensionality introduced in (3) below is not appropriate for commutative symbols.

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 [16], 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 Section 2.

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 6. *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 $s_0 \approx_{R(E)}^{\Sigma_c^e} t_0$.*

Proof. The only-if-direction is an easy consequence of the fact that $E \subseteq \approx_{R(E)}$ (see the proof of Lemma 2). In fact, $s_0 \approx_E^{\Sigma_c^e} t_0$ implies $s_0 \approx t_0 \in CC^{\Sigma_c^e}(E)$ by Proposition 1, and thus there is a sequence of identities $s_1 \approx t_1, s_2 \approx t_2, \ldots, s_k \approx t_k$ such that $s_k = s_0, t_k = t_0$, 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. Using the fact that $E \subseteq \approx_{R(E)}$, we can replace identities $s_i \approx t_i \in E$ in this sequence by a derivation of $s_i \approx t_i$ using identities in $R(E)$ as well as applications of reflexivity, transitivity, symmetry, and congruence. This shows that we have $s_0 \approx t_0 \in CC^{\Sigma_c^e}(R(E))$, and thus $s_0 \approx_{R(E)}^{\Sigma_c^e} t_0$.

To show the if-direction, it is again sufficient to prove that the following holds for all terms $u, v \in G(\Sigma, C)$:

$$u \approx_{R(E)}^{\Sigma_c^e} v \text{ implies } \widehat{u} \approx_E^{\Sigma_c^e} \widehat{v}. \tag{4}$$

Again, (4) immediately yields the if-direction of the lemma since, for terms $s_0, t_0 \in G(\Sigma, C_0)$, we have $\widehat{s_0} = s_0$ and $\widehat{t_0} = t_0$.

To show that (4) holds, assume that $u \approx^{\Sigma_c^e}_{R(E)} v$. Then $u \approx v \in CC^{\Sigma_c^e}(R(E))$, and thus there is a sequence of identities $u_1 \approx v_1, u_2 \approx v_2, \ldots, u_k \approx v_k$ such that $u_k = u, v_k = v$, and for all $i, 1 \leq i \leq k$, the identity $u_i \approx v_i$ belongs to $R(E)$ or can be derived from some of the identities $u_j \approx v_j$ with $j < i$ by apply reflexivity, transitivity, symmetry, congruence, commutativity, or extensionality.

Now, consider the corresponding sequence $\widehat{u}_1 \approx \widehat{v}_1, \widehat{u}_2 \approx \widehat{v}_2, \ldots, \widehat{u}_k \approx \widehat{v}_k$, and note that $\widehat{u}_k = \widehat{u}, \widehat{v}_k = \widehat{v}$. Using (1) we can replace identities $\widehat{u}_i \approx \widehat{v}_i$ for $u_i \approx v_i \in R(E)$ with their derivation from E. Application of reflexivity, transitivity, symmetry, commutativity, and congruence in the original sequence can clearly be mimicked in the new sequence. The same is true for extensionality: in fact, if $u_i \approx v_i$ is obtained by applying extensionality, then there is an identity $u_j \approx v_j$, where $j < i$, $u_j = f(g_1, \ldots, g_n), v_j = f(h_1, \ldots, h_n)$ for $f \in \Sigma^e$, and $u_i = g_\ell, v_i = h_\ell$ for some $\ell, 1 \leq \ell \leq n$. Since we have $\widehat{u}_j = f(\widehat{g}_1, \ldots, \widehat{g}_n)$ and $\widehat{v}_j = f(\widehat{h}_1, \ldots, \widehat{h}_n)$, extensionality can be used to derive $\widehat{u}_i \approx \widehat{v}_i$ from $\widehat{u}_j \approx \widehat{v}_j$. Thus, we have shown that $\widehat{u} \approx \widehat{v} \in CC^{\Sigma_c^e}(E)$, which completes the proof of (4), and thus of the lemma. □

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 add $f(c_2, c_1) \to d$.

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 $R_1^{\Sigma_c^e}(E')$.

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 identities 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 $\approx_E^{\Sigma_c^e}$. To this purposes, we first show a sequence of lemmas whose proofs are very similar to the proofs of the corresponding lemmas in the previous section.

