Rewrite Closure for Ground and Cancellative AC Theories*

Ashish Tiwari

SRI International,
333 Ravenswood Ave,
Menlo Park, CA, U.S.A
tiwari@csl.sri.com

Abstract. Given a binary relation $\mathbb{E} \cup \mathbb{R}$ on the set of ground terms over some signature, we define an abstract rewrite closure for $\mathbb{E} \cup \mathbb{R}$. An abstract rewrite closure can be interpreted as a specialized ground tree transducer (pair of bottom-up tree automata) and can be used to efficiently decide the reachability relation $\rightarrow^*_{\mathbb{E} \cup \mathbb{E}^- \cup \mathbb{R}}$. It is constructed using a completion like procedure. Correctness is established using proof ordering techniques. The procedure is extended, in a modular way, to deal with signatures containing cancellative associative commutative function symbols.

1 Introduction

Completion techniques for term rewriting systems, which are typically used for reasoning about congruence relations, have been extended in recent years to deal with non-symmetric relations. The general theory was outlined in [11] and sound and refutationally complete inference systems were obtained for dealing with partial congruence and partial equivalence relations [4]. Usually one obtains suitably restricted (via ordering restrictions) chaining calculi. The gain in efficiency with an ordered system over the unordered variants of chaining are comparable to the improvements achieved by superposition over unrestricted paramodulation.

This paper presents a completion based approach to *decide* the rewrite relation induced by a set of directed (i.e., non-symmetric) and undirected (i.e., symmetric) ground equations. The basic technique involves combining standard completion (for undirected equations) with non-symmetric completion (for directed equations). Standard completion is reflected in a *superposition* inference rule that deduces critical pairs between undirected equations. Non-symmetric completion yields a *chaining* inference rule to deduce critical pairs between directed equations. Finally, the interaction between the two kinds of equations is captured using a *paramodulation* inference rule. We first consider the problem of

* This research was supported in part by the National Science Foundation under grants CCR-9902031 and CCR-0082560, and NASA Langley Research Center under contract NAS1-00079.

constructing a "convergent system", called a *rewrite closure*, for a set of ground (directed and undirected) equations. Subsequently, we extend the method to allow for cancellative associative commutative function symbols in the signature. If all input equations are undirected, then the problem reduces to the construction of congruence closure and hence, an abstract rewrite closure is a generalization of an abstract congruence closure [6].

The reachability or rewrite relation induced by a ground term rewriting system was shown to be decidable in [9] and [13] using, respectively, tree automata techniques and explicit transitive closure computation. An abstract rewrite closure can be interpreted as a specialized "ground tree transducer" (GTT). In this paper, we give a set of abstract completion-like inference rules for construction of rewrite closures. These rules yield efficient algorithms under suitable strategies. Moreover, our method is extendible to richer signatures.

Correctness of the inference system is established using proof ordering techniques. Each proof is assigned a measure and all inference rules transform a proof with a larger measure into a proof with a smaller measure. The desired form of proof, for example a rewrite proof or a valley proof, is assigned a minimal measure. Correctness arguments based on proof orderings also show compatibility of the inference systems with certain kinds of simplifications.

Apart from our interest in extending rewriting techniques to non-symmetric relations, this work is also motivated by our interest in developing abstract transformation rules for constraint solving. Typical constraints consist of equational constraints, which are solved by a unification procedure, and ordering constraints, where the ordering is usually some kind of a path ordering. Almost all such orderings are rewrite relations that also satisfy certain additional properties, and hence an efficient procedure for deciding rewrite relations is a crucial first step [13]. Note that the cancellative axiom for AC symbols is satisfied by any AC compatible total simplification ordering.

Preliminaries

Let Σ be a set, called a *signature*, with an associated *arity function* $\alpha : \Sigma \rightarrow 2^{\mathbb{N}}$ and let $\mathcal{V}$ be a disjoint (denumerable) set. We define $\mathcal{T}(\Sigma, \mathcal{V})$ as the smallest set containing $\mathcal{V}$ and such that $f(t_1, \ldots, t_n) \in \mathcal{T}(\Sigma, \mathcal{V})$ whenever $f \in \Sigma, n \in \alpha(f)$ and $t_1, \ldots, t_n \in \mathcal{T}(\Sigma, \mathcal{V})$. The elements of the sets Σ, $\mathcal{V}$ and $\mathcal{T}(\Sigma, \mathcal{V})$ are respectively called *function symbols*, *variables* and *terms* (over Σ and $\mathcal{V}$). Elements c in Σ for which $\alpha(c) = \{0\}$ are called *constants*. By $\mathcal{T}(\Sigma)$ we denote the set $\mathcal{T}(\Sigma, \emptyset)$ of all variable-free, or *ground* terms. The symbols $s, t, u, \ldots$ are used to denote terms; $f, g, \ldots$, function symbols; and $x, y, z, \ldots$, variables.

An *(undirected) equation* is an unordered pair of terms, written $s \approx t$. A *directed equation* or *rule* is an ordered pair of terms, written $s \rightarrow t$. If $\mathcal{E}$ is a set of rules, then we define $\mathcal{E}^- = \{s \rightarrow t : t \rightarrow s \in \mathcal{E}\}$ and $\mathcal{E}^{\pm} = \mathcal{E} \cup \mathcal{E}^-$. A set $\mathcal{E}$ of rules is called a *rewrite system* and the *rewrite relation* $\rightarrow_{\mathcal{E}}$ induced by $\mathcal{E}$ is defined by: $u \rightarrow_{\mathcal{E}} v$ if, and only if, $u = u[l\sigma]$, $v = u[r\sigma]$ is obtained by replacing $l\sigma$ by $r\sigma$ in u, $l \rightarrow r$ is in $\mathcal{E}$, and σ is some substitution. If $\rightarrow$ is a binary relation, then $\leftarrow$ denotes its inverse, $\leftrightarrow$ its symmetric closure, $\rightarrow^+$ its transitive closure

and $\rightarrow^*$ its reflexive-transitive closure. A set of rules $\mathcal{E}$ is *terminating* if there exists no infinite reduction sequence $s_0 \rightarrow_{\mathcal{E}} s_1 \rightarrow_{\mathcal{E}} s_2 \cdots$ of terms.

