Congruence Closure Modulo Associativity and Commutativity*

L. Bachmair[1], I. V. Ramakrishnan[1], A. Tiwari[1], L. Vigneron[2]

[1] Department of Computer Science
State University of New York
Stony Brook, NY 11794-4400, U.S.A
{leo,ram,astiwari}@cs.sunysb.edu
[2] LORIA – Université Nancy 2
Campus Scientifique, B.P. 239
54506 Vandœuvre-lès-Nancy Cedex, France
vigneron@loria.fr

Abstract. We introduce the notion of an *associative-commutative congruence closure* and show how such closures can be constructed via completion-like transition rules. This method is based on combining completion algorithms for theories over disjoint signatures to produce a convergent rewrite system over an extended signature. This approach can also be used to solve the word problem for ground AC-theories without the need for AC-simplification orderings total on ground terms.
Associative-commutative congruence closure provides a novel way to construct a convergent rewrite system for a ground AC-theory. This is done by transforming an AC-congruence closure, which is described by rewrite rules over an extended signature, to a rewrite system over the original signature. The set of rewrite rules thus obtained is convergent with respect to a new and simpler notion of associative-commutative reduction.

1 Introduction

Congruence closure algorithms have been used to decide if an equality $s \approx t$ logically follows from a set of equalities $E = \{s_1 \approx t_1, s_2 \approx t_2, \cdots, s_k \approx t_k\}$, where all terms are constructed from *uninterpreted or free* function symbols and constants. They also provide a decision procedure for validity problem in the quantifier-free theory of equality (with uninterpreted function symbols) [15].

The Nelson-Oppen method [14, 17, 19] is a general procedure to decide satisfiability of quantifier-free formulas in combination of certain[1] first-order theories. The crucial steps involved are variable abstraction and equality propagation. The Nelson-Oppen method can be used to obtain a decision procedure for the

* The research described in this paper was supported in part by the National Science Foundation under grants CCR-9902031, CCR-9711386 and EIA-9705998.

[1] Each individual theory is assumed to be a stably infinite first-order theory (with equality) over a disjoint signature such that satisfiability of quantifier-free formulas in the theory is decidable.

quantifier-free theory of equality (with uninterpreted function symbols Σ)–as the quantifier-free theory of equality over Σ can be viewed as a combination of the quantifier-free theory of equality over the singleton sets $\Sigma_i = \{f\}$, where Σ is a disjoint union $\cup_i \Sigma_i$ of these sets.

Reconciling these two approaches for deciding satisfiability of quantifier-free formulas (in the theory of equality with uninterpreted function symbols) leads to the concept of an *abstract congruence closure*, as described in [3]. In this paper, we extend this notion to include associative-commutative function symbols. In other words, some function symbols in Σ are assumed to satisfy the associativity and commutativity axioms. Thus, if f is such a symbol, the theory over $\Sigma_i = \{f\}$ is that of a commutative semigroup (and not just the pure theory of equality).

We formally define an associative-commutative congruence closure in Section 2 and present a set of completion-like transition rules to construct such closures in Section 3. We prove the correctness of the transition rules using concepts of proof simplification and normalization, and also establish its termination in Section 4. This establishes decidability for ground AC-theories without the use of AC-simplification orderings that are total on ground terms [13, 11]. The construction of AC-congruence closure can be optimized. We discuss some optimizations in Section 5.

We also consider the problem of constructing a convergent rewrite system (over the same signature) for a ground AC-theory. However, rather than obtaining a system R of rules such that $\rightarrow_{AC\backslash R^e}$ is convergent, we construct a set R such that a different reduction relation, which we denote by $\rightarrow_{P\backslash R^s}$ is convergent. This new notion is much simpler than usual rewriting modulo AC. This set is constructed by transforming the obtained AC-congruence closure, which is a convergent system over an extended signature, to a set over the original signature, see Section 6.

Preliminaries

Let Σ be a signature consisting of constants and function symbols, and $\mathcal{V}$ be a set of variables. The arity of a symbol f in Σ, denoted by $\alpha(f)$, is a set of natural numbers. The set of ground terms $\mathcal{T}(\Sigma)$ over Σ is the smallest set containing $\{c \in \Sigma : \alpha(c) = \{0\}\}$ and such that $f(t_1, \ldots, t_n) \in \mathcal{T}(\Sigma)$ whenever $f \in \Sigma, n \in \alpha(f)$ and $t_1, \ldots, t_n \in \mathcal{T}(\Sigma)$. The symbols $s, t, u, \ldots$ are used to denote terms; $f, g, \ldots$, function symbols; and $x, y, z, \ldots$, variables. We write $E[t]$ to indicate that an expression E contains t as a subterm and (ambiguously) denote by $E[u]$ the result of replacing a particular occurrence of t by u. (The same notation will be employed if E is a set of expressions.) By $E\sigma$ we denote the result of applying a substitution σ to E.

An *equation* is a pair of terms, written $s \approx t$. We usually identify the two equations $s \approx t$ and $t \approx s$; if the distinction is important, we call the equation a *rewrite rule* and write $s \rightarrow t$. A *rewrite system* is a set of rewrite rules. The *rewrite relation* $\rightarrow_R$ induced by a set of equations R is defined by: $u \rightarrow_R t$ if, and only if, u contains a subterm $l\sigma$ and $t = u[r\sigma]$, for some rewrite rule $l \rightarrow r$ in R and some substitution σ.

A term t is said to be in *normal form* with respect to a rewrite system R, or in R-*normal form*, if there is no term u, such that $t \rightarrow_R u$. A rewrite system is said to be (ground) *confluent* if all (ground) terms have a unique normal form. Rewrite systems that are (ground) confluent and terminating are called (ground) *convergent*.