Lemma 7. *The system $\widehat{R}^{\Sigma_c^e}(E)$ can be computed from $R(E)$ in polynomial time.*

Proof. The proof of this lemma is basically identical to the proof of Lemma 3. The only additional observations to be made are that a single step (c) can be performed in polynomial time, and that also in step (c) the number of different equivalence classes of constants decreases by at least one if the procedure does not terminate in this step. □

Let $R(\Sigma_c)$ and $>_{lpo}$ be defined as in (2).

Lemma 8. *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., for all terms $s, t \in G(\Sigma, C)$ we have that $s \approx_{R(E)}^{\Sigma_c} t$ implies $s \approx_{\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)} t$.*

Proof. Regarding soundness for commutative and extensional reasoning, it is easy to show by induction on the number of applied steps that all rules $s \to t$ generated in steps (a), (b), and (c) satisfy $s \approx_E^{\Sigma_c^e} t$. For step (a) and step (b), this can be shown as in the proof of Lemma 4.

Thus, consider step (c). If $f(c_1, \ldots, c_n) \to d$ and $f(c'_1, \ldots, c'_n) \to d$ belong to $R_i^{\Sigma_c^e}(E)$, then by induction we obtain $f(c_1, \ldots, c_n) \approx_E^{\Sigma_c^e} d$ and $f(c'_1, \ldots, c'_n) \approx_E^{\Sigma_c^e} d$, and thus $f(c_1, \ldots, c_n) \approx_E^{\Sigma_c^e} f(c'_1, \ldots, c'_n)$. Finally, extensionality yields $c_i \approx_E^{\Sigma_c^e} c'_i$ for $i = 1, \ldots, n$.

Regarding completeness for commutative reasoning, step (a) and step (b) can be treated as in the proof of Lemma 4. Since, in step (c), none of the existing rules are deleted or changed, it trivially preserves completeness. □

Lemma 9. *Viewed as a term rewriting system, $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ is canonical, i.e., terminating and confluent.*

Proof. Termination and confluence of the term rewriting system $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ can be shown as in the proof of Lemma 3. □

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. The canonical forms generated by $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ and $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$ need not coincide, but due to the fact that $\widehat{R}^{\Sigma_c^e}(E)$ extends $\widehat{R}^{\Sigma_c}(E)$ in the way just described, they are related as follows.

Lemma 10. *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. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$.*

Proof. If the terms s, t have the same canonical forms w.r.t. the rewrite system $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$, then we have $s \approx_{\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)} t$, and thus $s \approx^{\Sigma_c}_{R(E)} t$ by Lemma 4. Completeness of $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ for commutative reasoning then yields $s \approx_{\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)} t$, and hence s, t have the same canonical forms w.r.t. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ since this rewrite system is canonical. □

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. To show the if-direction, assume that s and t have the same canonical form w.r.t. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$. Then, $s \approx_{\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)} t$, and thus soundness of $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ for commutative and extensional reasoning (Lemma 8) yields $s \approx^{\Sigma_c^e}_{R(E)} t$. Since $s, t \in G(\Sigma, C_0)$, this implies $s \approx^{\Sigma_c^e}_{E} t$ by Lemma 6, and thus $s \approx t \in CC^{\Sigma_c^e}(E)$.

To prove the only-if-direction, assume that $s, t \in G(\Sigma, C_0)$ 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 Lemma 10 yields that they also have the same canonical form w.r.t. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$.

In the step case, we consider the last application of the extensionality rule at $s_\ell \approx t_\ell$. Then, by induction, we know that, for each $i, 1 \leq i \leq \ell$, 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 $s_\ell \approx t_\ell$. We have $s_\ell = f(g_1, \ldots, g_n)$ and $t_\ell = f(h_1, \ldots, h_n)$ for some n-ary function symbol $f \in \Sigma^e$, and extensionality generates the new identity $g_i \approx h_i$, i.e., $s_{\ell+1} = g_i$ and $t_{\ell+1} = h_i$. For $j = 1, \ldots, n$, let g'_j be the canonical form of g_j w.r.t. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$, and h'_j the canonical form of h_j 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 $f(g'_1, \ldots, g'_n)$ and $f(h'_1, \ldots, h'_n)$. 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 $f(g'_1, \ldots, g'_n)$ and $f(h'_1, \ldots, h'_n)$, and thus the corresponding arguments are syntactically equal, i.e., $g'_j = h'_j$ for $j = 1, \ldots, n$. In this