We will mostly be interested in ground rewrite systems, denoted by non-calligraphic symbols $I\!E, I\!R$. In Section 2, the arity $\alpha(f)$ of a symbol $f \in \Sigma$ is assumed to a singleton and we focus on (the transitive closure of) the rewrite relation $\rightarrow^*_{I\!E^{\pm} \cup I\!R}$ induced by the ground rewrite system $I\!E \cup I\!R$ over such a signature Σ. In Section 3, we shall assume that $\Sigma_{AC} \subset \Sigma$ is a set of AC symbols. Such symbols are varyadic, with arity $\alpha(f) = \{2, 3, 4, \ldots\}$ for $f \in \Sigma_{AC}$. If $f \in \Sigma_{AC}$, then the *extension* of a rule $f(s_1, s_2) \rightarrow t$, call it ρ, is defined as $f(f(s_1, s_2), x) \rightarrow f(t, x)$ and is denoted by ρ^e. Given a rewrite system $\mathcal{R}$, by $\mathcal{R}^e$ we denote the set $\mathcal{R}$ plus extensions of rules in $\mathcal{R}$. By $AC \backslash \mathcal{R}$ we denote the rewrite system consisting of all rules $u \rightarrow v$ such that $u \leftrightarrow^*_{AC} u'\sigma$ and $v = v'\sigma$, for some rule $u' \rightarrow v'$ in $\mathcal{R}$ and some substitution σ.

A *proof* of $s \rightarrow t$ (in $\mathcal{E}$) is a finite sequence $s = s_0 \rightarrow_{\mathcal{E}} s_1$, $s_1 \rightarrow_{\mathcal{E}} s_2$, $\cdots, s_{k-1} \rightarrow_{\mathcal{E}} s_k = t$ $(k \geq 0)$, which is usually written in abbreviated form as $s = s_0 \rightarrow_{\mathcal{E}} s_1 \rightarrow_{\mathcal{E}} \cdots \rightarrow_{\mathcal{E}} s_k = t$ $(k \geq 0)$.

2 Abstract Rewrite Closure

We closely follow the idea of an abstract congruence closure [6] in defining the notion of an abstract rewrite closure. More specifically, we *flatten* out terms via introduction of new constants and corresponding definitions.

Definition 1. *Let Σ be a signature and K be a set of constants disjoint from Σ. A* D-rule *(with respect to Σ and K) is a rewrite rule of the form $f(c_1, \ldots, c_k) \rightarrow c$ where $f \in \Sigma$ is a k-ary function symbol and $c_1, \ldots, c_k, c$ are constants in set K. A rewrite rule of the form $c \rightarrow f(c_1, \ldots, c_k)$ will be called a* reverse D-rule.

A C-rule *(with respect to K) is a rule $c \rightarrow d$, where c and d are constants in K.*

A set of D-rules and C-rules (with respect to Σ and K) is a specification of a bottom-up tree automaton transitions [8]. The set K represents "states" in the tree automaton. Thus, D-rules and C-rules represent regular and ϵ-transitions respectively. A set of ground equations and rules, say $I\!I_0 = I\!E_0 \cup I\!R_0$, where $I\!E_0 = \{f(g(a, b), g(a, b)) \approx a\}$ and $I\!R_0 = \{a \rightarrow b\}$, can be represented as $I\!I_1 = \{f(c_3, c_3) \approx c_1,\ c_1 \rightarrow c_2\}$ by introducing the set $E_1 = \{a \rightarrow c_1, b \rightarrow c_2, g(c_1, c_2) \rightarrow c_3\}$ of D-rules.

A constant c in K is said to *represent* a term t in $\mathcal{T}(\Sigma)$ via the rewrite system E if $t \leftrightarrow^*_E c$. For example, the constant c_3 represents the term $g(a, b)$ via E_1.

Definition 2 (Abstract rewrite closure). *Let Σ be a signature and K be a set of constants disjoint from Σ. A ground rewrite system $E \cup F \cup B$ is said to be an* (abstract) rewrite closure *(with respect to Σ and K) if*

(i) E and F are both sets of D-rules and C-rules, B is a set of reverse D-rules and C-rules such that each constant $c \in K$ represents some term $t \in \mathcal{T}(\Sigma)$ via E, and

(ii) the rewrite systems $E \cup F$ and $E \cup B^-$ are terminating; and for all terms $s, t \in \mathcal{T}(\Sigma)$, if $s \rightarrow^*_{E^\pm \cup F \cup B} t$ then $s \rightarrow^*_{E \cup F} \circ \leftarrow^*_{E \cup B^-} t$.
Moreover, if $\mathit{\Pi} = \mathit{E} \cup \mathit{R}$ is a set of ground equations and rules over $\mathcal{T}(\Sigma)$ such that

(iii) for all terms s and t in $\mathcal{T}(\Sigma)$, $s \rightarrow^*_{\mathit{\Pi}^\pm \cup \mathit{R}} t$ if and only if $s \rightarrow^*_{E^\pm \cup F \cup B} t$, then $E \cup F \cup B$ will be called an (abstract) rewrite closure for *(the rewrite relation induced by) $\mathit{\Pi}$.*

From the set $E \cup F \cup B$, one can obtain a pair of (bottom-up) tree automata [8]: the set $E \cup F$ defines the transitions of the first automaton and the set $E \cup B^-$ defines the transitions of the second automaton (over the same set K of "states"). Such a pair defines a binary relation $\rightarrow^*_{E \cup F} \circ \leftarrow^*_{E \cup B^-}$ on the set of ground terms and is called a "ground tree transducer" in the tree automata literature [8].

