An Algorithm for Finding Canonical Sets of Ground Rewrite Rules in Polynomial Time

JEAN GALLIER

University of Pennsylvania, Philadelphia, Pennsylvania

PALIATH NARENDRAN

State University of New York at Albany, Albany, New York

DAVID PLAISTED

University of North Carolina, Chapel Hill, North Carolina

STAN RAATZ

Rutgers University, New Brunswick, New Jersey

AND

WAYNE SNYDER

Boston Unwersity, Boston, Massachusetts

Abstract. In this paper, it is shown that there is an algorithm that, given any finite set E of ground equations, produces a reduced canonical rewriting system R equivalent to E in polynomial time. This algorithm based on congruence closure performs simplification steps guided by a total simplification ordering on ground terms, and it runs in time $O(n^3)$.

Categories and Subject Descriptors: D.3 [**Programming Languages**]; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*computations on discrete structures*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*computational logic, Mechanical theorem proving*; F.4.2 [**Mathematical Logic and Formal Language**]: Grammars and Other Rewriting Systems—*decision problems*; I.1.3 [**Algebraic Manipulation**]: Languages and Systems—*special-purpose algebraic systems*; I.2.3 [**Artificial Intelligence**]: Deduction and Theorem Proving—*deduction*

This research was partially supported by the National Science Foundation under Grant DCR 86-07156, and by the Office of Naval Research under Grant N00014-88-K-0593.

Authors' addresses: J. Gallier, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104; P. Narendran, Department of Computer Science, State University of New York at Albany, Albany, NY 12222; D. Plaisted, University of North Carolina, Department of Computer Science, Chapel Hill, NC 27514; S. Raatz, Rutgers University, Department of Computer Science, New Brunswick, NJ; W. Snyder, Department of Computer Science, Boston University, 111 Cummington Street, Boston, MA 02215.

General Terms: Algorithms

Additional Key Words and Phrases: Completion procedures, congruence closure, equational logic, term rewriting

1. *Introduction*

In this paper, it is shown that there is an algorithm *Reduce* that, given any finite set E of ground equations, produces a reduced canonical rewriting system[1] R equivalent to E in polynomial time. The algorithm *Reduce* plays a crucial role in the decision procedure showing that rigid unification (first introduced in Gallier et al. [12]) is NP-complete, a result announced (without complete proofs) at LICS'88 [13], and proved rigorously in Gallier et al. [15]. Rigid E-unification itself arises naturally in generalizing the method of matings due to Bibel and Andrews [1, 4–6] to languages with equality. This extension of the method of matings, called equational matings, is discussed extensively in Gallier, Narendran, Plaisted, Raatz, and Snyder [12, 14, 16].

We wish to stress that we do not view our algorithm as a substitute for congruence closure (Kozen [21, 22], Nelson and Oppen [25], Downey et al. [10]). The original motivation for the algorithm *Reduce* arose in the process of proving the decidability of rigid E-unification. Later on, we realized that this algorithm could perhaps be useful in other areas. For instance:

—The study of this type of approach might yield generalizations to certain classes of ground term rewriting systems + nonground term-rewriting systems that can be completed efficiently. For example, Jieh Hsiang (private communication) has suggested that an extension of this method to account for associativity and commutativity would be useful for Grobner Bases, an important class of algorithms in symbolic computation.

—Our method might be a useful part of a Knuth–Bendix completion procedure for nonground terms. It could help speeding up the completion process. Also, from a theoretical point of view, it seems interesting to know how fast reduced systems can be found.

—It has been suggested that the results presented here might be useful in an "abstract interpretation" or "program analysis" approach to rewrite rules, particularly those used in language implementation.

2. *Preliminaries*

We review briefly the concepts that will be needed in this paper. As much as possible, we tried to use notation and terminology consistent with Huet and Oppen [18] and Gallier [11]. In the interest of brevity, the reader is referred to Gallier et al. [15] for all unexplained notation. In this paper, only *finite* ranked alphabets will be considered. Given a term t and a tree address α in t, t/α denotes the subterm of t rooted at α. Given two terms s, $t \in T_\Sigma$ (the set of ground terms over Σ) and a tree address α in s, the term $s[\alpha \leftarrow t]$ is the result of replacing the subterm rooted at α in s by t.

A *simplification ordering* $\prec$ is a strict ordering that is monotonic and has the subterm property. It is shown in Dershowitz [8] that for finite ranked alphabets,

[1] A canonical system is a confluent and Noetherian system. The term, *reduced*, is defined in the next section.

every simplification ordering is well founded, and that there exist total simplification orderings on ground terms. Note that if a strict ordering $\prec$ is total, monotonic, and well founded, we must have $s \prec f(\ldots,s,\ldots)$ for every s, since otherwise, by monotonicity, we would have an infinite decreasing chain.

Let $E \subseteq T_\Sigma \times T_\Sigma$ be a set of ground rewrite rules. We denote the rewrite relation associated with E as $\rightarrow_E$, and we let $\overset{*}{\leftrightarrow}_E$ be the reflexive, symmetric, and transitive, closure of $\rightarrow_E$. It is well known that $\overset{*}{\leftrightarrow}_E$ is the smallest congruence on T_Σ containing E. When we want to fully specify a rewrite step, we use the notation $t_1 \rightarrow_{[\alpha,s\rightarrow t]} t_2$.

Definition 2.1. Given a set R of ground rewrite rules and a total simplification ordering $\prec$, we say that R is *compatible with* $\prec$ iff $r \prec l$ for every $l \rightarrow r \in R$.

Given a set R of ground rewrite rules, we say that R *is reduced* iff

(1) No lefthand side of any rewrite rule $l \rightarrow r \in R$ is reducible by any rewrite rule in $R - \{l \rightarrow r\}$;
(2) No righthand side of any rewrite rule $l \rightarrow r \in R$ is reducible by any rewrite rule in R.

It is well know (Huet [17]) that a Noetherian relation is confluent iff it is locally confluent. We say that R is *canonical* iff it is Noetherian and confluent. Note that since a reduced set of ground rewrite rules has no critical pairs, by [17], it is locally confluent. A reduced set of ground rewrite rules compatible with $\succ$ is also Noetherian because $r \prec l$ for every rule $l \rightarrow r$, and $\prec$ is a simplification ordering. Hence, by [17], such a set is confluent.