case, the identity $s_{\ell+1} \approx t_{\ell+1}$ added by the application of the extensionality rule satisfies $s_{\ell+1} \approx_{\widehat{R}^{\Sigma_c^e}(E)\cup R(\Sigma_c)} t_{\ell+1}$ since we have $s_{\ell+1} = g_i \approx_{\widehat{R}^{\Sigma_c^e}(E)\cup R(\Sigma_c)} g'_i = h'_i \approx_{\widehat{R}^{\Sigma_c^e}(E)\cup R(\Sigma_c)} h_i = t_{\ell+1}$.

2. s_ℓ and t_ℓ reduce to the same constant d. Then $\widehat{R}^{\Sigma_c^e}(E)$ must contain rules $f(g'_1, \ldots, g'_n) \to d$ and $f(h'_1, \ldots, h'_n) \to d$. By the construction of $\widehat{R}^{\Sigma_c^e}(E)$, we again have that $g'_i = h'_i$, i.e., the two terms are syntactically equal. In fact, otherwise a new constant rule $g'_i \to h'_i$ or $h'_i \to g'_i$ would have been added, and the construction would not have terminated yet. We thus have again $s_{\ell+1} = g_i \approx_{\widehat{R}^{\Sigma_c^e}(E)\cup R(\Sigma_c)} g'_i = h'_i \approx_{\widehat{R}^{\Sigma_c^e}(E)\cup R(\Sigma_c)} h_i = t_{\ell+1}$.

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 $i, 1 \leq i \leq \ell+1$. Since the identities $s_j \approx t_j$ for $\ell+1 < j \leq k$ are generated from the identities $s_i \approx t_i$ for $i = 1, \ldots, \ell+1$ 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 remains 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.*

Proof. Again, it is sufficient to show that computing canonical forms w.r.t. $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ can be done in polynomial time. Since the rules of $\widehat{R}^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ have the same shape as the rules of $\widehat{R}^{\Sigma_c}(E) \cup R(\Sigma_c)$, the proof of this fact is analogous to the proof of Corollary 1. □

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 $g_i \approx h_i$ 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.

4 Symbols that are commutativity and extensional

In the previous section, we have made the assumption that the sets Σ_c and Σ^e are disjoint, i.e., we did not consider extensionality for commutative symbols. While our approach could easily be extended to deal with symbols that are both commutative and extensional, we have not done so since, for such symbols, we would obtain more consequences than we bargained for.

For example, if $f(a, b)$ is assumed to name the couple consisting of the individuals a, b, then it is reasonable to assume that such a couple-building operator is commutative, i.e., it is irrelevant in which order the elements building the couple are written. In addition, in this setting one also wants a form of extensionality: if two couples are supposed to be the same, then they must consist of the same two individuals. However, requiring the extensionality rule (3) for the symbol f is too strong here since, together with commutativity, it would imply that all individuals participating in two couples that are equal are identical. In fact, assume that the ground identity $f(a, b) \approx f(c, d)$ has been derived. Then extensionality yields the identities $a \approx c$ and $b \approx d$. Additionally, commutativity can be used to derive the identity $f(a, b) \approx f(d, c)$, which in turn yields the identities $a \approx d$ and $b \approx c$ by extensionality. Consequently, the four constants are identified.

Thus, it is more reasonable to require the following variant of extensionality for commutative function symbols f, which we call *d-extensionality* (where "d" stands for "disjunctive"):

$$f(x_1, x_2) \approx f(y_1, y_2) \Rightarrow (x_1 \approx y_1 \wedge x_2 \approx y_2) \vee (x_1 \approx y_2 \wedge x_2 \approx y_1). \tag{5}$$

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 d-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 (5). We then consider ground identities that axiomatize the truth tables for $\vee, \wedge, \neg$, i.e.,

$$\begin{array}{llll} f_\vee(0,0) \approx 0, & f_\vee(1,0) \approx 1, & f_\vee(0,1) \approx 1, & f_\vee(1,1) \approx 1, \\ f_\wedge(0,0) \approx 0, & f_\wedge(1,0) \approx 0, & f_\wedge(0,1) \approx 0, & f_\wedge(1,1) \approx 1, \\ f_\neg(0) \approx 1, & f_\neg(1) \approx 0. & & \end{array} \tag{6}$$