The set of Σ-terms with variables in $\mathcal{V}$ is denoted by $\mathcal{T}(\Sigma, \mathcal{V})$. The *equational theory* induced by a set $\mathcal{E} \subseteq \mathcal{T}(\Sigma, \mathcal{V}) \times \mathcal{T}(\Sigma, \mathcal{V})$ of equations is defined as the reflexive, symmetric, and transitive closure of the rewrite relation induced by $\mathcal{E}$. The pure theory of equality is obtained when the set $\mathcal{E}$ is empty. If $\Sigma_{AC} \subset \Sigma$ is a finite set of function symbols, we denote by P the identitites

$$f(x_1, \ldots, x_k, s, y_1, \ldots, y_l, t, z_1, \ldots, z_m) \approx f(x_1, \ldots, x_k, t, y_1, \ldots, y_l, s, z_1, \ldots, z_m)$$

where $f \in \Sigma_{AC}$ and $k + l + m \in \alpha(f)$; and by F the set of identities

$$f(x_1, \ldots, x_m, f(y_1, \ldots, y_r), z_1, \ldots, z_n) \approx f(x_1, \ldots, x_m, y_1, \ldots, y_r, z_1, \ldots, z_n)$$

where $f \in \Sigma_{AC}$ and $\{m+n+1, m+n+r\} \subset \alpha(f)$. The congruence induced by P is called a *permutation congruence*. Flattening refers to normalizing a term with respect to the set F (considered as rewrite rules). The set $AC = F \cup P$ defines an AC-theory. The symbols in Σ_{AC} are called associative-commutative operators[2]. We require that $\alpha(f)$ be singleton for all $f \in \Sigma - \Sigma_{AC}$ and $\alpha(f) = \{2, 3, 4, \ldots\}$ for all $f \in \Sigma_{AC}$.

Rewriting in presence of associative and commutative operators requires *extensions* of rules [16]. Given an AC-operator f and a rewrite rule ρ : $f(c_1, \ldots, c_k) \rightarrow c$, we define its extension ρ^e as $f(c_1, \ldots, c_k, x) \rightarrow f(c, x)$. Given a set of rewrite rules R, by R^e we denote the set R plus extensions of rules in R. In short, when working with AC-symbols (i) extensions have to be used for rewriting terms and computing critical pairs, and (ii) syntactic matching and unification is replaced by matching and unification modulo AC. Reduction modulo AC is defined by $s \rightarrow_{AC \backslash R} t$ iff there exists a rule $l \rightarrow r$ in R and a substitution σ such that $s = s[l']$, $l' \leftrightarrow^*_{AC} l\sigma$, and $t = s[r\sigma]$; see [6] for details.

A *multiset* over a set S is a mapping M from S to the natural numbers. Any ordering $\succ$ on a set S can be extended to an ordering $\succ^{mult}$ on multisets over S as follows: $M \succ^{mult} N$ iff $M \neq N$ and whenever $N(x) > M(x)$ then $M(y) > N(y)$, for some $y \succ x$. The multiset ordering $\succ^{mult}$ (on finite multisets) is well founded if the ordering $\succ$ is well founded [7]. Multiset inclusion is defined as $M \subseteq N$ if $M(x) \leq N(x)$ for all $x \in S$.

We consider the problem of deciding satisfiability of quantifier-free formulas in an equational theory presented by $\mathcal{E}$. When $\mathcal{E}$ is empty, congruence closure provides an efficient procedure [15]. The notion of AC-congruence closure handles the case when $\mathcal{E} = AC$.

[2] The equations $F \cup P$ define a conservative extension of the theory of associativity and commutativity to varyadic terms. For a fixed arity binary function symbol, the equations $f(x, y) \approx f(y, x)$ and $f(f(x, y), z) \approx f(x, f(y, z))$ define an AC-theory.

2 Associative-Commutative Congruence Closure

Let Σ be a signature, and E a set of ground equations over Σ. Let Σ_{AC} be some subset of Σ, containing all the associative-commutative operators and AC be the set $F \cup P$ as defined above. The set Σ can be written as a disjoint union of singleton sets Σ_i's, where each Σ_i contains exactly one function symbol in Σ. Inspired by the variable abstraction phase in the Nelson-Oppen method [14] and the idea of using constants as "names" of subterms [9], we first attempt to transform the set E of equations into a set of "pure" equations, i.e., each equation is over some signature $\Sigma_i \cup K$, where K is a set of new constants (names) introduced by variable abstraction, and shared between the individual theories. Each pure equation thus obtained is in a special form.

Definition 1. *Let Σ be a signature and K be a set of constants disjoint from Σ. By a D-*rule *(with respect to Σ and K) we mean a rewrite rule of the form*

$$f(c_1, \dots, c_k) \to c_0$$

where $f \in \Sigma$ and $c_0, c_1, \dots, c_k$ are constants in K.

*An equation $c \to d$, where c and d are constants in K, is called a C-*rule *(with respect to K). An undirected equation $c \approx d$ is called a C-*equation.

Rewrite rules of the form $f(c_1, \dots, c_k) \to f(d_1, \dots, d_l)$, where $f \in \Sigma_{AC}$, and $c_1, \cdots, c_k, d_1, \cdots, d_l \in K$, will be called AC-rules.

Each D- and AC-rule is over exactly one signature $\Sigma_i \cup K$. The C-rules are equations over every signature $\Sigma_i \cup K$.

For example, let Σ consist of function symbols, a, b, c, f and g, (f is AC) and let E_0 be a set of three equations $f(a, c) \approx a$, $f(c, g(f(b, c))) \approx b$ and $g(f(b, c)) \approx f(b, c)$. Viewing Σ as the disjoint union of signatures $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$, $\Sigma_3 = \{c\}$, $\Sigma_4 = \{f\}$ and $\Sigma_5 = \{g\}$, we can introduce new constants $K = \{c_0, c_1, c_2, c_3, c_4\}$ so that each equation (or rule) is over some $\Sigma_i \cup K$. Therefore, if we let

$$D_1 = \{a \to c_1,\ b \to c_2,\ c \to c_3,\ f(c_2, c_3) \to c_4,\ g(c_4) \to c_5\},$$

then, using these rules we can simplify the original equations in E_0 to obtain the new set $E_1 = \{f(c_1, c_3) \approx c_1,\ f(c_3, c_5) \approx c_2,\ c_5 \approx c_4\}$.

Definition 2. *Let R be a set of D-rules, C-rules and AC-rules (with respect to Σ and K). We say that a constant c in K* represents *a term t in $\mathcal{T}(\Sigma \cup K)$ (via the rewrite system R) if $t \to^*_{AC \backslash R^e} c$. A term t is also said to be* represented *by R if it is represented by some constant via R.*

For example, the constant c_4 represents the term $f(c, b)$ via D_1.