3. *The Procedure* Reduce

It has been known for some time that because total reduction orderings on ground terms exist, Knuth–Bendix type completion procedures do not fail on input sets consisting of *ground equations* and terminate with a canonical system equivalent to the input set. This has been noted by Dershowitz [9] who attributes the result to Lankford [24]. The precise reason is that newly formed equations can always be oriented (because a reduction ordering total on ground terms can be used). Actually, if one examines carefully the inference rules describing the Knuth–Bendix completion procedure (Knuth and Bendix [20]) in the formalism of Bachmair [2], Bachmair et al. [3], one will notice that because the rules are ground, the inference rule yielding critical pairs never applies, but instead the simplification rules apply. From this and the fact that newly formed equations can always be oriented, it is easy to see that the completion procedure always halts with success. However, the complexity of such a procedure is unclear. In this section, we give such an algorithm based on congruence closure (Kozen [21, 22], Nelson and Oppen [25], Downey et al. [10]) that runs in time $O(n^3)$. The correctness of this algorithm is nontrivial, and we give a rigorous proof.

We shall define a sequence of triples $\langle \mathscr{E}_i, \Pi_i, \mathscr{R}_i \rangle$ where $\mathscr{E}_i$ is a finite set of ground equations, Π_i is a partition (associated with $\mathscr{E}_i$), and $\mathscr{R}_i$ is a set of ground rewrite rules. Given a triple $\langle \mathscr{E}_i, \Pi_i, \mathscr{R}_i \rangle$, we let $\mathscr{T}_i$ be the set of all subterms of terms occurring in equations in $\mathscr{E}_i$ or in rewrite rules in $\mathscr{R}_i$. The algorithm below makes use of the *congruence closure* of a finite set of ground equations (Kozen [21, 22], Nelson and Oppen [25], Downey et al. [10]).

Congruence closures are represented by their associated partition Π. Given an equivalence relation represented by its partition Π, the equivalence class of t is denoted by $[t]_\Pi$, or $[t]$. Recall that s, t are in the same equivalence class of Π iff s and t are subterms of the terms occurring in E and $s \overset{*}{\leftrightarrow}_E t$ (for details, see Gallier [11]). The congruence closure algorithm will only be run once on E to obtain Π_0, but the partition Π_ι may change due to further steps (simplification steps). Note that for the purpose of defining the algorithm, it is sufficient to deal with pairs $\langle \Pi_\iota, \mathscr{R}_\iota \rangle$, but the component $\mathscr{E}_\iota$ is necessary for the proof of correctness, and this is why the method is presented in terms of triples. The following conventions are used in the algorithm *Reduce* shown below:

—A *nontrivial class* is a class containing at least two elements, in which case $\mathscr{E}_\iota$ has at least one nontrivial equation.
—$|C|$ denotes the cardinality of the set C;
—For simplicity of notation, we occasionally omit the subscript $i + 1$.
—By a maximal redex of λ, we mean a redex of λ that is not a proper subterm of any other redex of λ. The simplified term is irreducible with respect to $S_{\iota+1}$, so these replacements are only done once, and they can be done in parallel because they apply to independent subterms of λ.
—The ordering on ground terms is $\prec$.

begin algorithm *Reduce*
Initially, we set $\mathscr{E}_0 = E$, $\mathscr{R}_0 = \emptyset$, and run a congruence closure algorithm on the ground set E to obtain Π_0. $\iota := 0$;
while Π_ι, has some nontrivial equivalence class **do** {Simplification steps}
Let $\rho_{\iota+1}$ be the smallest element of the set

$$\bigcup_{C \in \Pi_\iota, |C| \geq 2} C$$

of terms belonging to nontrivial classes in Π_ι. Let $C_{\iota+1}$ be the nontrivial class that contains $\rho_{\iota+1}$, and write $C_{\iota+1} = \{\rho_{\iota+1}, \lambda_{\iota+1}^1, \ldots, \lambda_{\iota+1}^{k_{\iota+1}}\}$, where $k_{\iota+1} \geq 1$, since $C_{\iota+1}$ is nontrivial. Let $\mathscr{S}_{\iota+1}^{\cdot} = \{\lambda_{\iota+1}^1 \to \rho_{\iota+1}, \ldots, \lambda_{\iota+1}^{k_{\iota+1}} \to \rho_{\iota+1}\}$.
{Next, we use the rewrite rules in $\mathscr{S}_{\iota+1}^{\cdot}$ to simplify the rewrite rules in $\mathscr{R}_\iota \cup \mathscr{S}_{\iota+1}^{\cdot}$, the partition Π_ι, and the equations in $\mathscr{E}_\iota$.}[2]
To get $\mathscr{R}_{\iota+1}$, first, we get a canonical system equivalent to $\mathscr{S}_{\iota+1}^{\cdot}$. For this, for every lefthand side λ of a rule in $\mathscr{S}_{\iota+1}^{\cdot}$, replace every maximal redex of λ of the form λ^j by ρ, where $\lambda^j \to \rho \in \mathscr{S}_{\iota+1}^{\cdot} - \{\lambda \to \rho\}$. Let $\mathscr{S}_{\iota+1}{}'$ be the set of simplified rules. Also, let $\mathscr{R}'_{\iota+1}$ be the set obtained by simplifying the lefthand sides of rules in $\mathscr{R}_\iota$ using $\mathscr{S}_{\iota+1}^{\cdot}$ (reducing maximal redexes only), and let

$$\mathscr{R}_{\iota+1} = \mathscr{R}'_{\iota+1} \cup \mathscr{S}_{\iota+1}{}'.$$

Finally, use $\mathscr{S}_{\iota+1}^{\cdot}$ to simplify all terms in Π_ι and $\mathscr{E}_\iota$, using the simplification process described earlier to obtain $\Pi_{\iota+1}$ and $\mathscr{E}_{\iota+1}$.
$i := i + 1$
endwhile
{All classes of Π_ι are trivial, and the set $\mathscr{R}_\iota$ is a canonical system equivalent to E.}
end algorithm

At the end of the algorithm, $\mathscr{E}_\iota$ must consist entirely of trivial equations, that is, equations of the form $s \doteq s$. We have to justify the fact that when the lefthand side of a rule $l \to r \in \mathscr{R}_\iota$ is simplified to a rule $l' \to r$, it is still true that $r \prec l'$ holds, and that righthand sides are never simplified. Note that during the step where $\mathscr{S}_{\iota+1}^{\cdot}$ is used to simplify all terms in Π_ι and $\mathscr{E}_\iota$, the class $C_{\iota+1}$ is simplified to the trivial class $\{\rho_{\iota+1}\}$.