In addition, we consider, for every $i, 1 \leq i \leq n$, the identity

$$f(p_i, \overline{p}_i) \approx f(0, 1).$$

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 11. *The identity $t_\phi \approx 1$ holds in every algebra satisfying E_ϕ together with* (5) *and commutativity of f iff ϕ is valid.*

Proof. First assume that ϕ is valid, and let $\mathcal{A}$ be an algebra satisfying E_ϕ together with (5) and commutativity of f. Using the fact that $\mathcal{A}$ satisfies the identities $f(p_i, \overline{p}_i) \approx f(0,1)$ for $i = 1, \ldots, n$, as well as (5), we can deduce that, for all $i = 1, \ldots, n$, we have $p_i^{\mathcal{A}} \in \{0^{\mathcal{A}}, 1^{\mathcal{A}}\}$. Let v be the propositional valuation that satisfies $v(p_i) = 1$ iff $p_i^{\mathcal{A}} = 1^{\mathcal{A}}$. Since ϕ is valid, we know that $v(\phi) = 1$. Using this fact and the identities axiomatizing the truth tables for $\vee, \wedge, \neg$, we obtain that $t_\phi^{\mathcal{A}} = 1^{\mathcal{A}}$. Thus, the identity $t_\phi \approx 1$ holds in $\mathcal{A}$.

Conversely, assume that ϕ is not valid, and let v be a propositional valuation such that $v(\phi) = 0$. Let $\mathcal{C}$ be the free commutative algebra with generators $0, 1$ and the binary function symbol f as binary operation. Thus, the domain C of $\mathcal{C}$ consists of the equivalence classes if ground terms in $G(\{f\}, \{0,1\})$ modulo commutativity. We expand $\mathcal{C}$ to an algebra $\mathcal{A}$ that also interprets the symbols $p_1, \ldots, p_n, \overline{p}_1, \ldots, \overline{p}_n, f_\vee, f_\wedge, f_\neg$ as follows:

- the domain of $\mathcal{A}$ is C;
- $p_i^{\mathcal{A}} = v(p_i)$ and $\overline{p}_i^{\mathcal{A}} = 1 - v(p_i)$ for $i = 1, \ldots, n$;
- on $\{0,1\}$, $f_\vee^{\mathcal{A}}$ is disjunction, $f_\wedge^{\mathcal{A}}$ is conjunction, and $f_\neg^{\mathcal{A}}$ is negation, and on tuples not in $\{0,1\} \times \{0,1\}$ they yield an arbitrary value, say 0;
- $f^{\mathcal{A}}$ is $f^{\mathcal{C}}$.

It is easy to see that $\mathcal{A}$ satisfies the identities in E_ϕ. In addition, commutativity of f and (5) are satisfied since these properties hold in the free commutative algebra [13]. It is now easy to see that $t_\phi^{\mathcal{A}} = v(\phi) = 0 = 0^{\mathcal{A}}$, and thus $\mathcal{A}$ does not satisfy the identity $t_\phi \approx 1$ since in $\mathcal{C}$ 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 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 (3), whereas the commutative extensional symbols in $\Sigma^e \cap \Sigma_c$ satisfy d-extensionality (5). 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 (6) 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

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

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

$$\begin{array}{llll} f_\vee(1,0) \to 1, & c_{f_\vee(1,0)} \to 1, & f_\vee(0,1) \to 1, & c_{f_\vee(0,1)} \to 1, \\ f_\neg(0) \to 1, & c_{f_\neg(0)} \to 1, & f_\neg(1) \to 0, & c_{f_\neg(1)} \to 0, \\ f(p,\overline{p}) \to c_{f(1,0)}, & f(1,0) \to c_{f(1,0)}, & c_{f(p,\overline{p})} \to c_{f(1,0)}. & \end{array} \tag{7}$$

Since no new constant rule is added, the construction proceeds with step (c). Due to the presence of the rules $f(p,\overline{p}) \to c_{f(1,0)}$ and $f(1,0) \to c_{f(1,0)}$ 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 $p \to 1, \overline{p} \to 0$ are added, and in the second $p \to 0, \overline{p} \to 1$ 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 $\widehat{R}^{\Sigma^e_c}_{r_1}(E)$ and $\widehat{R}^{\Sigma^e_c}_{r_2}(E)$ share the rules in (7), but the first contains $p \to 1$ whereas the second contains $p \to 0$.