Definition 3. *Let Σ be a signature and K be a set of constants disjoint from Σ. A ground rewrite system $R = A^e \cup D^e \cup C$ is said to be an* associative-commutative congruence closure *(with respect to Σ and K) if* *(i) D is a set of D-rules, C is a set of C-rules, A is a set of AC-rules, and if a constant $c \in K$ is in normal form with respect to R, then c represents at least one term $t \in \mathcal{T}(\Sigma)$ via R, and*

(ii) $AC\backslash R$ is a ground convergent modulo AC over $\mathcal{T}(\Sigma \cup K)$.

In addition, if E is a set of ground equations over $\mathcal{T}(\Sigma)$ such that,

(iii) If s and t are terms over $\mathcal{T}(\Sigma)$, then $s \leftrightarrow^*_{AC\cup E} t$ if, and only if, $s \rightarrow^*_{AC\backslash R} \circ \leftrightarrow^*_{AC} \circ \leftarrow^*_{AC\backslash R} t$,

then R will be called an associative-commutative congruence closure for E.

When Σ_{AC} is empty this definition specializes to that of an abstract congruence closure [3].

For instance, the rewrite system $D_1 \cup E_1$ above is not a congruence closure for E_0, as it is not a ground convergent rewrite system. But we can transform $D_1 \cup E_1$ into a suitable rewrite system, using a completion-like process described in more detail in the next section, to obtain a congruence closure (Example 4),

$$\begin{aligned} R' = \{ & a \rightarrow c_1,\ b \rightarrow c_2,\ c \rightarrow c_3,\ fc_2c_3 \rightarrow c_4,\ fc_3c_4 \rightarrow c_2,\ fc_1c_3 \rightarrow c_1, \\ & fc_2c_2 \rightarrow fc_4c_4,\ fc_1c_2 \rightarrow fc_1c_4,\ gc_4 \rightarrow c_4\} \end{aligned}$$

that provides a more compact representation of E_0. Note that the constant c_1 represents infinitely many terms via R'.

3 Construction of Congruence Closures

The completion procedure is obtained by putting together completion procedures over individual signatures $\Sigma_i \cup K$. Clearly, there are two distinct cases that need to be combined: when $f_i \in \Sigma_i$ is AC, and when it is not. The individual completion procedures working on equations in $\Sigma_i \cup K$ need to exchange equations between constants in K that are generated during the completion procedure. For this purpose, we choose an ordering in which the only terms smaller than a constant in K are other constants in K.

Let U be a set of symbols from which new names (constants) are chosen. We need a (partial) AC-compatible reduction ordering which orients the D-rules in the right way, and orients all the C- and AC-equations. The precedence-based AC-compatible ordering $\succ$ of [18], with any precedence in which $f \succ_{\Sigma\cup U} c$, whenever $f \in \Sigma$ and $c \in U$, serves the purpose. However, much simpler partial orderings would suffice too, but for convenience we use the ordering in [18]. In our case, this simply means that, orientation of D-rules is from left to right. Additionally, the orientation of an AC-rule will be given by: $f(c_1, \ldots, c_i) \succ f(c'_1, \ldots, c'_j)$ iff either $i > j$, or $i = j$ and $\{c_1, \ldots, c_i\} \succ^{mult} \{c'_1, \ldots, c'_j\}$, i.e., if the two terms have the same number of arguments, we compare the multisets of constants using a multiset extension $\succ^{mult}$ of the precedence $\succ_{\Sigma\cup U}$, see [7].

We next present a general method for construction of associative-commutative congruence closures. Our description is fairly abstract, in terms of transition rules that operate on triples (K, E, R), where K is a set of new constants that are introduced (the original signature Σ is fixed); E is a set of ground equations (over $\Sigma \cup K$) yet to be processed; and R is a set of C-rules, D-rules and AC-rules. Triples represent possible *states* in the process of constructing a closure. The initial state is $(\emptyset, E, \emptyset)$, where E is the input set of ground equations.

Abstraction

New constants are introduced by the following transition.

Extension: $$\frac{(K, E[f(c_1, \ldots, c_k)], R)}{(K \cup \{c\}, E[c], R \cup \{f(c_1, \cdots, c_k) \to c\})}$$

where $f \in \Sigma$; $c_1, \ldots, c_k$ are constants in K; $f(c_1, \ldots, c_k)$ is a term occurring in (some equation in) E; $c \notin \Sigma \cup K$; and, either $f \notin \Sigma_{AC}$, or, $f(c_1, \ldots, c_k)$ occurs as a proper subterm of a flattened term, or, it occurs as one side of an equation in which the top function symbol at the other side is some $g \neq f$.

Once a D-rule $f(c_1, \ldots, c_k) \to c$ has been introduced by extension, it can be used to eliminate any other occurrence of $f(c_1, \ldots, c_k)$ using the following rule.

Simplification: $$\frac{(K, E[s], R)}{(K, E[t], R)}$$

where s occurs in some equation in E, and, $s \to_{AC\backslash R^e} t$.

We will also use flattening to replace a term in E or R by its corresponding flattened form.

Deduction

It is fairly easy to see that any equation in E can be transformed to a C- or a AC-equation by suitable extension, flattening and simplification. Once we have obtained rules that contain terms only from a specific signature $\Sigma_i \cup K$, we can perform deduction steps on the separate parts.

In the case when $f_i \in \Sigma_i$ is an AC-symbol we use AC-superposition wherein we consider overlaps between extensions of AC-rules.

ACSuperposition: $$\frac{(K, E, R)}{(K, E \cup \{f(s, x\sigma) \approx f(t, y\sigma)\}, R)}$$

if $f \in \Sigma_{AC}$, there exist D- or AC-rules $f(c_1, \ldots, c_k) \to s$ and $f(d_1, \ldots, d_l) \to t$ in R, substitution σ is a most-general unifier modulo AC of $f(c_1, \ldots, c_k, x)$ and $f(d_1, \ldots, d_l, y)$[3] and $\{c_1, \ldots, c_k\} \cap \{d_1, \ldots, d_l\} \neq \emptyset$.

In the special case when one multiset is contained in the other, we obtain the AC-collapse rule.