[2] This is one of the crucial steps that ensures a polynomial-time algorithm.

We claim that any sequence defined by the above procedure terminates, and that the set $\mathscr{R}_t$ obtained in the last step is a reduced canonical system equivalent to the original set E of equations. For this, we need a number of lemmas, but first, the method is illustrated in the following example.

Example 3.1. Let $\mathscr{E}_0 = E$ be the following set of ground equations:

$$\begin{aligned} E = \{&f^3a \doteq a,\\ &f^5a \doteq a,\\ &a \doteq d,\\ &gha \doteq a,\\ &gma \doteq a,\\ &ha \doteq c,\\ &mgc \doteq b\}. \end{aligned}$$

Let $\prec$ be a total simplification ordering such that, $d \prec c \prec b \prec a \prec f \prec g \prec h \prec m$. After computing the congruence closure for E, we have the initial partition

$$\begin{aligned} \Pi_0 = \{&\{d, a, fa, f^2a, f^3a, f^4a, f^5a, gc, gha, gma\},\\ &\{c, ha\},\\ &\{b, mgc, ma\}\}. \end{aligned}$$

The class $\{d, a, fa, f^2a, f^3a, f^4a, f^5a, gc, gha, gma\}$ is selected, since d is the least term. We have

$$\begin{aligned} \mathscr{S}_1 = \{&a \to d,\\ &fa \to d,\\ &f^2a \to d,\\ &f^3a \to d,\\ &f^4a \to d,\\ &f^5a \to d,\\ &gc \to d,\\ &gha \to d,\\ &gma \to d\}. \end{aligned}$$

After simplification, we obtain the reduced system

$$\begin{aligned} \mathscr{R}_1 = \{&a \to d,\\ &fd \to d,\\ &gc \to d,\\ &ghd \to d,\\ &gmd \to d\}. \end{aligned}$$

The partition Π_0 simplifies to

$$\begin{aligned}\Pi_1 = \{&\{d\},\\ &\{c, hd\},\\ &\{b, md\}\},\end{aligned}$$

and $\mathscr{E}_0$ to

$$\begin{aligned}\mathscr{E}_1 = \{&d \doteq d,\\ &hd \doteq c,\\ &md \doteq b\}.\end{aligned}$$

The next class selected is $\{c, hd\}$. We have

$$\mathscr{S}_2 = \{hd \to c\}.$$

After simplification, we have

$$\begin{aligned}\mathscr{R}_2 = \{&a \to d,\\ &fd \to d,\\ &gc \to d,\\ &gmd \to d,\\ &hd \to c\},\\ \Pi_2 = \{&\{d\},\\ &\{c\},\\ &\{b, md\}\},\end{aligned}$$

and

$$\begin{aligned}\mathscr{E}_2 = \{&d \doteq d,\\ &c \doteq c,\\ &md \doteq b\}.\end{aligned}$$

Finally, the class $\{b, md\}$ is selected, we have

$$\mathscr{S}_3 = \{md \to b\},$$

and after simplification, we have

$$\begin{aligned}\mathscr{R}_3 = \{&a \to d,\\ &fd \to d,\\ &gc \to d,\\ &gb \to d,\\ &hd \to c,\\ &md \to b\},\\ \Pi_3 = \{&\{d\},\\ &\{c\},\\ &\{b\}\},\end{aligned}$$

and

$$\mathscr{E}_3 = \{d \doteq d,$$
$$c \doteq c,$$
$$b \doteq b\}.$$

The reduced canonical system equivalent to E is $\mathscr{R}_3$.

4. *Correctness and Termination of the Procedure* Reduce

We now prove that the procedure *Reduce* (described in Section 3) terminates and produces a reduced canonical system equivalent to the original ground set E.

Definition 4.1. A set S of ground rewrite rules is *right uniform* iff $r = r'$ for all $l \to r$ and $l' \to r' \in S$.

Definition 4.2. Let S be a right uniform set of rules. The relation $\succ\!\!\rightarrow_S$ is defined such that $\succ\!\!\rightarrow_S$ is like rewriting using S, except restricted to maximal redexes. Formally, given any two ground terms s, t,

$$s \succ\!\!\rightarrow_S t \quad \text{iff} \quad t = s[\beta \leftarrow r],$$

where s/β is a maximal proper redex of s such that $s/\beta \to r \in S$ (a maximal redex is a redex that is not a proper subterm of any other redex). Note, the definition implies that $\beta \neq \epsilon$ (where ϵ denotes the empty string). Let $\succ\!\!\rightarrow_S^+$ be the transitive closure of $\succ\!\!\rightarrow_S$, and $\succ\!\!\rightarrow_S^*$ its reflexive and transitive closure.

LEMMA 4.3. *Let S be a set of ground rewrite rules compatible with $\prec$ such that*

(1) *S is right uniform;*
(2) *Whenever $\lambda \to r \in S$ and $\lambda' \overset{\cdot}{\leftrightarrow}_S \lambda$ where λ' is a subterm of the lefthand side of some rule in S ($\lambda' = l/\beta$ for some rule $l \to r \in S$), then $\lambda' \to r \in S$.*

Let $s \to r$ be any rule in S and let $S_1 = S - \{s \to r\}$. Then, the following statements hold:

(i) In any simplification sequence $s \succ\!\!\rightarrow_{S_1}^+ s'$, there cannot be two steps applied at addresses β_1 and β_2 in that order such that β_1 is an ancestor of β_2;
(ii) In any simplification sequence $s \succ\!\!\rightarrow_{S_1}^+ s'$, there cannot be two steps applied at addresses β_1 and β_2 in that order such that β_2 is an ancestor of β_1.