Using a combination of standard completion and non-symmetric completion, which we present next, we can obtain a rewrite closure $E_2 \cup F_2 \cup B_2$ for the set $\mathit{\Pi}_0 = \mathit{E}_0 \cup \mathit{R}_0$, where $E_2 = \{a \rightarrow c_1,\ b \rightarrow c_2,\ g(c_1, c_2) \rightarrow c_3,\ f(c_3, c_3) \rightarrow c_1\}$, $F_2 = \{c_1 \rightarrow c_2\}$, and $B_2 = \{c_3 \rightarrow g(c_2, c_2)\}$. A rewrite closure for $\mathit{E} \cup \mathit{R}$ gives a decision procedure for (deciding) the rewrite relation $\rightarrow^*_{\mathit{E}^\pm \cup \mathit{R}}$.

Construction of Rewrite Closure

We next present an inference system to construct a rewrite closure for a finite set $\mathit{\Pi}$ of ground equations and rules over the signature Σ. Our description is fairly abstract, in terms of transition rules that operate on tuples $(\mathit{\Pi}, E, R)$, where $\mathit{\Pi} = \mathit{E} \cup \mathit{R}$ is a set of ground equations and rules (over Σ), and E and R[1] are sets of (reverse) D-rules and C-rules. Tuples represent possible *states* in the process of constructing a rewrite closure. The initial state is $(\mathit{\Pi}_0, \emptyset, \emptyset)$, where $\mathit{\Pi}_0$ is the input set of ground equations and rules (over $\mathcal{T}(\Sigma)$).

The transition rules can be derived from those for standard completion and non-symmetric completion as described in [3] and [11], with some differences so that a system is constructed over an *extended* signature. We assume that the new constants are chosen from an infinite set U disjoint from Σ, which is endowed with an ordering[2] $\succ_U$.

Equations and rules are flattened using extension and simplification.

Extension:
$$\frac{(\mathit{\Pi}[s], E, R)}{(\mathit{\Pi}[c], E \cup \{s \rightarrow c\}, R)}$$

if $s \rightarrow c$ is a D-rule and $c \in U$ is a new constant[3].

Simplification1:
$$\frac{(\mathit{\Pi}[s], E \cup \{s \rightarrow c\}, R)}{(\mathit{\Pi}[c], E \cup \{s \rightarrow c\}, R)}$$

[1] The set R will later be partitioned into the set F of forward rules and the set B of backward rules.

[2] By an *ordering* we mean any irreflexive and transitive relation on terms.

[3] The notation $\mathit{\Pi}[s]$ denotes that s occurs as a subterm in some equation or rule in $\mathit{\Pi}$ and $\mathit{\Pi}[c]$ denotes the new set obtained by replacing that occurrence of s in $\mathit{\Pi}$ by c.

Once an equation or rule in $\mathcal{I}$ is of the form of a D-rule, reverse D-rule, or a C-rule, it can be oriented.

$$\textbf{Ori}\text{entation:} \qquad \frac{(\mathcal{I} \cup \{s \approx c\}, E, R)}{(\mathcal{I}, E \cup \{s \to c\}, R)} \qquad \frac{(\mathcal{I} \cup \{u \to v\}, E, R)}{(\mathcal{I}, E, R \cup \{u \to v\})}$$

if $s \to c$ is either a D-rule or a C-rule with $s \succ_U c$ and $u \to v$ is either a C-rule, D-rule, or a reverse D-rule.

Trivial equations and rules are deleted.

$$\textbf{Del}\text{etion:} \qquad \frac{(\mathcal{I} \cup \{s \approx s\}, E, R)}{(\mathcal{I}, E, R)} \qquad \frac{(\mathcal{I} \cup \{s \to s\}, E, R)}{(\mathcal{I}, E, R)} \qquad \frac{(\mathcal{I}, E, R \cup \{s \to s\})}{(\mathcal{I}, E, R)}$$

Deduction in standard completion, as well as in non-symmetric completion, is based on computation of critical pairs. There are three kinds of critical pair computations—(i) between two rules in E, which are handled by superposition; (ii) between a rule in E and a rule in R, which are handled by paramodulation; and (iii) between two rules in R, which are handled by chaining.

$$\textbf{Sup}\text{erposition:} \qquad \frac{(\mathcal{I}, E \cup \{t \to c, s[t] \to d\}, R)}{(\mathcal{I}, E \cup \{t \to c, s[c] \to d\}, R)} \qquad \frac{(\mathcal{I}, E \cup \{t \to c, t \to d\}, R)}{(\mathcal{I}, E \cup \{t \to c, d \to c\}, R)}$$

if $s[t] \neq t$ in the first case and $d \succ_U c$ in the second case.

The set R can be partitioned into the set $F = \{s \to t \in R : s \to t$ is a D-rule or a C-rule with $s \succ_U t\}$ of *forward* rules and the set $B = \{s \to t \in R : t \to s$ is a D-rule or a C-rule with $t \succ_U s\}$ of *backward* rules.