ACCollapse: $$\frac{(K, E, R \cup \{t \to s\})}{(K, E \cup \{t' \approx s\}, R)}$$

[3] For the special case in hand, there is exactly one most general unifier modulo AC, and it is easy to compute.

if for some $u \to v \in R$, $t \to_{AC\backslash\{u\to v\}^e} t'$, and if $t \leftrightarrow^*_{AC} u$ then $s \succ v$.

In case of rules that do not contain AC-symbols at the top, superposition reduces to simplification, and hence AC-collapse also captures such superpositions. Note that we do not explicitly add AC extensions of rules to the set R, and so any rule in R is either a C-rule, or a D-rule, or an AC-rules, and *not* its extension. We implicitly work with extensions in AC-superposition.

Orientation

Equations are moved from the E-component of the state to the R-component by orientation. All rules added to the R-component are either C-rules, D-rules or AC-rules.

Orientation: $$\frac{(K, E \cup \{s \approx t\}, R)}{(K, E, R \cup \{s \to t\})}$$

if $s \succ t$, and, $s \to t$ is either a D-rule, or a C-rule, or a AC-rule.

Deletion allows us to delete trivial equations.

Deletion: $$\frac{(K, E \cup \{s \approx t\}, R)}{(K, E, R)}$$

if $s \leftrightarrow^*_{AC} t$.

Simplification

We need additional transition rules to incorporate interaction between the individual completion processes. In particular, C-rules can be used to perform simplifications on the left- and right-hand sides of other rules. The use of C-rules to simplify left-hand sides of rules is captured by AC-collapse. The simplification on the right-hand sides is subsumed by the following generalized composition rule.

Composition: $$\frac{(K, E, R \cup \{t \to s\})}{(K, E, R \cup \{t \to s'\})}$$

where $s \to_{AC\backslash R^e} s'$.

Example 4. Let $E_0 = \{f(a, c) \approx a,\ f(c, g(f(b, c))) \approx b,\ g(f(b, c)) \approx f(b, c)\}$. We show below some intermediate stages of a derivation.

i	Constants K_i	Equations E_i	Rules R_i	Transitions
0	$\emptyset$	E_0	$\emptyset$	
1	$\{c_1, c_3\}$	$\{fcgfbc \approx b,$ $gfbc \approx fbc\}$	$\{a \to c_1, c \to c_3,$ $fc_1c_3 \to c_1\}$	$\mathbf{Ext}^2 \circ \mathbf{Sim} \circ$ $\mathbf{Ori}$
2	$K_1 \cup \{c_2, c_4\}$	$\{fcgfbc \approx b\}$	$R_1 \cup \{b \to c_2,$ $fc_2c_3 \to c_4, gc_4 \to c_4\}$	$\mathbf{Sim}^2 \circ \mathbf{Ext}^2 \circ$ $\mathbf{Sim} \circ \mathbf{Ori}$
3	K_2	$\emptyset$	$R_2 \cup \{fc_3c_4 \to c_2\}$	$\mathbf{Sim}^6 \circ \mathbf{Ori}$
4	K_2	$\emptyset$	$R_3 \cup \{fc_1c_2 \to fc_1c_4\}$	$\mathbf{ACSup} \circ \mathbf{Ori}$
5	K_2	$\emptyset$	$R_4 \cup \{fc_2c_2 \to fc_4c_4\}$	$\mathbf{ACSup} \circ \mathbf{Ori}$

The derivation moves equations, one by one, from the E-component of the state to the R-component through simplification, extension and orientation. It can be verified that the set R_5 is an AC congruence closure for E_0. Note that the side-condition in extension disallows breaking of an AC-rule into two D-rules, which is crucial for termination. We assume that f is AC and $c_i \succ c_j$ if $i < j$.

4 Termination and Correctness

Definition 5. *We use the symbol $\vdash$ to denote the one-step transition relation on states induced by the above transition rules. A derivation is a sequence of states $\xi_0 \vdash \xi_1 \vdash \cdots$. A derivation is said to be fair if any transition rule which is continuously enabled is eventually applied. The set R_∞ of persisting rules is defined as $\cup_i \cap_{j>i} R_j$; and similarly, $K_\infty = \cup_i \cap_{j>i} K_j$.*

We shall prove that any fair derivation will only generate finitely many persisting rewrite rules (in the third component) using Dickson's lemma [5]. Multisets over K_∞ can be compared using the multiset inclusion relation. If K_∞ is finite, this relation defines a Dickson partial order.

Lemma 6. *The set of persisting rules R_∞ in any fair derivation starting from state $(\emptyset, E, \emptyset)$ is finite.*

Proof. We first claim that K_∞ is finite. To see this, note that new constants are created by extension. Using finitely many applications of extension, simplification, flattening and orientation, we can move all rules from the initial second component E of the state tuple to the third component R. Fairness ensures that this will eventually happen. Thereafter, any equations ever added to E can be oriented using flattening and orientation, hence we never apply extension subsequently (see the side condition of the extension rule). Let $K_\infty = \{c_1, \ldots, c_n\}$.

Next we claim that R_∞ is finite. Suppose, R_∞ contains infinitely many rules. No transition rule (except extension) increases the number of D- and C-rules in $E \cup R$. Therefore, R_∞ contains infinitely many AC-rules, and since Σ_{AC} is finite, one AC-operator, say $f \in \Sigma_{AC}$, must occur infinitely often on the left-hand sides of R_∞. By Dickson's lemma, there exists an infinite chain of rules, $f(c_{11}, \ldots, c_{1k_1}) \to s_0, f(c_{21}, \ldots, c_{2k_2}) \to s_1, \ldots$, such that $\{c_{11}, \ldots, c_{1k_1}\} \subseteq \{c_{21}, \ldots, c_{2k_2}\} \subseteq \cdots$, where $\{c_{i1}, \ldots, c_{ik_i}\}$ denotes a multiset and $\subseteq$ denotes multiset inclusion. But, this contradicts fairness (in application of AC-collapse).

Proof ordering

The correctness of the procedure will be established using proof simplification techniques for associative-commutative completion, as described by Bachmair [1] and Bachmair and Dershowitz [2]. In fact, we can directly use the results and the proof measure from [2]. However, since all rules in R have a special form, we can choose a simpler proof ordering.