PROOF

(i) Since $l \succ r$ for every rule in S, r is irreducible with respect to S. By the definition of $\succ\!\!\rightarrow$, $\beta \neq \epsilon$ for every redex β in the sequence $s \succ\!\!\rightarrow_{S_1}^+ s'$ and in particular, β_1, $\beta_2 \neq \epsilon$. Immediately after the step performed at β_1, $s/\beta_1 = r$. Since r is irreducible with respect to S, no step can be applied at β_2 if β_1 is an ancestor of β_2.
(ii) If there is some simplification sequence $s \succ\!\!\rightarrow_{S_1}^+ s'$ with two steps applied at addresses β_1 and β_2 in that order such that β_2 is an ancestor of β_1, by (i), no step can be applied to any ancestor of β_2 (including β_2) until the step applied to β_2, and so $s/\beta_2 \succ\!\!\rightarrow_{S_1}^+ t$ where all steps are applied strictly below β_2, and there is some rule $l \to r \in S_1$ such that $l = t$, that is, t is reducible by S_1. Since $s/\beta_2 \overset{*}{\leftrightarrow}_{S_1}$, l, $S_1 \subset S$, and $s \to r \in S$, then by condition (2),

$s/\beta_2 \to r \in S$. Actually, $s/\beta_2 \to r \in S_1$ since $\beta_2 \neq \epsilon$. This contradicts the fact that maximal rules were applied in the sequence $s/\beta_2 \succ\!\!\rightarrow^+_{S_1} t$ (since all steps in this sequence are applied strictly below β_2). □

From Lemma 4.3, we obtain the following corollary.

COROLLARY 4.4. *If S is a set of rewrite rules compatible with a total simplification ordering $\prec$ and satisfying conditions* (1) *and* (2) *of Lemma* 4.3, *then for every ground term u, there is a unique ground term v irreducible with respect to $S - \{u \to r\}$ such that $u \succ\!\!\rightarrow^+_S v$, and v is obtained from u by replacing all maximal* (*independent*) *redexes of u by r, the common righthand side of all rules in S.*

PROOF. Since S is compatible with $\prec$, it is clear that $\succ\!\!\rightarrow_S$ is well founded. Hence, there is some irreducible term v with respect to $S - \{u \to r\}$ such that $u \succ\!\!\rightarrow^+_S v$. For any two redexes occurring at addresses β_1 and β_2 in this derivation, β_1 and β_2 must be independent since otherwise one of the two redexes would not be maximal. Then, by Lemma 4.3, the derivation $u \succ\!\!\rightarrow^+_S v$ consists in replacing all maximal independent proper redexes of u by r, and it is clear that this yields a unique irreducible term v. □

LEMMA 4.5. *Let S be a set of ground rewrite rules compatible with $\prec$ and satisfying conditions* (1) *and* (2) *of Lemma* 4.3. *Let*

$$S' = \{l' \to r \mid l \succ\!\!\rightarrow^*_S l', l' \succ r, l \to r \in S\},$$

where l' is the normal form of l with respect to $\succ\!\!\rightarrow_S$ guaranteed by Corollary 4.4. *Then S' is a reduced canonical system equivalent to S.*

PROOF. First, we show that S and S' are equivalent, that is, generate the same congruence. First, let $S'_0 = \langle l_1 \to r, \ldots, l_n \to r \rangle$ be S viewed a sequence ordered such that $l_n \prec \cdots \prec l_1$. Let

$$S'_i = \langle l'_1 \to r, \ldots, l'_i \to r, l_{i+1} \to r, \ldots, l_n \to r \rangle,$$

$1 \leq i \leq n$. We show by induction on i that S and S'_i are equivalent. This is obvious for $i = 0$ since S'_0 is just S as a sequence. Assume that S and S'_i are equivalent for $i < n$. Observe that due to the ordering of the l_j, the only rules involved in the simplification steps $l_j \succ\!\!\rightarrow^*_S l'_j$ are the rules in the set $\{l_{j+1} \to r, \ldots, l_n \to r\}$. But then, it is immediate that

$$S'_i = \langle l'_1 \to r, \ldots, l'_i \to r, l_{i+1} \to r, l_{i+2} \to r, \ldots, l_n \to r \rangle$$

and

$$S'_{i+1} = \langle l'_1 \to r \ldots, l'_i \to r, l'_{i+1} \to r, l_{i+2} \to r, \ldots, l_n \to r \rangle$$

are equivalent, and since S and S'_i are equivalent (by the induction hypothesis), S and S'_{i+1} are also equivalent. This concludes the induction and shows that S and S'_n are equivalent. It is obvious that if we eliminate duplicate rules and trivial rules from S'_n we obtain the set of rules S', and it is also obvious that S'_n and S' are equivalent. Thus, S and S' are equivalent.

Assume that $l'_1 \to r$ and $l'_2 \to r$ are rules in S', where $l_1 \to r$ and $l_2 \to r$ are rules in S, l'_1 and l'_2 are the normal forms of l_1 and l_2 with respect to S, and that $l'_2 = l'_1/\beta$, that is, S' is not reduced. Then, because $r \prec l$ for all $l \to r \in S$, it must be the case that no simplification steps are applied to l_1 at or above β

(since otherwise $r \succeq l_1'/\beta = l_2'$ and so $r \succeq l_2'$, contradicting $l_2' \to r$ is compatible with $\prec$), and so l_1'/β is the normal form of l_1/β in S and $l_1/\beta \overset{*}{\leftrightarrow}_S l_1'/\beta = l_2' \overset{*}{\leftrightarrow}_S l_2$. Since $l_2 \to r \in S$ by condition (2), we also have $l_1/\beta \to r \in S$. But this contradicts that simplification steps were applied at maximal subterms of l_1, since we showed previously that no simplification step can be performed at or above β. Hence, S' is reduced. Since it is also Noetherian, it is canonical. □

LEMMA 4.6. *For every* $i \geq 0$,

$$\rho_{i+1} \prec \rho_{i+2} = min\left(\bigcup_{C \in \Pi_{i+1}, |C| \geq 2} C \right).$$

PROOF. First, note the following fact:

Fact. If S is a right uniform set of ground rules and r is the common righthand side of all rules in S, for any ground term $s \succeq r$, if $s \overset{*}{\to}_S s'$, then $s' \succeq r$.

Next we prove the lemma by induction on i. For $i = 0$, ρ_1 is the least element of $\mathcal{T}_0$, (the set of all subterms occurring in equations in E) and the claim follows from the fact stated above and because $\rho_1 \notin \bigcup_{C \in \Pi_1, |C| \geq 2} C$. By the induction hypothesis,

$$\rho_{i+1} \prec \rho_{i+2} = \min\left(\bigcup_{C \in \Pi_{i+1}, |C| \geq 2} C \right).$$

Since Π_{i+2} is obtained from Π_{i+1} by simplification using rules in S_{i+2}, which are of the form $\lambda \to \rho_{i+2}$, and since $\rho_1 \prec \cdots \prec \rho_{i+1} \prec \rho_{i+2}$, by the fact stated above, it is clear that ρ_{i+2} is the smallest element of the set

$$\left(\bigcup_{C \in \Pi_{i+2}} C \right) - \{\rho_1, \ldots, \rho_{i+1}\},$$

which implies that

$$\rho_{i+2} \prec \rho_{i+3} = \min\left(\bigcup_{C \in \Pi_{i+2}, |C| \geq 2} C \right),$$

since $\rho_{i+2} \notin \bigcup_{C \in \Pi_{i+2}, |C| \geq 2} C$. This concludes the induction step. □

From Lemma 4.6, we obtain the following corollary:

COROLLARY 4.7. *For every* $l \to r \in \mathcal{R}_i$, r *is never simplified by any rule in* S_{i+1}, *and if* l *simplifies to* l', *then* $r \prec l'$.