Coming back to the general case, as in the proofs of Lemma 7 and Lemma 9, we can show the following for the rewrite systems $\widehat{R}^{\Sigma^e_c}_r(E)$.

Lemma 12. *For every run r, the term rewriting system $\widehat{R}^{\Sigma^e_c}_r(E)$ is produced in polynomial time, and the system $\widehat{R}^{\Sigma^e_c}_r(E) \cup R(\Sigma_c)$ is canonical.*

Using the canonical rewrite systems $\widehat{R}^{\Sigma^e_c}_r(E) \cup R(\Sigma_c)$, 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 $\Sigma^e \setminus \Sigma_c$, an d-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 d-extensionality for every $f \in \Sigma^e \cap \Sigma_c$ iff s_0, t_0 have the same canonical forms w.r.t. $\widehat{R}^{\Sigma^e_c}_r(E) \cup R(\Sigma_c)$ for every run r of the nondeterministic construction.*

Proof. First, we show the if-direction of the theorem by contraposition. Thus, assume that $\mathcal{A}$ is an algebra that satisfies E, commutativity for every $f \in \Sigma_c$, extensionality for every $f \in \Sigma^e \setminus \Sigma_c$, and d-extensionality for every $f \in \Sigma^e \cap \Sigma_c$, but in which s_0 and t_0 are not identified, i.e., $s_0^{\mathcal{A}} \neq t_0^{\mathcal{A}}$ holds. We expand $\mathcal{A}$ to the new constants in C_1 by setting $c_s^{\mathcal{A}} := s^{\mathcal{A}}$, and call the expansion obtained this way still $\mathcal{A}$. Since $\mathcal{A}$ 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 the construction. The same is true for the rules added in step (c) when treating symbols in $\Sigma^e \setminus \Sigma_c$. For symbols $f \in \Sigma^e \cap \Sigma_c$, we let $\mathcal{A}$ decide which option to take. To be more precise, for all pairs of distinct rules $f(c_1, c_2) \to d, f(c_1', c_2') \to d$ in $F_i^{\Sigma^e_c}(E)$, we can assume by induction that $f(c_1, c_2)^{\mathcal{A}} = f^{\mathcal{A}}(c_1^{\mathcal{A}}, c_2^{\mathcal{A}}) = d^{\mathcal{A}} = f^{\mathcal{A}}(c_1'^{\mathcal{A}}, c_2'^{\mathcal{A}}) = f(c_1', c_2')^{\mathcal{A}}$ holds. Since $\mathcal{A}$ satisfies d-extensionality for f, this implies that we have $c_1^{\mathcal{A}} = c_1'^{\mathcal{A}}, c_2^{\mathcal{A}} = c_2'^{\mathcal{A}}$ or $c_1^{\mathcal{A}} = c_2'^{\mathcal{A}}, c_2^{\mathcal{A}} = c_1'^{\mathcal{A}}$. 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 $\mathcal{A}$ is a model of $\widehat{R}^{\Sigma^e_c}_r(E)$. Since $\mathcal{A}$ satisfies commutativity for every symbol in Σ_c, it is also a model of $R(\Sigma_c)$.

Now, assume that s_0, t_0 have the same canonical forms w.r.t. $\widehat{R}^{\Sigma^e_c}_r(E) \cup R(\Sigma_c)$. This implies that $s_0 \approx_{\widehat{R}^{\Sigma^e_c}_r(E) \cup R(\Sigma_c)} t_0$. Since $\mathcal{A}$ is a model of $\widehat{R}^{\Sigma^e_c}_r(E) \cup R(\Sigma_c)$, we thus obtain $s_0^{\mathcal{A}} = t_0^{\mathcal{A}}$, which contradicts our assumption that $s_0^{\mathcal{A}} \neq t_0^{\mathcal{A}}$ holds. Thus, s_0, t_0 must have different canonical forms w.r.t. $\widehat{R}^{\Sigma^e_c}_r(E) \cup R(\Sigma_c)$.