Let $s = s[u\sigma] \leftrightarrow s[v\sigma] = t$ be a proof step using the equation (rule) $u \approx v \in AC \cup E \cup R$. The complexity of this proof step is defined by

$$\begin{array}{rlrl}
(\{s,t\},\bot,\bot) & \text{if } u \approx v \in E & (\{s\},\bot,t) & \text{if } u \approx v \in AC \\
(\{s\},u,t) & \text{if } u \rightarrow v \in R & (\{t\},v,s) & \text{if } v \rightarrow u \in R \\
(\{s\},\bot,t) & \text{if } u \rightarrow v \in R^e - R & (\{t\},\bot,s) & \text{if } v \rightarrow u \in R^e - R
\end{array}$$

where $\bot$ is a new symbol. Tuples are compared lexicographically using the multiset extension of the reduction ordering $\succ$ on terms over $\Sigma \cup K_\infty$ in the first component, and the ordering $\succ$ on the second and third component. The constant $\bot$ is assumed to be minimum. The complexity of a proof is the multiset of complexities of its proof steps. The multiset extension of the ordering on tuples yields a proof ordering, denoted by the symbol $\succ_{\mathcal{P}}$. The ordering $\succ_{\mathcal{P}}$ on proofs is well founded as it is a lexicographic combination of well founded orderings.

Lemma 7. *Suppose $(K, E, R) \vdash (K', E', R')$. Then, for any two terms $s, t \in \mathcal{T}(\Sigma)$, it is the case that $s \leftrightarrow^*_{AC\cup E'\cup R'} t$ iff $s \leftrightarrow^*_{AC\cup E\cup R} t$. Further, if π is a ground proof, $s_0 \leftrightarrow s_1 \leftrightarrow \cdots \leftrightarrow s_k$, in $AC \cup E \cup R$, then there is a proof π', $s_0 = s'_0 \leftrightarrow s'_1 \leftrightarrow \cdots \leftrightarrow s'_l = s_k$, in $AC \cup E' \cup R'$ such that $\pi \succeq_{\mathcal{P}} \pi'$.*

The first part of the lemma, which states that the congruence on $\mathcal{T}(\Sigma)$ remains unchanged, is easily verified by exhaustively checking it for each transition rule. For the second part, one needs to check that each equation in $(E - E') \cup (R - R')$ has a simpler proof in $E' \cup R' \cup AC$ for each transition rule application, see [2]. Note that extensions of rules are not added explicitly, and hence, they are never deleted either. Once we converge to R_∞, we introduce extensions to take care of cliffs in proofs.

Lemma 8. *If R_∞ is a set of persisting rules of a fair derivation starting from the state $(\emptyset, E, \emptyset)$, then, R^e_∞ is a ground convergent (modulo AC) rewrite system. Furthermore, $E_\infty = \emptyset$.*

Proof. (Sketch) Fairness implies that all critical pairs (modulo AC) between rules in R^e_∞ are contained in the set $\cup_i E_i$. Since a fair derivation is non failing, $E_\infty = \emptyset$. Since the proof ordering is noetherian, for every proof in $E_i \cup R_i \cup AC$, there exists a minimal proof π in $E_\infty \cup R_\infty \cup AC$. The minimal proof π in $R_\infty \cup AC$ can be further transformed to a minimal proof π' in $R^e_\infty \cup AC$ by eliminating cliffs arising from proper overlaps between AC-rules and R^e_∞-rules.

The proof π' is a rewrite proof. This follows from observing that π' does not contain any non-overlap, variable overlap, cliff or a peak. Any such pattern can be replaced by a smaller proof pattern, thus contradicting the minimality of the proof π. The details are similar, though much simpler in our special case, to those presented in [2].

Using Lemmas 7 and 8, we can easily prove that:

Proposition 9. *Let R_∞ be the set of persisting rules of a fair derivation starting from state $(\emptyset, E, \emptyset)$. Let $s, t \in \mathcal{T}(\Sigma)$. Then $s \leftrightarrow^*_{E\cup AC} t$ iff $s \rightarrow^*_{AC\backslash R^e_\infty} \circ \leftrightarrow^*_{AC} \circ \leftarrow^*_{AC\backslash R^e_\infty} t$. Furthermore, the set R^e_∞ is an associative-commutative congruence closure for E.*

Since R_∞ is finite, there exists a k such that $R_\infty \subset R_k$. Though there exist non-terminating fair derivations, we can still identify the finite set of persisting rules if we eagerly apply AC-collapse before any application of AC-superposition at any point after stage k.

Example 10. Let us describe one interesting example. Consider the initial set of equations $E_0 = \{\ f(a,b) \approx b,\ g(b) \approx f(a,a),\ f(b, g(f(a,a,b))) \approx c\}$. Let f be an AC-symbol. We show some of the intermediate states of a fair derivation below:

i	Constants K_i	Equations E_i	Rules R_i
0	$\emptyset$	E_0	$\emptyset$
1	$\{c_1, c_2\}$	$\{gb \approx faa,\ fbgfaab \approx c\}$	$\{a \to c_1,\ b \to c_2,\ fc_1c_2 \to c_2\}$
2	$\{c_1, c_2, c_3\}$	$\{fbgfaab \approx c\}$	$R_1 \cup \{gc_2 \to c_3,\ fc_1c_1 \to c_3\}$
3	$\{c_1, c_2, c_3\}$	$\{fbgfaab \approx c\}$	$R_2 \cup \{fc_2c_3 \to c_2\}$
4	$\{c_1, c_2, c_3\}$	$\emptyset$	$R_3 \cup \{c \to c_2\}$

Note that since we applied AC-superposition in step 3 before simplifying the last equation in E, we avoided introducing a lot of new names.

5 Optimizations

Certain optimizations can be incorporated into the basic set of transition rules given above for computation of congruence closures. A lot of inferences can be avoided if we note that we do not need to consider extensions of all rules that have an AC symbol on top of their left-hand sides (for the purpose of critical pair computations). For example, we need not consider extensions of those D-rules that are created by extension to name a *proper subterm* in E. This fact can be easily incorporated using constraints, but we chose not to show it here for the sake of simplicity and clarity. Rather than formalizing this completely here, we illustrate this with an example below.