Definition 3. *Let E and $R = F \cup B$ be sets of (reverse) D-rules and C-rules. The set $CP(E, R)$ of critical pairs between rules in E and R is defined as:*

$$\begin{aligned} CP(E, R) = \{&f(\ldots, d, \ldots) \to c : f(\ldots, d', \ldots) \to c \in E \text{ and } d \to d' \in B\} \\ \cup \{&c \to f(\ldots, d, \ldots) : f(\ldots, d', \ldots) \to c \in E \text{ and } d' \to d \in F\}. \end{aligned}$$

The set $CP(R)$ of critical pairs between rules in R is defined as:

$$\begin{aligned} CP(R) = \{&t[d] \to c : d \to s \in B \text{ and } t[s] \to c \in F\} \\ \cup \{&c \to t[d] : c \to t[s] \in B \text{ and } s \to d \in F\}. \end{aligned}$$

Note that if the sets E and R contain only D-rules, reverse D-rules, and C-rules, then so do the sets $CP(E, R)$ and $CP(R)$.

$$\textbf{Cha}\text{ining and } \textbf{Par}\text{amodulation:} \qquad \frac{(\mathcal{I}, E, R)}{(\mathcal{I}, E, R \cup \{s \to t\})}$$

if $s \to t \in CP(R) \cup CP(E, R)$.

A crucial component of deductive inference systems is simplification. In the ground case, several deduction steps reduce to simplification. In particular, the rules in E can be used to simplify terms in R.

Simplification2:
$$\frac{(\mathit{I\!I}, E \cup \{s \to c\}, R[s])}{(\mathit{I\!I}, E \cup \{s \to c\}, R[c])}$$

Composition:
$$\frac{(\mathit{I\!I}, E \cup \{c \to d, s \to c\}, R)}{(\mathit{I\!I}, E \cup \{c \to d, s \to d\}, R)}$$

Example 1. Consider the set $\mathit{I\!I}_0 = \{f(g(a,b), g(a,b)) \approx a, a \to b\}$ of equations and rules. An abstract rewrite closure for $\mathit{I\!I}_0$ can be derived from $(\mathit{I\!I}_0, E_0, R_0) = (\mathit{I\!I}_0, \emptyset, \emptyset)$ as follows (assuming $U = \{c_0, c_1, c_2, \ldots\}$ with $c_i \succ_U c_j$ for $i < j$):

i	Input $\mathit{I\!I}_i$	Equations E_i	Rules R_i	Transition Rule
0	$\mathit{I\!I}_0$	$\emptyset$	$\emptyset$	
1	$\{fgabgab \approx a\}$	$\{a \to c_1, b \to c_2\}$	$\{c_1 \to c_2\}$	$\mathbf{Ext}^2 \circ \mathbf{Ori}$
2	$\{fc_3c_3 \approx a\}$	$E_1 \cup \{gc_1c_2 \to c_3\}$	R_1	$\mathbf{Sim}^4 \circ \mathbf{Ext} \circ \mathbf{Sim}$
3	$\emptyset$	$E_2 \cup \{fc_3c_3 \to c_1\}$	R_1	$\mathbf{Sim} \circ \mathbf{Ori}$
4	$\emptyset$	E_3	$R_1 \cup \{c_3 \to gc_2c_2\}$	$\mathbf{Par}$

Since no further rules are added, the rewrite system $E_4 \cup F_4 \cup B_4$, where $F_4 = \{c_1 \to c_2\}$ and $B_4 = \{c_3 \to gc_2c_2\}$, is an abstract rewrite closure for $\mathit{I\!I}_0$.

Correctness

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 $(\mathit{I\!I}_0, E_0, R_0) \vdash (\mathit{I\!I}_1, E_1, R_1) \vdash \cdots$. A derivation is said to be *fair* if any transition rule which is continuously enabled is eventually applied. The set E_∞ of *persisting* rules is defined as $\cup_i \cap_{j>i} E_j$; and similarly, $R_\infty = \cup_i \cap_{j>i} R_j$.

We shall prove that any fair derivation will only generate finitely many persisting rewrite rules in the second and third components.

Theorem 1. *Let $\mathit{I\!I}_0$ be a finite set of ground equations and rules. The set $E_\infty \cup R_\infty$ of persisting rules in any fair derivation starting from the state $(\mathit{I\!I}_0, \emptyset, \emptyset)$ is finite.*

Proof. Each inference step either reduces, or leaves unchanged, the number of Σ-symbols in the $\mathit{I\!I}$-component. The inference rule which introduces new constants, extension, always reduces this number. Therefore, it follows that the number of new constants introduced in any derivation is finite. Let this number be n.

If the maximum arity of any function symbol in Σ is c, then the number of distinct D-rules is bounded by $|\Sigma|n^{c+1}$ and the number of distinct C-rules is n^2. Consequently, the sets E_∞ and R_∞ are finite. ∎

Theorem 2 (Soundness). *If $(\mathit{I\!E}_0 \cup \mathit{I\!R}_0, E_0, R_0) \vdash (\mathit{I\!E}_1 \cup \mathit{I\!R}_1, E_1, R_1)$, then, the rewrite relation induced by $\mathit{I\!E}_1^{\pm} \cup \mathit{I\!R}_1 \cup E_1^{\pm} \cup R_1$ is identical to the rewrite relation induced by $\mathit{I\!E}_0^{\pm} \cup \mathit{I\!R}_0 \cup E_0^{\pm} \cup R_0$ over the set $\mathcal{T}(\Sigma \cup K_0)$ of terms, where $K_0 \subset U$ is the set of new constants introduced until state $(\mathit{I\!I}_0, E_0, R_0)$.*

Proof Ordering The correctness of the procedure will be established using proof simplification techniques, as described by Bachmair [1] and Bachmair and Dershowitz [2], but specialized to our case of standard and non-symmetric ground completion. Let $\succ$ be any reduction ordering[4] which contains $\succ_U$ and also orients D-rules from left to right. For instance, a recursive path ordering with an appropriate precedence on function symbols is such an ordering.