PROOF. By Lemma 4.6, $\rho_{i+1} \prec \rho_{i+2}$ for all $i \geq 0$. Since the simplification rules in $\mathcal{S}_{i+1}$ are of the form $\lambda \to \rho_{i+1}$ and the set of righthand sides of rules in $\mathcal{R}_i$ is $\{\rho_1, \ldots, \rho_i\}$, the result is clear. □

LEMMA 4.8

(1) *The sequence* $\langle \mathcal{E}_i, \Pi_i, \mathcal{R}_i \rangle$ *is finite and its length* m *is bounded by the number of nontrivial equivalence classes in* Π_0.
(2) Π_i *is the partition associated with the congruence closure of* $\mathcal{E}_i$ *for every* i,

where $0 \leq i \leq m$.

PROOF

(1) This part follows from the fact that when Π_{i+1} is derived from Π_i, the equivalence class C_{i+1} (in Π_i) of ρ_{i+1} collapses to the trivial class $\{\rho_{i+1}\}$.
(2) This part is shown by induction on i. The details are straightforward and are left to the reader (we use the fact that both Π_i and $\mathscr{E}_i$ are simplified by the set S_{i+1}). □

LEMMA 4.9. *Let m be the length of the sequence* $\langle \mathscr{E}_i, \Pi_i, \mathscr{R}_i \rangle$. *Then,*

$$\overset{*}{\leftrightarrow}_E = \overset{*}{\leftrightarrow}_{\mathscr{E}_i \cup \mathscr{R}_i},$$

for all i, $0 \leq i \leq m$.

PROOF. The proof is by induction on i. By the definition of $\langle \mathscr{E}_0, \Pi_0, \mathscr{R}_0 \rangle$, it is obvious that $\overset{*}{\leftrightarrow}_E = \overset{*}{\leftrightarrow}_{\mathscr{E}_0 \cup \mathscr{R}_0}$. Secondly, note that every use of an equation $s \doteq t \in \mathscr{E}_i$ such that $s, t \in C_{i+1}$ can be simulated by using the rewrite rules $s \to \rho_{i+1}$ and $t \to \rho_{i+1}$ that are in S_{i+1}. Since by Lemma 4.8, Π_i is the partition associated with the congruence closure of $\mathscr{E}_i$, from the way it is constructed the set of simplification rules S_{i+1} satisfies the conditions of Lemma 4.5, and $\mathscr{S}_{i+1'}$ is a reduced canonical system equivalent to $\mathscr{S}_{i+1}$. Thus, every use of an equation $s \doteq t$ as above can be simulated by rewrite rules in $\mathscr{S}_{i+1'}$. Thirdly, it is easy to show that the use of a rewrite rule $l \to r$ in $\mathscr{R}_i$ can be simulated by the simplified rule $l' \to r$ in $\mathscr{R}'_{i+1}$ and rules in $\mathscr{S}_{i+1'}$. Similarly, it is easy to show that the use of an equation $l \doteq r$ in $\mathscr{E}_i$ such that $l, r \notin C_{i+1}$ can be simulated by the simplified equation $l' \doteq r'$ in $\mathscr{R}'_{i+1}$ and rules in S'_{i+1}. This shows that

$$\overset{*}{\leftrightarrow}_{\mathscr{E}_i \cup \mathscr{R}_i} \subseteq \overset{*}{\leftrightarrow}_{\mathscr{E}_{i+1} \cup \mathscr{R}_{i+1}},$$

that is,

$$\overset{*}{\leftrightarrow}_E \subseteq \overset{*}{\leftrightarrow}_{\mathscr{E}_{i+1} \cup \mathscr{R}_{i+1}},$$

since by the induction hypothesis we have

$$\overset{*}{\leftrightarrow}_E = \overset{*}{\leftrightarrow}_{\mathscr{E}_i \cup \mathscr{R}_i}.$$

Conversely, since $\mathscr{S}_{i+1}$ is formed from an equivalence class of Π_i and, by Lemma 4.8, Π_i is the partition associated with the congruence closure of $\mathscr{E}_i$, since by the induction hypothesis we have $\overset{*}{\leftrightarrow}_E = \overset{*}{\leftrightarrow}_{\mathscr{E}_i \cup \mathscr{R}_i}$, it is clear that $\overset{*}{\leftrightarrow}_{\mathscr{S}_{i+1}} \subseteq \overset{*}{\leftrightarrow}_E$. But then, from the way $\mathscr{E}_{i+1} \cup \mathscr{R}_{i+1}$ is obtained from $\mathscr{E}_i \cup \mathscr{R}_i$ using $\mathscr{S}_{i+1}$, it is easy to see that

$$\overset{*}{\leftrightarrow}_{\mathscr{E}_{i+1} \cup \mathscr{R}_{i+1}} \subseteq \overset{*}{\leftrightarrow}_E.$$

Hence,

$$\overset{*}{\leftrightarrow}_E = \overset{*}{\leftrightarrow}_{\mathscr{E}_{i+1} \cup \mathscr{R}_{i+1}},$$

establishing the induction step. □

We now prove the following crucial lemma.

LEMMA 4.10. *The system* $\mathscr{R}_m$ *is reduced.*

PROOF. For every i, $1 \leq i \leq m$, the set $\{C'_1, \ldots, C'_{n_i}\}$ is defined as follows: $\{C'_1, \ldots, C'_{n_i}\}$ consists of all classes in the set $\{C_1, \ldots, C_i\}$ of nontrivial equiva-

lence classes selected by the algorithm, and all singletons of the form $\{u\}$, where u is some subterm of a term in one of the classes $C_1, \ldots, C_i$ and $u \notin C_k$ for every k, $1 \leq k \leq i$. Given any set C'_j, its representative ρ'_j is defined such that $\rho'_j = \rho_k$, the representative chosen by the algorithm if $C'_j = C_k$ is a nontrivial class, else $\rho'_j = u$, the single element in the set $C'_j = \{u\}$. We also order the set $\{C'_1, \ldots, C'_{n_i}\}$ to form the sequence $\langle C'_1, \ldots, C'_{n_i} \rangle$ as follows: C'_k precedes C'_l iff $\rho'_k \prec \rho'_l$.