Example 11. Again suppose that the initial set of equations E_0 is: $\{f(a,b) \approx b,\ g(b) \approx f(a,a),\ f(b, g(f(b, g(f(a,a,b))))) \approx c\}$. If we choose to first apply extension and simplification until all the initial equations can be oriented using orientation, then, we will generate the following set of eleven D-rules:

$$\begin{array}{rrrr} a \to c_1 & b \to c_2 & fc_1c_2 \to c_2 & fc_1c_1 \to c_3 \\ gc_2 \to c_3 & fc_3c_2 \to c_4 & gc_4 \to c_5 & fc_2c_5 \to c_6 \\ gc_6 \to c_7 & fc_2c_7 \to c_8 & c \to c_8 & \end{array}$$

Now, note the huge number of possible AC-superpositions. However, we need not consider all of them. In particular, we need to consider extensions only of three rules: $fc_1c_2 \to c_2$, $fc_1c_1 \to c_3$, and $fc_2c_7 \to c_8$, and hence, AC-Superpositions between only these.

Secondly, note that, we can choose the ordering between two constants in K on-the-fly. As an optimization we could always choose it in a way so as to minimize the applications of AC-collapse and composition later. In other words, when we need to choose the orientation for $c \approx d$, we can look at the number of occurrences of c and d in the set of D- and AC-rules (in the R-component of the state), and the constant with fewer occurrences is made larger.

6 Construction of Ground Convergent AC Rewrite Systems

We next discuss the problem of obtaining a ground convergent AC rewrite system for the given ground AC-theory *in the original signature*. Hence, now we focus our attention to the problem of transforming a convergent system over an extended signature to a convergent system in the original signature. If we can successfully achieve this goal, it would mean that we can do ground AC-completion *without* having an AC-compatible ordering.

In the case when there are no AC-symbols in the signature, we can construct ground convergent systems from any given abstract congruence closure. This gives an indirect way to construct ground convergent systems equivalent to a given set of ground equations. However, we run into problems when we use the same method for translation in presence of AC-symbols. Typically, after translating back, the set of rules one gets is usually non-terminating modulo AC (Example 12). But if we suitably define the notion of AC-rewriting, the rules are seen to be convergent in the new definition. This is useful in two ways: (i) the new notion of AC-rewriting seems to be more practical, in the sense that it involves strictly less work than a usual $AC \backslash R^e$ reduction [6]; and, (ii) it helps to clarify the advantage offered by the use of extended signatures when dealing with a set of ground AC-equations.

Transition Rules

We describe the process of transforming a rewrite system R over an extended signature $\Sigma \cup K$ to a rewrite system R over the original signature Σ by transformation rules on states (K, R), where K is the set of constants to be eliminated, and R is a set of rewrite rules over $\Sigma \cup K$ to be transformed.

Constants in K that occur as a left-hand side of some rule in R are called *redundant* in R. Redundant constants can be easily eliminated by the compression rule.

Compression: $$\frac{(K \cup \{c\}, R \cup \{c \to t\})}{(K, R')}$$

if R' is obtained from R by replacing all occurrences of c in R by t.

The basic idea for eliminating a constant c that is not redundant in R involves picking a representative term t (over the signature Σ) in the equivalence class of c, and replacing c by t everywhere in R. The selection rule achieves this.

Selection: $$\frac{(K \cup \{c\}, R \cup \{t \to c\})}{(K, R')}$$

where c is not redundant in R, $t \in \mathcal{T}(\Sigma)$, R' is the set $\{l\sigma \to r\sigma : l \to r \in R\} \cup \{f(t_1, \ldots, t_k, X) \to f(f(t_1, \ldots, t_k), X) : t \equiv f(t_1, \ldots, t_k) \text{ and } f \in \Sigma_{AC}\}$, and σ is the substitution $\{c \mapsto t\}$. After applying the substitution, we do *not* flatten terms. The variable X is a special variable and its role will be be discussed later.

Example 12. Consider the problem of constructing a ground convergent system for the set E_0 of Example 4. A fully-reduced congruence closure for E_0 is given by the set R_0

$$\begin{array}{cccc} a \to c_1 & b \to c_2 & c \to c_3 & fc_2c_3 \to c_4 \\ fc_3c_4 \to c_2 & fc_1c_3 \to c_1 & fc_2c_2 \to fc_4c_4 & fc_1c_2 \to fc_1c_4 \\ gc_4 \to c_4 & & & \end{array}$$

under the ordering $c_2 \succ c_4$ between constants. For the constants c_1, c_2 and c_3 we have no choice but to choose a, b and c as representatives respectively. Thus after three applications of selection, we get

$$\begin{array}{cccc} fcc_4 \to b & fac \to a & fbb \to fc_4c_4 & fab \to fac_4 \\ fbc \to c_4 & gc_4 \to c_4 & & \end{array}$$

Next we are forced to choose fbc as the representative for the class c_4. This gives us the transformed set R_1,

$$\begin{array}{cccc} fc(fbc) \to b & fac \to a & fbb \to f(fbc)(fbc) & fab \to fa(fbc) \\ fbcX \to f(fbc)X & gfbc \to fbc & & \end{array}$$

The relation $\to_{AC\backslash R_1^e}$ is clearly non-terminating (with the variable X considered as a regular term variable).

Rewriting with sequence extensions modulo permutation congruence

Let X denote a variable ranging over sequences of terms. A sequence substitution σ is a substitution that maps variables to the sequences. If σ is a sequence substitution that maps X to the sequence $\langle s'_1, \ldots, s'_m \rangle$, then $f(s_1, \ldots, s_k, X)\sigma$ is the term $f(t_1, \ldots, s_k, s'_1, \ldots, s'_m)$.

Definition 13. *Let ρ be a ground rule of the form $f(t_1, \ldots, t_k) \to g(s_1, \ldots, s_m)$ where $f \in \Sigma_{AC}$. We define the* sequence extension *ρ^s of ρ as, $f(t_1, \ldots, t_k, X) \to f(s_1, \ldots, s_m, X)$ if $f = g$ and, $f(t_1, \ldots, t_k, X) \to f(g(s_1, \ldots, s_m), X)$ if $f \neq g$.*

Now we are ready to define the notion of rewriting we use. Recall that P denotes the equations defining the permutation congruence, and that $AC = F \cup P$.