Let $s = C[u] \to C[v] = t$ be a proof step using the equation or rule $u \approx v \in I\!E^{\pm} \cup I\!R \cup E^{\pm} \cup R$. The complexity of this proof step is defined by

$$\begin{array}{llll} (\{s,t\},\bot,\bot,\bot) & \texttt{if} & u \approx v \in I\!E^{\pm} & (\{s,t\},\bot,\bot,\bot) \ \texttt{if} \ u \to v \in I\!R \\ (\{s\},u,\bot,t) & \texttt{if} & u \to v \in E & (\{s\},u,\top,t) \ \texttt{if} \ u \to v \in R, u \succ v \\ (\{t\},v,\bot,s) & \texttt{if} & u \to v \in E^{-} & (\{t\},v,\top,s) \ \texttt{if} \ u \to v \in R, v \succ u \end{array}$$

where $\bot$ and $\top$ are new symbols assumed to be minimum and maximum respectively. Tuples are compared lexicographically using the multiset extension of the ordering $\succ$ on terms in the first component, and the ordering $\succ$ in the second and fourth component. 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 $\succ_P$. The ordering $\succ_P$ is well-founded as it is a lexicographic combination of well-founded orderings.

Lemma 1. *Suppose* $(I\!I_0, E_0, R_0) \vdash (I\!I_1, E_1, R_1)$. *If* π *is a ground proof,* $s_0 \to s_1 \to \cdots \to s_k$, *in* $I\!E_0^{\pm} \cup I\!R_0 \cup E_0^{\pm} \cup R_0$, *then there is a proof* π', $s_0 = s'_0 \to s'_1 \to \cdots \to s'_l = s_k$, *in* $I\!E_1^{\pm} \cup I\!R_1 \cup E_1^{\pm} \cup R_1$, *such that* $\pi \succeq_{\mathcal{P}} \pi'$.

Proof. We need to check that each equation or rule in $(I\!E_0 - I\!E_1)^{\pm} \cup (I\!R_0 - I\!R_1)^{\pm} \cup (E_0 - E_1) \cup (R_0 - R_1)$ has a simpler proof in $I\!E_1^{\pm} \cup I\!R_1 \cup E_1^{\pm} \cup R_1$ for each transition rule application. The details can be found in [16].

For instance, consider the case of simplification2 inference rule where $s[u] \to t \in R_0$ is simplified to $s[v] \to t \in R_1$ by the rule $u \to v \in E_0$. The old proof $s[u] \to_{R_0} t$ is replaced by the new proof $s[u] \to_{E_1} s[v] \to_{R_1} t$. If $s[u] \succ t$, then the new proof is smaller because the rewrite step $s[u] \to_{R_0} t$ is more complex than (a) the proof step $s[u] \to s[v]$ in either the second component, if $s[u] \neq u$, or the third component; and (b) the proof step $s[v] \to t$ in the first component as $s[u] \succ s[v]$ and $s[u] \succ t$. Next suppose that $t \succ s[u]$. In this case, the old rewrite step $s[u] \to_{R_0} t$ is more complex than (a) the proof step $s[u] \to s[v]$ in the first component as $t \succ s[u]$; and (b) the proof step $s[v] \to t$ in the fourth component as $s[u] \succ s[v]$. ∎

Theorem 3 (Completeness). *Let* $I\!I_0$ *be a finite set of equations and rules. If* $(I\!I_\infty, E_\infty, F_\infty \cup B_\infty)$ *is the persisting state of a fair derivation starting from* $(I\!I_0, \emptyset, \emptyset)$, *then, the rewrite system* $E_\infty \cup F_\infty \cup B_\infty$ *is an abstract rewrite closure for* $I\!I_0$.

Proof. (Sketch) Fairness implies that all superposition, paramodulation, and chaining inferences between rules in E_∞ and R_∞ are contained in the set

[4] A *reduction ordering* is an ordering that is well-founded and closed under contexts.

$(\cup_i E_i) \cup (\cup_i R_i)$. Fairness also implies that $\mathbb{I}_\infty$ is empty. Since the proof ordering is well-founded, it follows from Lemma 1 that for every proof in $\mathbb{E}_i^{\pm} \cup \mathbb{R}_i \cup E_i^{\pm} \cup R_i$, there exists a minimal proof in $E_\infty^{\pm} \cup R_\infty$. We argue by contradiction that peaks, which are proof patterns of the form $s \to u \to t$ with $u \succ s$ and $u \succ t$, can not occur in the minimal proof. This implies that for all terms $s, t \in \mathcal{T}(\Sigma)$, if $s \to^*_{F_\infty \cup B_\infty \cup E_\infty^{\pm}} t$ then $s \to^*_{E_\infty \cup F_\infty} \circ \leftarrow^*_{E_\infty \cup B_\infty^-} t$. Moreover, the rewrite systems $E_\infty \cup F_\infty$ and $E_\infty \cup B_\infty^-$ are terminating as they are contained in $\succ$. Finally, property (i) and (ii) of Definition 2 follow from correctness of congruence closure [6] and Lemma 1. This establishes that $E_\infty \cup F_\infty \cup B_\infty$ is a rewrite closure *for* $\mathbb{E}_0 \cup \mathbb{R}_0$. ■