We say that a term u in Π_i is *simplified* (*at stage i*) if, either u is not in any of the classes $C'_1, \ldots, C'_{n_i}$, or $u = \rho'_k$ for some k, $1 \leq k \leq n_i$. For $i = 0$, we define $\{C'_1, \ldots, C'_{n_i}\}$ as the empty set and $\langle C'_1, \ldots, C'_{n_i} \rangle$ as the empty sequence. We shall prove the following claim by induction on i.

CLAIM. *For every* i, $0 \leq i \leq m - 1$, *the following properties hold*:

(*a*) *If* $l \to r$ *is a rule in* $\mathscr{R}_{i+1}$, *then all proper subterms of* l *and all subterms of* r *are simplified, and every proper subterm of any term* $u \in \Pi_{i+1} - \mathscr{R}_{i+1}$ *is simplified.*

(*b*) *If* u *is a representative of some* C'_l, $1 \leq l \leq n_i$, *then every proper subterm of* u *is also the representative of some* C'_k, $1 \leq k < l$.

PROOF OF CLAIM. The claim is true for $i = 0$ since $\{C'_1, \ldots, C'_{n_i}\}$ is the empty set. The induction step is established as follows: Observe that the new class $C_{i+1} = C'_{n_{i+1}}$ chosen by the algorithm has the property that all proper subterms of the representative $\rho_{i+1} = \rho'_{n_{i+1}}$ are previously chosen representatives. This is because since $\prec$ has the subterm property, $u \prec \rho'_{n_{i+1}}$ for every proper subterm u of $\rho'_{n_{i+1}}$. Then, because $\rho'_{n_{i+1}}$ is chosen minimal, we must have $u \in C'_k$ for some $k < n_{i+1}$ and the induction hypothesis applies. Thus, property (b) holds. To prove (a), we simply note three facts:

(1) Righthand sides of rules in $\mathscr{S}_{i+1}$ or $\mathscr{R}_{i+1}$ are representatives of $C_1, \ldots, C_{i+1}$, since by Lemma 4.7, only lefthand sides of rules in $\mathscr{R}_i$ are simplified.

(2) When the lefthand side l of a rule in S_{i+1} or $\mathscr{R}_i$ is simplified, either the rule disappears, or some proper subterm u or l is replaced by ρ_{i+1}. But then, every proper subterm u of l is either a subterm of $\rho_{i+1} = \rho'_{n_{i+1}}$, or by the induction hypothesis a subterm of ρ'_k for some k, $1 \leq k \leq n_i$, or u is not in any of the classes $C'_1, \ldots, C'_{n_i}$. This shows that u is either the representative of one of the classes C'_k, $1 \leq k \leq n_{i+1}$, or that $u \notin C'_k$ for every k, $1 \leq k \leq n_{i+1}$. Hence, u is simplified at stage $i + 1$.

(3) The same property applies to proper subterms of terms in $\Pi_{i+1} - \mathscr{R}_{i+1}$.

This proves (a) and concludes the proof of the claim. □

We now apply the claim to the rules in $\mathscr{R}_m$. Therefore, every subterm u of a term in a rule from $\mathscr{R}_m$ is the representative of some equivalence class in $C'_1, \ldots, C'_{n_m}$ or $u \notin C'_k$ for every k, $1 \leq k \leq n_m$, except possibly for lefthand sides of rules. Thus, every subterm u of a term in a rule from $\mathscr{R}_m$ is the representative of some equivalence class in $C'_1, \ldots, C'_{n_m}$ or belongs to some trivial class of Π_m, except possibly for lefthand sides of rules. This means that no rewrite rule in $\mathscr{R}_m$ can be used to further simplify $\mathscr{R}_m$, except possibly to simplify a lefthand side at the top level. Assume that some rule $l_2 \to r_2$ in $\mathscr{R}_m$ simplifies the lefthand side of some rule $l_1 \to r_1$ in $\mathscr{R}_m$. Then, l_2 and l_1 must be identical, so l_1, l_2, r_1, and r_2 are all in the same equivalence class. Since r_1

and r_2 are both the representative of this class, we have $r_1 = r_2$. However, by definition, a rule does not reduce itself. Thus, $\mathscr{R}_m$ is reduced. □

We finally have our main result.

THEOREM 4.11. *Given a finite set E of ground equations, the procedure* Reduce *terminates with a reduced canonical system $\mathscr{R}_m$ equivalent to E.*

PROOF. The termination of the procedure *Reduce* is shown in Lemma 4.8. By Lemma 4.9, $\overset{*}{\leftrightarrow}_E = \overset{*}{\leftrightarrow}_{\mathscr{E}_m \cup \mathscr{R}_m}$, where m is the index of the final triple $\langle \mathscr{E}_m, \Pi_m, \mathscr{R}_m \rangle$. However, for this last triple, $\mathscr{E}_m$ consists of trivial equations, and so, $\overset{*}{\leftrightarrow}_E = \overset{*}{\leftrightarrow}_{\mathscr{R}_m}$. Finally, by Lemma 4.10, $\mathscr{R}_m$ is reduced. Because $\mathscr{R}_m$ is reduced and ground, there are no critical pairs, and since $\mathscr{R}_m$ is also Noetherian, it is confluent. □

5. *Complexity of the Procedure* Reduce

In this section, we analyze the complexity of the procedure *Reduce*.