Definition 14. *Let R be a set of ground rules. For ground terms $s, t \in \mathcal{T}(\Sigma)$, we say that $s \to_{P\backslash R^s} t$ if there exists a rule $l \to r \in R^s$ and a sequence substitution σ such that $s = C[l']$, $l' \leftrightarrow_P^* l\sigma$ and $r' = r\sigma$.*

Note that the difference with standard rewriting modulo AC is that instead of performing matching modulo AC, we do matching modulo P. For example, if ρ is $fac \to a$, then the term $f(f(a,b),c)$ is not reducible by $\to_{P\backslash\rho^s}$, although it is reducible by $\to_{AC\backslash\rho^e}$. The term $f(f(a,b),c,a)$ can be rewritten by $\to_{P\backslash\rho^s}$ to $f(f(a,b),a)$.

Example 15. Following up on Example 12, we note that the relation $P\backslash R_1^s$ is convergent. For instance, a normalizing rewrite derivation for the term $fabc$ is,

$$fabc \to_{P\backslash R_1^s} fa(fbc)c \to_{P\backslash R_1^s} fab \to_{P\backslash R_1^s} fa(fbc).$$

On closer inspection, we find that we are essentially doing a derivation in the original rewrite system R_0 (over the extended signature).

$$fabc \to_{P\backslash R_0^s} fc_1c_2c_3 \to_{P\backslash R_0^s} fc_1c_4c_3 \to_{P\backslash R_0^s} fc_1c_2 \to_{P\backslash R_0^s} fc_1c_4.$$

There is a one-to-one bijection between a step using $P\backslash R_1^s$ and a step using $P\backslash R_0^s$. This essentially is at the core of the proof of correctness.

Correctness

We note that any derivation starting in the state (K,R), where R defines an AC-congruence closure over Σ and K, is finite. This is because K is finite, and each application of compression and selection reduces the cardinality of K by one. Furthermore, in any intermediate state (K,R), R is always a rewrite system over $\Sigma \cup K$. Hence, in the final state (K_∞, R_∞), if $K_\infty = \emptyset$, then, R_∞ is a rewrite system over Σ, the original signature. We will show that K_∞ is actually $\emptyset$, and that the reduction relation $P\backslash R_\infty^e$ is terminating on $\mathcal{T}(\Sigma)$ and confluent on fully flattened terms in $\mathcal{T}(\Sigma)$.

Lemma 16. *Suppose (K_1,R_1) is obtained from (K_0,R_0) using one application of the AC-Selection or AC-Compression rule. If R_0 is left-reduced and terminating (with respect to $P\backslash R_0^s$-reduction) then R_1 is left-reduced and terminating (with respect to $P\backslash R_1^s$-reduction). Additionally, for all flat terms $s,t \in \mathcal{T}(\Sigma)$, $s \leftrightarrow^*_{P\backslash R_0^s} t$, if and only if, $s \leftrightarrow^*_{P\backslash R_1^s} t$.*

Proof. First consider the case of AC-Compression. Say $(K_0 = K_1 \cup \{c\}, R_0 = R_0' \cup \{c \to t\}) \vdash (K_1, R_1 = R_0'[c \mapsto t])$. This step has the same effect as a sequence of composition steps followed by a deletion step. Hence, it perserves termination. The left-hand side terms do not change, and hence the system continues to remain left-reduced. Note that R_0 and $R_1 \cup \{c \approx t\}$ define the same equational theory. But since c does not occur in R_1, the equational theories induced by $R_0^s \cup P$ and by $R_1^s \cup P$ over $\mathcal{T}(\Sigma)$ are identical.

The interesting case is that of AC-Selection. Say $(K_0 = K_1 \cup \{c\}, R_0 = R_0' \cup \{t \to c\}) \vdash (K_1, R_1 = R_0'[c \mapsto t] \cup R_1')$. Let σ denote the substitution $\langle c \mapsto t\rangle$ and σ^{-1} denote the substitution $\langle t \mapsto c\rangle$. The crucial observation is that if $l_1 \to r_1$ is a rule in R_1 obtained from $l_0 \to r_0$ in R_0, then, for *any* context $C[]$ and *any* sequence substitution σ^s, we have, $(C[l_1\sigma^s])\sigma^{-1} = (C\sigma^{-1})[l_1\sigma^{-1}(\sigma^s\sigma^{-1})] = (C\sigma^{-1})[l_0(\sigma^s\sigma^{-1})]$. If R_1 is not left-reduced, then, by application of σ^{-1} to the left-hand sides of the overlapping rules we obtain two rules in R_0 that overlap.

Additionally, if $C[l_1\sigma^s] \rightarrow_{P\backslash R_1^s} C[r_1\sigma^s]$, then,

$$(C[l_1\sigma^s])\sigma^{-1} = (C\sigma^{-1})[l_0(\sigma^s\sigma^{-1})] \rightarrow_{P\backslash R_0^s} (C\sigma^{-1})[r_0(\sigma^s\sigma^{-1})] \rightarrow^*_{P\backslash R_0^s} (C[r_1\sigma^s])\sigma^{-1}.$$

Hence, every sequence of n rewrite steps in $P\backslash R_1^s$ can be projected to n or more steps in $P\backslash R_0^s$ (by applying substitution σ^{-1} to each term in the proof). Hence, if $P\backslash R_0^s$ is terminating, so is $P\backslash R_1^s$.

Finally, since the constant c does not occur in R_1 any more, it is easily established that for all flat terms $s, t \in \mathcal{T}(\Sigma)$, $s \leftrightarrow^*_{P\backslash R_0^s} t$, if and only if, $s \leftrightarrow^*_{P\backslash R_1^s} t$.

Theorem 17. *If $(\emptyset, R_\infty)$ is the final state of a maximal derivation starting from state (K, R), where R is a fully-reduced AC-congruence closure, then $\rightarrow_{P\backslash R_\infty^s}$ is ground convergent on all fully flattened terms over Σ. The equivalence over flattened $\mathcal{T}(\Sigma)$ terms defined by this relation is the same as the equational theory induced by $R \cup AC$ over flattened $\mathcal{T}(\Sigma)$ terms.*

Note that in the special case when Σ_{AC} is empty, the notion of rewriting corresponds to the standard notion, and hence R_∞ is convergent in the standard sense by this theorem.