Second, we show the only-if-direction, again 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. $\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)$. Since $\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ is canonical, this implies that $s_0 \not\approx_{\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)} t_0$. Let $\mathcal{A}$ be the initial algebra (i.e., the free algebra over the empty set of generators) for $\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)$ 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. $\approx_{\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)}$ as its elements, and any term $s \in G(\Sigma, C)$ is interpreted in $\mathcal{A}$ as the class of s. Consequently, $s_0 \not\approx_{\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)} t_0$ implies $s_0^{\mathcal{A}} \neq t_0^{\mathcal{A}}$. Thus, it is sufficient to show that $\mathcal{A}$ satisfies E, commutativity for every $f \in \Sigma_c$, extensionality for every $f \in \Sigma^e \setminus \Sigma_c$, and d-extensionality for every $f \in \Sigma^e \cap \Sigma_c$.

If $s \approx t \in E$, then $s \approx_{R(E)} t$ by Lemma 2, and thus $s \approx_{\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)} t$ (see the proof of Lemma 4). Consequently, these two terms are evaluated to the same element of $\mathcal{A}$, which shows that $\mathcal{A}$ satisfies the identities in E.

Let $f \in \Sigma_c$, and consider terms $s, t \in G(\Sigma, C)$. Then $f(s,t) \approx_{R(\Sigma_c)} f(t,s)$ according to the definition of $R(\Sigma_c)$, which implies that $f(s,t)$ and $f(t,s)$ are evaluated to the same element of $\mathcal{A}$. This shows that $\mathcal{A}$ interprets the elements of Σ_c as commutative functions.

Let $f \in \Sigma^e \setminus \Sigma_c$ 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 $\mathcal{A}$. Then $f(s_1, \ldots, s_n) \approx_{\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)} f(t_1, \ldots, t_n)$, and thus these terms have the same canonical forms w.r.t. $\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)$. 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. As in the proof of Theorem 2, there are two cases:

- 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, we have $s_j \approx_{\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)} s'_j = t'_j \approx_{\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)} t_j$ 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 $\mathcal{A}$.
- The terms $f(s_1, \ldots, s_n)$ and $f(t_1, \ldots, t_n)$ reduce to the same constant d. Then $\widehat{R}_r^{\Sigma_c^e}(E)$ must contain the rules $f(s'_1, \ldots, s'_n) \to d$ and $f(t'_1, \ldots, t'_n) \to d$. By the construction of $\widehat{R}_r^{\Sigma_c^e}(E)$, we thus have $s'_j = t'_j$ for all $j = 1, \ldots, n$ since otherwise new constant rules would have been added and the construction would not yet have terminated. This yields $s_j \approx_{\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)} s'_j = t'_j \approx_{\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)} t_j$ 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 $\mathcal{A}$.

Summing up, we have thus shown that $\mathcal{A}$ satisfies extensionality for the symbols in $\Sigma^e \setminus \Sigma_c$.

Finally, let $f \in \Sigma^e \cap \Sigma_c$ be a commutative and d-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 $\mathcal{A}$. Again, this implies that these terms have the same canonical forms w.r.t. $\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)$. 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 $R(\Sigma_c)$ 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 $R(\Sigma_c)$ 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 above, this implies that s_1 and t_1 as well as s_2 and t_2 respectively evaluate to the same elements of $\mathcal{A}$.

- 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 $f(s'_2, s'_1)$ 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 $\mathcal{A}$.
- 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 $f(t'_2, t'_1)$, 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 $R(\Sigma_c)$ have been applied before the reduction to the constant d:
  - The term rewriting system $\widehat{R}_r^{\Sigma_c^e}(E)$ contains the rules $f(s'_1, s'_2) \to d$ and $f(t'_1, t'_2) \to d$. By the construction of $\widehat{R}_r^{\Sigma_c^e}(E)$, 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 $\mathcal{A}$, or s_1 and t_2 as well as s_2 and t_1 respectively evaluate to the same elements of $\mathcal{A}$.
  - 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 $\mathcal{A}$ satisfies d-extensionality for the symbols in $\Sigma^e \cap \Sigma_c$, which completes the proof of the theorem. □

Coming back to Example 3, we note that $\phi = p \vee \neg p$ is valid, and thus (by Lemma 11), the identity $f_\vee(p, f_\neg(p)) \approx 1$ holds in all algebras that satisfy E_ϕ and interpret f as a commutative and d-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.