LEMMA 5.1. *The algorithm* Reduce (*presented in Section* 3) *runs in time $O(n^3)$, where n measures the size of E.*

PROOF. First, observe that the number of subterms of terms occurring in E is $O(n)$, where n measures the size of E (say the length of the string obtained by concatenating the equations in E written in prefix notation). The number m of nontrivial equivalence classes of Π_0 is bounded by $\lfloor n/2 \rfloor$. Every term t is simplified using rules in $\mathscr{S}_{t+1}$ by replacing maximal (independent) subterms of t by ρ_{t+1}. If a DAG structure with sharing of common subterms is used, for each round, the simplification of all terms in Π_t and $\mathscr{R}_t$ by rules in $\mathscr{S}_{t+1}$ can be performed in $O(n)$. Hence, the complexity of the simplifications for m rounds is $O(n^2)$. The contribution of the congruence closure is $O(n^2)$. Finally, we need to make sure that there are total simplification orderings such that the least element of a set of k ground terms of total size n can be determined in time $O(n^2)$. However, this is not difficult to achieve. For example, one can use a recursive path ordering where sequences of subtrees are compared using a lexicographic ordering (see Dershowitz [8]); then we can use an $O(n^2)$ dynamic programming algorithm such as found in Krishnamoorthy and Narendran [23] or Snyder [27]. Hence, the complexity of the comparisons for m rounds is $O(n^3)$. Therefore, the complexity of the algorithm is $O(n^3)$. □

Note that the dominant factor in the time complexity of the procedure is the process of finding least elements with respect to a simplification ordering. If this can be reduced, the time complexity of the algorithm will also be reduced. The following example shows that a naive approach to simplification can lead to an exponential-time complexity.

Example 5.2. Given any integer $k > 1$, consider the following set E of equations:

$$
\begin{aligned}
gf^k c &\doteq fg^k c \\
\cdots &\doteq \cdots \\
gfffffc &\doteq fggggc \\
gffffc &\doteq fgggc \\
gfffc &\doteq fggc
\end{aligned}
$$

$$gfc \doteq fgc$$
$$gc \doteq fc.$$

The set of nontrivial classes of the partition Π_0 obtained after computing the congruence closure of E is

$$\begin{Bmatrix} \{fc, gc\}, \\ \{ffc, fgc, gfc, ggc\}, \\ \{fffc, fggc, gffc, gggc\}, \\ \vdots \\ \{f^k c, fg^{k-1}c, gf^{k-1}c, g^k c\}, \\ \{fg^k c, gf^k c\} \end{Bmatrix}.$$

It is clear that there are total simplification orderings induced by the total order on the function symbols such that $c \prec f \prec g$. The order in which the classes are selected by our algorithm amounts to simplifying E bottom-up. It is easy to see that we obtain the reduced system $\mathscr{R}_{k+1}$

$$\begin{aligned} gf^k c &\to f^{k+1}c \\ \cdots &\to \cdots \\ gfffc &\to ffffc \\ gffc &\to fffc \\ gfc &\to ffc \\ gc &\to fc \end{aligned}$$

in time $O(k^2)$. On the other hand, if we do not compute the congruence closure of E but simply transform E into the following set R of rewrite rules

$$\begin{aligned} gf^k c &\to fg^k c \\ \cdots &\to \cdots \\ gfffc &\to fgggc \\ gffc &\to fggc \\ gfc &\to fgc \\ gc &\to fc, \end{aligned}$$

and simplify R from the top-down, this takes exponential time in k. Indeed, in order to simplify $gf^k c \to fg^k c$ to $gf^k c \to f^{k+1}c$, $2^k - 1$ steps are required. This is shown by proving by induction that $g^k c$ simplifies to $f^k c$ in $2^k - 1$ steps. For $k = 1$, this is obvious using the last rule $gc \to fc$. Assuming inductively that $g^{k-1}c$ simplifies to $f^{k-1}c$ in $2^{k-1} - 1$ steps, then

$$g^k c = gg^{k-1}c \Rightarrow^* gf^{k-1}c$$

in $2^{k-1} - 1$ steps, $gf^{k-1}c \Rightarrow fg^{k-1}c$ using the second rule, and $fg^{k-1}c \Rightarrow^* ff^{k-1}c$ in $2^{k-1} - 1$ steps again. The total number of steps is $2^{k-1} - 1 + 1 +$

$2^{k-1} - 1 = 2^k - 1$, as claimed. Hence, it will take

$$2^k - 1 + 2^{k-1} - 1 + \cdots + 2^2 - 1 + 2^1 - 1 = 2^{k+1} - (k + 2)$$

steps to reduce R top-down.

6. *Relation to Other Work*

In this section, we clarify the relationship between our work and the work of Dauchet et al. [7], Otto and Squier [26], and Kapur and Narendran [19], and clear up some possibly confusing points. Dauchet et al. [7] prove that it is decidable whether a set of ground rewrite rules is confluent. The algorithm is fairly involved and its complexity is not clear, but it is unlikely that it runs in polynomial time. This is not in contradiction with our result. In fact, this decidability result has no bearing on our problem. Indeed, since our goal is to find a canonical system equivalent to the input system R, the orientation of the rules in R is irrelevant, and we are free to reorient the rules so that we have a Noetherian system. Having oriented the rules in R properly, we force confluence by interreducing the rules using our algorithm. Hence, we do not care whether the original set is confluent or not. Of course, Dauchet et al. [7] must accept the original orientation of the rules in R and they cannot change it. It is somewhat amusing to think that it might be faster to apply our algorithm to get a reduced canonical system than to test whether the given rules are confluent! Whether our work can be helpful for giving an alternate confluence test is another story, but we have not explored this path.

Both Otto and Squier [26] and Kapur and Narendran [19] show that there exist finite The systems with a decidable word problem for which no equivalent finite canonical system exists. Otto and Squier actually prove this result for finitely presented monoids with a decidable word problem. At first glance, this may seem to contradict our result. Indeed, strings are ground terms after all! However, we are forgetting that the free monoid over an alphabet Σ satisfies the *associativity* axiom

$$\forall x \forall y \forall z [x \cdot (y \cdot z) = (x \cdot y) \cdot z],$$

which is *not* equivalent to any finite set of ground equations. In fact, the associativity axiom is equivalent to *infinitely many* ground equations, all ground instances of the form $u \cdot (v \cdot w) = (u \cdot v) \cdot w$ obtained by substituting arbitrary strings $u, v, w \in \Sigma^*$ for the variables x, y, z. This explains the apparent contradiction. Our algorithm deals with a *finite* set of ground equations on the *initial* Σ-algebra T_Σ, where Σ is a finite ranked alphabet. The free monoid Σ^* is isomorphic to the *quotient* $T_\Delta / \equiv$ of the initial algebra T_Δ on the ranked alphabet $\Delta = \Sigma \cup \{\cdot, \epsilon\}$ (where $\cdot$ is a binary symbol, ϵ a constant, and every letter in Σ is a constant) by the least stable congruence $\equiv$ containing the set of (nonground) equations

$$\{x \cdot (y \cdot z) = (x \cdot y) \cdot z,\ x \cdot \epsilon = x,\ \epsilon \cdot x = x\}.$$

This is not the free Δ-algebra.