7 Conclusion

The fact that we can construct an AC-congruence closure implies that the word problem for finitely presented ground AC-theories is decidable, see [11], [13] and [8]. Note that we arrive at this result *without* assuming the existence of an AC-simplification ordering that is total on ground terms. The existence of such AC-simplification orderings was established in [13], but required a non-trivial proof.

Since we construct a convergent rewrite system, even the problem of determining whether two finitely presented ground AC-theories are equivalent, is decidable. Since commutative semigroups are special kinds of AC-theories, where the signature consists of a single AC-symbol and a finite set of constants, these results carry over to this special case [12], [10].

We have shown that we can construct an AC-congruence closure for a finite set E of ground equations (over a signature Σ containing AC-symbols) using procedures that construct such closures for finitely presented commutative semigroups. In fact the number of invocations of the latter procedure is bounded above by $n|\Sigma|$, where n denotes the size of set E, and $|\Sigma|$ denotes the cardinality of the set Σ. This establishes that the word problem for ground AC-theories is no more difficult than the word problem for finitely presented commutative semigroups (modulo a polynomial time factor).

The idea of using variable abstraction to transform a set equations over several AC-symbols into a set of equations in which each equation contains exactly one AC-symbol appears in [8]. All equations containing the same AC-symbol are separated out, and completed into a canonical rewriting system (modulo AC) using the method proposed in [4]. However, the combination of ground AC-theories with other ground theories is done differently here. In [8], the ground theory (non-AC part) is handled using ground completion (and uses an recursive path ordering during completion). We, on the other hand, use a congruence closure. The usefulness of our approach can also be seen from the simplicity of the correctness proof and the results we obtain for transforming a convergent system over an extended signature to one over the original signature.

The method for completing a finitely presented commutative semigroup (using what we call AC-rules here) has been described in various forms in the literature, e.g. [4])[4]. It is essentially a specialization of Buchberger's algorithm for polynomial ideals to the case of binomial ideals (i.e. when the ideal is defined by polynomials consisting of exactly two monomials with coefficients $+1$ and -1).

Acknowledgements. We wish to thank the anonymous reviewers for helpful comments.

References

1. L. Bachmair. *Canonical Equational Proofs.* Birkhäuser, Boston, 1991.
2. L. Bachmair and N. Dershowitz. Completion for rewriting modulo a congruence. *Theoretical Computer Science*, 67(2 & 3):173–201, Oct. 1989.
3. L. Bachmair, C. Ramakrishnan, I. Ramakrishnan, and A. Tiwari. Normalization via rewrite closures. In P. Narendran and M. Rusinowitch, editors, *Proc. 10th Int. Conf. on Rewriting Techniques and Applications*, volume 1631 of *Lecture Notes in Computer Science*, pages 190–204. Springer-Verlag, Trento, Italy, 1999.
4. A. M. Ballantyne and D. S. Lankford. New decision algorithms for finitely presented commutative semigroups. *Comp. and Maths. with Appls.*, 7:159–165, 1981.
5. T. Becker and V. Weispfenning. *Gröbner bases: a computational approach to commutative algebra.* Springer-Verlag, Berlin, 1993.
6. N. Dershowitz and J. P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science (Vol. B: Formal Models and Semantics)*, Amsterdam, 1990. North-Holland.
7. N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, 1979.
8. E. Domenjoud and F. Klay. Shallow AC theories. In *Proceedings of the 2nd CCL Workshop*, La Escala, Spain, Sept. 1993.

[4] Actually there is a subtle difference between the proposed method here and the various other algorithms for deciding the word problem for commutative semigroups too. For example, working with rule extensions is not the same as working with rules on equivalence classes (under AC) of terms. Hence, in our method, we can apply certain optimizations as mentioned in Section 5.

9. D. Kapur. Shostak's congruence closure as completion. In H. Comon, editor, *Proceedings of the 8th International Conference on Rewriting Techniques and Applications*, pages 23–37, 1997. Vol. 1232 of *Lecture Notes in Computer Science*, Springer, Berlin.
10. U. Koppenhagen and E. W. Mayr. An optimal algorithm for constructing the reduced Gröbner basis of binomial ideals. In Y. D. Lakshman, editor, *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 55–62, 1996.
11. C. Marche. On ground AC-completion. In R. V. Book, editor, *4th International Conference on Rewriting Techniques and Applications*, pages 411–422, 1991. Vol. 488 of *Lecture Notes in Computer Science*, Springer, Berlin.
12. E. W. Mayr and A. R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Advances in Mathematics*, 46:305–329, 1982.
13. P. Narendran and M. Rusinowitch. Any ground associative-commutative theory has a finite canonical system. In R. V. Book, editor, *4th International Conference on Rewriting Techniques and Applications*, pages 423–434, 1991. Vol. 488 of *Lecture Notes in Computer Science*, Springer, Berlin.
14. G. Nelson and D. Oppen. Simplification by cooperating decision procedures. *ACM Transactions on Programming Languages and Systems*, 1(2):245–257, October 1979.
15. G. Nelson and D. Oppen. Fast decision procedures based on congruence closure. *Journal of the Association for Computing Machinery*, 27(2):356–364, Apr. 1980.
16. G. E. Peterson and M. E. Stickel. Complete sets of reductions for some equational theories. *J. ACM*, 28(2):233–264, Apr. 1981.
17. C. Ringeissen. Cooperation of decision procedures for the satisfiability problem. In F. Baader and K. U. Schulz, editors, *Frontiers of Combining Systems*, pages 121–139, 1996. Kluwer Academic Publishers.
18. A. Rubio and R. Nieuwenhuis. A precedence-based total AC-compatible ordering. In C. Kirchner, editor, *Proceedings of the 5 Intl. Conference on Rewriting Techniques and Applications*, pages 374–388, 1993. Vol. 690 of *Lecture Notes in Computer Science*, Springer.
19. C. Tinelli and M. Harandi. A new correctness proof of the Nelson-Oppen combination procedure. In F. Baader and K. U. Schulz, editors, *Frontiers of Combining Systems*, pages 103–119, 1996. Kluwer Academic Publishers.