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

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 d-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. $\widehat{R}_r^{\Sigma_c^e}(E) \cup R(\Sigma_c)$. The system $\widehat{R}_r^{\Sigma_c^e}(E)$ can be generated in nondeterministic polynomial time, and the canonical forms can be computed in polynomial time. □

5 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 d-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 $O(n \log n)$ [10,11]. Since this complexity upper bound can also be achieved using a rewriting-based approach [14,7], 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 considered in Section 3, but this question was not in the focus of the present paper.

The rules specifying extensionality are Horn rules whose atoms are (non-ground) identities, and where the consequence is an identity between variables occurring in the precondition. The question arises which other such Horn rules can be added without increasing the complexity of the word problem. Note that our proof for the case of extensionality (see the proof of Theorem 2, Case 1 in the case distinction) uses the fact that the variables to be identified occur in the same argument position of the symbol f. It is not clear how to deal with this case if this is not satisfied (as e.g., in the rule $f(x, y) \approx f(x', y') \to x \approx y'$). Note, however, that Case 1 cannot even occur if the root function symbols of the identity in the precondition are not the same, and thus a Horn rule for which this is the case (like $f(x, y) \approx g(x', y') \to x \approx y'$) should be harmless.

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).

References

1. Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.
2. Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
3. Leo Bachmair, I.V. Ramakrishnan, Ashish Tiwari, and Laurent Vigneron. Congruence closure modulo associativity and commutativity. In Hélène Kirchner and Christophe Ringeissen, editors, *Proc. of the Third International Workshop on Frontiers of Combining Systems (FroCoS 2000)*, volume 1794 of *Lecture Notes in Computer Science*, pages 245–259. Springer, 2000.
4. Peter J. Downey, Ravi Sethi, and Robert Endre Tarjan. Variations on the common subexpression problem. *J. ACM*, 27(4):758–771, 1980.
5. Jean H. Gallier, Paliath Narendran, David A. Plaisted, Stan Raatz, and Wayne Snyder. An algorithm for finding canonical sets of ground rewrite rules in polynomial time. *J. ACM*, 40(1):1–16, 1993.
6. Deepak Kapur. Shostak's congruence closure as completion. In *Proc. of the 8th Int. Conf. on Rewriting Techniques and Applications (RTA 1997)*, volume 1232 of *Lecture Notes in Computer Science*, pages 23–37. Springer, 1997.
7. Deepak Kapur. Conditional congruence closure over uninterpreted and interpreted symbols. *J. Systems Science & Complexity*, 32(1):317–355, 2019.
8. Dexter Kozen. Complexity of finitely presented algebras. In *Proc. of the 9th ACM Symposium on Theory of Computing*, pages 164–177. ACM, 1977.
9. Paliath Narendran and Michaël Rusinowitch. Any ground associative-commutative theory has a finite canonical system. *J. Autom. Reasoning*, 17(1):131–143, 1996.
10. Greg Nelson and Derek Oppen. Fast decision procedures based on congruence closure. *J. of the ACM*, 27(2):356–364, 1980.
11. Robert Nieuwenhuis and Albert Oliveras. Fast congruence closure and extensions. *Inf. Comput.*, 205(4):557–580, 2007.

12. Robert E. Shostak. An algorithm for reasoning about equality. *Commun. ACM*, 21(7):583–585, 1978.
13. J. H. Siekmann. Unification of commutative terms. In *Proceedings of the International Symposium on Symbolic and Algebraic Manipulation, EUROSAM'79*, volume 72 of *Lecture Notes in Computer Science*, pages 531–545, Marseille, France, 1979. Springer.
14. Wayne Snyder. A fast algorithm for generating reduced ground rewriting systems from a set of ground equations. *J. Symb. Comput.*, 15(4):415–450, 1993.
15. Aaron Stump, Clark W. Barrett, David L. Dill, and Jeremy R. Levitt. A decision procedure for an extensional theory of arrays. In *Proc. of 16th Annual IEEE Symposium on Logic in Computer Science (LICS 2001)*, pages 29–37. IEEE Computer Society, 2001.
16. Wolfgang Wechler. *Universal Algebra for Computer Scientists*, volume 25 of *EATCS Monographs on Theoretical Computer Science*. Springer, 1992.