Related Work and Other Remarks

Note that the relation $\to^*_{E \cup F} \circ \leftarrow^*_{E \cup B^-}$ is decidable as the rewrite systems $E \cup F$ and $E \cup B^-$ are terminating [11]. Although the search for a proof of the above form involves guessing the correct rewrite rules to apply, we can still decide in polynomial time if $s \to^*_{E \cup F} \circ \leftarrow^*_{E \cup B^-} t$, as (i) the non-deterministic choices can be eliminated by maintaining subsets of K, that is, doing subset determinization along the computation, and (ii) the common context $C[\_]$ of terms s and t such that $s \to^*_{E \cup F} C[c_1, \ldots, c_k] \leftarrow^*_{E \cup B^-} t$ can be determined by starting with the largest common context of s and t and moving (only polynomial number of times) to a smaller context if necessary. Furthermore, using the result that establishes a quadratic bound on the length of a derivation for construction of congruence closure [6], we can show that we can reach a state consisting of all persisting rules using derivations of length $O(n^2 + n^{c+1})$, where n is the size of the input and c is the maximum arity[5] of any symbol in Σ. Reachability for ground rewrite systems was shown to be decidable in polynomial time in [13].

The construction of abstract rewrite closure is similar to performing "iterative (or transitive) closure" on a ground tree transducer (representing the one-step rewriting relation). However, there are the following differences: (a) whereas a GTT is specified as a pair of bottom-up tree automata, an abstract rewrite closure has an additional component, E, which keeps track of the term representation[6] and the undirected equations, like $s \approx t$, in the input. Thus, the undirected equations are treated using congruence closure and not as two distinct rules, $s \to t$ and $t \to s$ (as would be done in the GTT approach); (b) our deduction rules are local and have ordering constraints. The computation of an iterative closure for GTT is done using exhaustive closure under the following rule (described in our framework as): "deduce $c \to d$ if $f(c_1, \ldots, c_k) \to c \in E \cup B^-$, $f(c'_1, \ldots, c'_k) \to d \in E \cup F$, and for each i, c_i and c'_i represent some common term in $\mathcal{T}(\Sigma)$"[7]. In [13], *all* possible transitivity inferences are explicitly done; (c) our procedure is

[5] Without loss of generality, the maximum arity c can be treated as a constant.

[6] The D-rules in E (introduced by Extension) are interpreted as representing the term DAG [6].

[7] A stronger requirement (assuming each constant represents some term in $\mathcal{T}(\Sigma)$) is $c_i \leftrightarrow^*_C c'_i$, where C represents all the C-rules in $E \cup F \cup B$. This inference rule

based on standard completion techniques and redundant inferences are avoided; (d) the correctness argument is in terms of proof orderings; and (e) our procedure can be extended to AC symbols, whereas tree automata techniques have not been extended to such richer signatures. We explain the last three points further below.

Correctness arguments based on proof orderings allow for clear identification of redundant inferences and compatible simplifications. To illustrate this point, consider an inference rule $(\mathbb{I}, E, R \cup \{s \to t, t \to s\}) \vdash (\mathbb{I}, E \cup \{s \to t\}, R)$, where $s \succ t$. This inference rule[8] is clearly sound. The completeness of the inference system that includes this rule easily follows by observing that the deleted rules, $s \to t$ and $t \to s$ in the R-component, have simpler proofs using the new rule in the E-component. The new proofs are simpler in the third component.

3 Ground Cancellative *AC* Theories

We next enrich the signature with additional AC symbols Σ_{AC}. Apart from the associative and commutative axioms, the symbols $f \in \Sigma_{AC}$ are assumed to satisfy the *cancellative* axioms (or inverse monotonicity axioms),

$$f(x_1, x_2, \ldots, x_m) \approx f(x_1, y_2, \ldots, y_m) \text{ iff } f(x_2, \ldots, x_m) \approx f(y_2, \ldots, y_m),$$
$$f(x_1, x_2, \ldots, x_m) \to f(x_1, y_2, \ldots, y_m) \text{ iff } f(x_2, \ldots, x_m) \to f(y_2, \ldots, y_m),$$

and the identity axiom $f(x, e_f) \approx x$, where e_f is the identity element for f.

In the presence of AC-symbols, apart from D-rules and C-rules, we additionally require A-rules of the form $f(c_1, c_2, \ldots, c_m) \to f(d_1, d_2, \ldots, d_k)$, where $m, k \in \alpha(f)$. Unlike D-rules and C-rules, A-rules do not correspond to any standard notion of a transition in bottom-up tree automata. The definition of a rewrite closure can be extended by allowing for A-rules and replacing standard rewriting by rewriting modulo AC [16].

We first consider the simple case of cancellative abelian monoid. Let signature $\Sigma = \Sigma_{AC} = \{\cdot\}$ and let $K = \{e, c_1, c_2, \cdots, c_m\}$ be a finite number of constants where e is an identity element for $\cdot$. We denote an application of $\cdot$ by juxtaposition and use exponentiation notation and write, for example, c_1^2 for the term $c_1 \cdot c_1$. Moreover, we denote by $[s, t]$ the term that is the greatest common divisor of s and t. Thus, $[c_1^2 c_2 c_3, c_1 c_2^2 c_4] = c_1 c_2$.