In principle, our algorithm can deal with a finite set E of nonground equations provided that there is a known bound k on the number of instances of equations used, but then the running time of our algorithm is $O(k^3)$, where k has nothing to do with the number of equations in the input set E.

7. *Conclusion*

An algorithm that produces a (reduced) canonical system equivalent to a set of ground equations has been presented and proved correct. This algorithm calls the congruence closure algorithm only once and performs simplification steps carefully. The present version of the algorithm runs in time $O(n^3)$. It is possible that using more sophisticated data structures the running time of the algorithm can be improved, but in this paper we are more concerned with correctness, and the issue of efficiency is left for further research. It is worth noting that this algorithm is at the heart of the decision procedure showing that rigid unification (first introduced in Gallier et al. [12]) is NP-complete, a result proved in Gallier et al. [15]. The algorithm of this paper seems attractive in applications where it is useful to compile a set of ground equations into a canonical set of rules efficiently, but this remains to be explored.

REFERENCES

1. ANDREWS, P. Theorem proving via general matings. *J. ACM 28*, 2 (1981), 193–214.
2. BACHMAIR, L. Proof methods for equational theories. Ph.D dissertation, Univ. Illinois, Urbana-Champaign, Ill., 1987.
3. BACHMAIR, L., DERSHOWITZ, N., AND PLAISTED, D. Completion without failure. In *Resolution of Equations in Algebraic Structures*, vol. 2. H. Aït-Kaci and M. Nivat, Eds. Academic Press, Orlando, Fla., 1989, pp. 1–30.
4. BIBEL, W. Tautology testing with a generalized matrix reduction method. *Theoret. Comput. Sci. 8* (1979), 31–44.
5. BIBEL, W. On matrices with connections. *J. ACM 28*, 4 (Oct. 1981), 633–645.
6. BIBEL, W. *Automated Theorem Proving*. Friedr. Vieweg & Sohn, Braunschweig, Germany, 1982.
7. DAUCHET, M., TISON, S., HEUILLARD, T., AND LESCANNE, P. Decidability of the confluence of ground term rewriting systems. In *Proceedings of the LICS'87* (Ithaca, N.Y.). IEEE, New York, 1987, pp. 353–359.
8. DERSHOWITZ, N. Termination of Rewriting, *J. Symb. Comput. 3* (1987), 69–116.
9. DERSHOWITZ, N. Completion and its applications, In *Resolution of Equations in Algebraic Structures*, vol. 2. H. Aït-Kaci and M. Nivat, Eds. Academic Press, Orlando, Fla., 1989, pp. 31–85.
10. DOWNEY, P. J., SETHI, R. AND TARJAN, R. E. Variations on the common subexpression problem. *J. ACM 27*, 4 (Oct. 1980), 758–771.
11. GALLIER, J. H. *Logic for Computer Science: Foundations of Automatic Theorem Proving*. Harper and Row, New York, 1986.
12. GALLIER, J. H., RAATZ, S., AND SNYDER, W. Theorem proving using rigid E-unification: Equational matings. In *Proceedings of the LICS'87* (Ithaca, N.Y.). IEEE, New York, 1987, pp. 338–346.
13. GALLIER, J. H., NARENDRAN, P., PLAISTED, D., AND SNYDER, W. Rigid E-unification is NP-complete. In *Proceedings of the LICS'88* (Edinburgh, Scotland, July 5–8). IEEE, New York, 1988, pp. 218–227.
14. GALLIER, J. H., RAATZ, S. AND SNYDER, W. Rigid E-unification and its applications to equational matings. In *Resolution of Equations in Algebraic Structures*, vol. 1. H. Aït-Kaci and M. Nivat, Eds. Academic Press, Orlando, Fla., 1989, pp. 151–216.
15. GALLIER, J. H., NARENDRAN, P., PLAISTED, D., AND SNYDER, W. Rigid E-unification: NP-completeness and applications to theorem proving. *Inf. Comput. 87*, 1/2 (special issue) (1990), pp. 129–195.
16. GALLIER, J. H., NARENDRAN, P., RAATZ, S., AND SNYDER, W. Theorem proving using equational matings and rigid E-unification. *J. ACM 39*, 2 (Apr. 1992), 377–429.
17. HUET, G. Confluent reductions: Abstract properties and applications to term rewriting systems. *J. ACM 27*, 4 (Oct. 1980), 797–821.
18. HUET, G., AND OPPEN, D. C. Equations and rewrite rules: A survey. In *Formal Languages: Perspectives and Open Problems*, R. V. Book, Ed. Academic Press, Orlando, Fla., 1982.

19. KAPUR, D., AND NARENDRAN, P. A finite thue system with decidable word problem and without equivalent finite canonical system. *Theoret. Comput. Sci. 35* (1985), 337–344.
20. KNUTH, D. E., AND BENDIX, P. B. Simple word problems in univeral algebras. In *Computational Problems in Abstract Algebra*, J. Leech, Ed. Pergamon Press, New York, 1970.
21. KOZEN, D. Complexity of finitely presented algebras. Tech. Rep. TR 76–294. Dept. Comput. Sci. Cornell Univ., Ithaca, N.Y., 1976.
22. KOZEN, D. Complexity of finitely presented algebras. In *Proceedings of the 9th Annual Symposium on Theory of Computing*, (Boulder, Colo., May). ACM, New York, 1977, pp. 164–177.
23. KRISHNAMOORTHY, M. S., AND NARENDRAN, P. On recursive path ordering. *Theoret. Comput. Sci. 40* (1985), 323–328.
24. LANKFORD, D. S. Canonical inference. Rep. ATP-32. Univ. Texas, Houston, Tex. 1975.
25. NELSON, G. AND OPPEN, D. C. Fast decision procedures based on congruence closure. *J. ACM 27*, 2 (Apr. 1980), 356–364.
26. OTTO, F., AND SQUIER, C. The word problem for finitely presented monoids and finite canonical rewriting systems. In *Proceedings of the RTA'87* (Bordeaux, France) (1987), pp. 74–82.
27. SNYDER, W. A note on the complexity of simplification ordering. Tech. Rep. BU-8909. Boston Univ. Boston, Mass., 1989.

RECEIVED MARCH 1985; REVISED JANUARY 1988; ACCEPTED JANUARY 1992