Let $R_0 = \{s_1 \to t_1, s_2 \to t_2, \ldots, s_n \to t_n\}$ be a set of directed rules over the signature $\Sigma \cup K$, where each rule $s_i \to t_i$ is (when fully flattened and reduced using the identity axiom) either a D-rule, a reverse D-rule, a C-rule, or an A-rule. We first show how to "complete" this set. We associate a measure with every rule. The measure will be a vector from the set N^n, where $N = \{0, 1, 2, \ldots\}$ is the set of natural numbers. For the initial set R_0 of rules, we assign measures

is similar in spirit to the inference rule used in Nelson-Oppen congruence closure algorithm [6].

[8] Having the rule $s \to t$ in E is advantageous as rules in E can be used for simplification (see the Simplification2 and Composition inference rules).

as follows: the rule $s_i \to t_i$ is assigned the measure $\boldsymbol{\varepsilon_i} = \langle 0, \ldots, 0, 1, 0, \ldots, 0 \rangle$, where 1 is in exactly the i-th component.

We maintain the invariant that $[l_i, r_i] = e$ for all rules $l_i \to r_i \in R$ and hence, we assume that $[s_i, t_i] = e$ for every $i = 1, 2, \ldots, n$. Let $\succ$ be the lexicographic ordering, or the total degree lexicographic ordering [7].

ACC-Chaining: $$\frac{(\mathcal{I\!I}, E, R \cup \{s \to t, u \to v\})}{(\mathcal{I\!I}, E, R \cup \{s \to t, u \to v, \frac{su}{[s,v][t,u]} \to \frac{tv}{[s,v][t,u]}\})}$$

if $[t, u] \neq e$ and either (a) $t \succ s$ and $u \succ v$; or (b) $s \succ t$, $u \succ v$, and $s \to t$ is not a C-rule; or (c) $t \succ s$, $v \succ u$, and $u \to v$ is not a C-rule. The new rule is assigned the measure $\boldsymbol{\alpha} + \boldsymbol{\beta}$, where $\boldsymbol{\alpha}$ is the measure associated with the rule $s \to t$ and $\boldsymbol{\beta}$ is the measure associated with the rule $u \to v$.

In order to ensure termination, we need to identify and delete redundant rules. The measure vector helps in doing this.

ACC-Collapse: $$\frac{(\mathcal{I\!I}, E, R \cup \{s \to t, u' \to v'\})}{(\mathcal{I\!I}, E, R \cup \{s \to t, u \to v\})}$$

if $u' \leftrightarrow^*_{AC} su$, $v' \leftrightarrow^*_{AC} tv$, and $\boldsymbol{\alpha} < \boldsymbol{\beta}$, where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the measures associated with the rules $s \to t$ and $u' \to v'$ respectively. The rule $u \to v$ is assigned the measure $\boldsymbol{\beta} - \boldsymbol{\alpha}$.

Along with the Deletion3 rule, this forms a set of transformations that can be used to complete a given finite set of (reverse) D-, C-, and AC-rules over a signature $\Sigma = \Sigma_{AC}$ containing exactly one cancellative AC function symbol.

Example 2. Consider the set $R_0 = \{c_1^2 \to c_2, c_2^2 \to c_1\}$ of directed rules. We can complete this set as follows (we show only the third component of the state here as the other components remain unchanged):

i	Rules R_i	Meas	Inference
0	R_0		
1	$R_0 \cup \{c_1c_2 \to e\}$	$[1,1]$	$\mathbf{ACC-Ch}$
2	$R_1 \cup \{c_1^3 \to e\}$	$[2,1]$	$\mathbf{ACC-Ch}$

i	Rules R_i	Meas	Inference
3	$R_2 \cup \{c_2^3 \to e\}$	$[1,2]$	$\mathbf{ACC-Ch}$
4	$R_3 \cup \{c_1^2c_2^2 \to e\}$	$[2,2]$	$\mathbf{ACC-Ch}$
5	R_3		$\mathbf{ACC-Col}$

Any rule subsequently deduced by chaining can be simplified by collapse and no additional rules are added to the set R_3. Thus, the system R_3 is the desired completion.

We say $s \to^{[]}_R t$ if, and only if, there exists a term u such that $s \cdot u \to_R t \cdot u$. The reflexive-transitive closure of the relation $\to^{[]}_R$ and is denoted by $\to^{[*]}_R$.

Theorem 4 (Soundness). *Suppose $s \approx t \in R_i$, where $(\emptyset, \emptyset, R_i)$ is a state in any derivation starting from state $(\emptyset, \emptyset, R_0)$. Then, $s \to^{[*]}_{R_0 \cup AC} t$.*

Theorem 5 (Completeness). *Let R_0 be a finite set of (reverse) D-, C-, and AC-rules over $\Sigma \cup K$. The set R_∞ of persisting rules in any fair derivation starting from the state $(\emptyset, \emptyset, R_0)$ is finite. Furthermore, if $s \to^{[*]}_{R_0 \cup AC} t$, then there is a proof of the form $s \to^*_{AC \setminus F^e} \circ \leftrightarrow^*_{AC} \circ \leftarrow^*_{AC \setminus B^{-e}} t$.*

We combine the inference rules for the individual cancellative AC symbols and the inference rules for uninterpreted ground terms to get a procedure for constructing a rewrite closure for a set of equations and rules over a signature containing cancellative AC function symbols [16]. There are a few technical difficulties here however. First, in the case of a monoid, the length of the measure vector assigned to a rule was determined by the number of rules in the initial R-component, R_0. In the general case, these rules are created by orientation and moved from the $\mathit{\Pi}$-component to the R-component. Secondly, in the case of a monoid, all the C-rules in the R-component had exactly one measure vector associated with them. In case of a signature with $|\Sigma_{AC}|$ AC symbols, each C-rule will have a measure vector associated with it for each $f \in \Sigma_{AC}$. Third, we need an AC-compatible ordering that orients the D-rules in the right way. For this purpose, we use the ordering $\succ$ defined in [15]. When comparing two terms from a monoid, it reduces to the total degree lexicographic ordering. Finally, we additionally need ACC-superposition and ACC-paramodulation rules, for details and correctness see [16].

Other Remarks The equational theory induced by a set of ground equations over a signature containing (non-cancellative) AC-symbols can be conservatively represented by D-rules, C-rules, and A-rules [5]. But, if we are interested in the *rewrite* relation, then the problem becomes much harder, as classical petrinet reachability is equivalent to the decidability of the rewrite relation induced by a set of ground rules over an abelian semigroup. A derivation using the inference rules presented here does not converge in the case of abelian semigroups. For instance, consider the petrinet with two states c_1 and c_2 and two transitions $c_1^4c_2 \to c_1c_2^2$ and $c_1^4c_2 \to c_1^3c_2^2$. ACC-Chaining inferences (assuming a total degree lexicographic ordering with $c_1 \succ c_2$) yield infinitely many persisting rules $c_1^5c_2 \to c_1c_2^3$, $c_1^6c_2 \to c_1c_2^4, \ldots, c_1^nc_2 \to c_1c_2^{n-2}$. The reachability problem for petri nets was shown to be decidable in [14, 12].

The problem of deciding reachability in the case of a cancellative monoid is related to solving a system of linear diophantine equations by "duality". Consider the system $\{4x_1 - x_2 - 2x_3 = 0, 3x_1 - 4x_2 + 5x_3 = 0\}$. This system can be transformed into the three rewrite rules $c_1^4c_2^3 \to e$, $e \to c_1c_2^4$, and $c_2^5 \to c_1^2$. The original system has a non-trivial solution if and only if $e \to^+ e$. The converse translation can be similarly done. This connection is not surprising since one motivation for considering the cancellative axiom for AC-symbols comes from AC-unification, where linear diophantine equations arise naturally.

4 Conclusion

We have presented a set of inference rules, derived from standard completion and non-symmetric completion, to construct a rewrite closure for a set of ground equations and rules over a signature that can possibly contain cancellative AC symbols. The procedure works over an extended signature, incorporates essential simplifications, and is terminating.

There are several directions in which we envisage future work. The inference rules can be extended by including rules for unification and for special kinds of rewrite relations, like the various path orderings. This would give abstract transformation rules for constraint solving. Another possible extension is to (obtain decision procedures for) ordered fields. In this context, the non-symmetric relation will be interpreted as the ordering relation $>$ on the field elements. This work can also be extended along the lines of tree automata techniques and could be used to obtain *efficient* decision procedures for several properties of ground rewrite systems, for example confluence.

Acknowledgements. We would like to thank the anonymous reviewers for their 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, October 1989.
[3] L. Bachmair and N. Dershowitz. Equational inference, canonical proofs, and proof orderings. *Journal of the ACM*, 41:236–276, 1994.
[4] L. Bachmair and H. Ganzinger. *Ordered chaining calculi for first-order theories of binary relations*. Technical Report MPI-I-95-2-009, 1995.
[5] L. Bachmair, I.V.Ramakrishnan, A. Tiwari, and L. Vigneron. Congruence closure modulo associativity and commutativity. In *Frontiers of Combining Systems, 3rd Intl Workshop FroCoS 2000*, pages 245–259. Springer, 2000. LNAI 1794.
[6] L. Bachmair and A. Tiwari. Abstract congruence closure and specializations. In D. McAllester, editor, *CADE*, pages 64–78. Springer, 2000. LNAI 1831. Full version to appear in J. of Automated Reasoning, `www.csl.sri.com/users/tiwari/`.
[7] T. Becker and V. Weispfenning. *Gröbner bases: a computational approach to commutative algebra*. Springer-Verlag, Berlin, 1993.
[8] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available at: `http://www.grappa.univ-lille3.fr/tata`, 1997.
[9] M. Dauchet, T. Heuillard, P. Lescanne, and S. Tison. Decidability of the confluence of ground term rewriting systems. In *Proc IEEE Symposium on Logic in Computer Science, LICS*, pages 353–359. IEEE Computer Society Press, 1987.
[10] C. Kirchner, editor. *Rewriting Techniques and Applications, RTA-93*, Montreal, Canada, 1993. Springer-Verlag. LNCS 690.
[11] A. Levy and J. Agusti. Bi-rewriting, a term rewriting technique for monotone order relations. In Kirchner [10], pages 17–31. LNCS 690.
[12] E. W. Mayr. Persistence of vector replacement systems is decidable. *Acta Informatica 15*, pages 309–318, 1981. NewsletterInfo: 8.
[13] D. A. Plaisted. Polynomial time termination and constraint satisfaction tests. In Kirchner [10], pages 405–420. LNCS 690.
[14] S. Rao Kosaraju. Decidability of reachability in vector addition systems. In *Proc. 14th annual ACM symposium on theory of computing*, pages 267–281, May 1982.
[15] A. Rubio and R. Nieuwenhuis. A precedence-based total AC-compatible ordering. In Kirchner [10], pages 374–388. LNCS 690.
[16] A. Tiwari. Rewrite closure for ground and cancellative AC theories. Available: `www.csl.sri.com/users/tiwari/fsttcs01.html`, 2001. Full version of this paper.