Acta Cybernetica **22** (2015) 517–551.

On Ground Word Problem of Term Equation Systems

Sándor Vágvölgyi*

To the memory of my teacher and colleague Ferenc Gécseg

Abstract

We give semi-decision procedures for the ground word problem of variable preserving term equation systems and term equation systems. They are natural improvements of two well known trivial semi-decision procedures. We show the correctness of our procedures.

Keywords: term equation systems; ground word problem; Knuth-Bendix completion procedure; ground term rewriting systems

1 Introduction

A term equation $l \approx r$ is called variable preserving if the same variables occur in the left-hand side l as in the right-hand side r. A term equation system (TES) E is called variable preserving if all of its equations are variable preserving. The ground word problem is undecidable even for variable-preserving TESs, see Example 4.1.4 on page 60 in [1]. We recall the well known trivial semi-decision procedure *PRO1* for the ground word problem of variable preserving TESs and its straightforward generalization, the trivial semi-decision procedure *PRO2* for the ground word problem of TESs.

On the basis of *PRO1*, we give a semi-decision procedure *PRO3* for the ground word problem of variable preserving TESs. Given a TES E and ground terms p, q over the ranked alphabet Σ, procedure *PRO3* constructs the ground TESs (GTESs) P_i and Q_i, $i \geq 1$ such that

(a) $P_i \cup Q_i \subseteq \leftrightarrow_E^*$ for $i \geq 1$.

Condition (a) ensures that the congruence closure of $P_i \cup Q_i$ is a subset of $\leftrightarrow_E^*$. Procedure *PRO3* outputs an answer and halts if and only if

(b) there is a $j \geq 1$ such that

$p \leftrightarrow_{P_j \cup Q_j}^* q$ or

$\leftrightarrow_{P_j}^* \cap (\{ p \} \times T_\Sigma) = \leftrightarrow_E^* \cap (\{ p \} \times T_\Sigma)$ or

*Department of Foundations of Computer Science, University of Szeged, Árpád tér 2, H-6720 Szeged, Hungary. E-mail: vagvolgy@inf.u-szeged.hu

DOI: 10.14232/actacyb.22.2.2015.16

$\leftrightarrow^*_{Q_j} \cap (\{ q \} \times T_\Sigma) = \leftrightarrow^*_E \cap (\{ q \} \times T_\Sigma)$.
Condition (b) says that we have a proof of $p \leftrightarrow^*_E q$, or the intersection of $\leftrightarrow^*_{P_j}$ with $(\{ p \} \times T_\Sigma)$ is equal to that of $\leftrightarrow^*_E$, or the intersection of $\leftrightarrow^*_{Q_j}$ with $(\{ q \} \times T_\Sigma)$ is equal to that of $\leftrightarrow^*_E$. Assume that (b) holds. If $p \leftrightarrow^*_{P_j \cup Q_j} q$ holds, *PRO3* outputs 'yes', and halts. Otherwise, if

- the intersection of $\leftrightarrow^*_{P_j}$ with $(\{ p \} \times T_\Sigma)$ is equal to that of $\leftrightarrow^*_E$, or
- the intersection of $\leftrightarrow^*_{Q_j}$ with $(\{ q \} \times T_\Sigma)$ is equal to that of $\leftrightarrow^*_E$,

then $p \leftrightarrow^*_E q$ does not hold either. Hence semi-decision procedure *PRO3* outputs 'no' and halts.

Procedure *PRO3* constructs the ground TESs (GTESs) P_i and Q_i, $i \geq 1$ in the following way. We put a ground instance $l' \approx r'$ of an equation $l \approx r$ of $E \cup E^{-1}$ in P_1 if l' is a subterm of p. Then we iterate the following computation items.

- We convert the GTES P_i into an equivalent reduced ground term rewrite system R_i applying Snyder's fast ground completion algorithm [19].
- We define the GTES P_{i+1} from the reduced ground term rewrite system R_i by adding all ground instances $l \approx r$ of equations in $E \cup E^{-1}$ such that

- $l \approx r$ is not in $\leftrightarrow^*_{P_i}$ and that

- there exists a term s such that the conversion $p \leftrightarrow^*_{P_i} s$ can be continued applying $l \approx r$ to s. If $P_{i+1} = R_i$, then we let $R_{i+1} = R_i$, and hence $R_i = P_j = R_j$ holds for $j \geq i + 1$.

Here we consider both the reduced ground term rewrite system R_i and the GTES P_{i+1} as subsets of $T_\Sigma \times T_\Sigma$. Furthermore, we consider a ground instance of an equation in $E \cup E^{-1}$ as an element of $T_\Sigma \times T_\Sigma$.

We define the GTES Q_i symmetrically to P_i for $i \geq 1$.

Procedure *PRO3* computes in the following way. For each $i = 1, 2, \ldots$,

- if $p \leftrightarrow^*_{P_i \cup Q_i} q$, then we output the answer 'yes' and halt;
- otherwise, if $i \geq 2$ and we did not add ground instances of equations in $E \cup E^{-1}$ to the reduced ground term rewrite system $R_{P_{i-1}}$, equivalent to P_{i-1}, or to the reduced ground term rewrite system R_{Q_i}, equivalent to Q_{i-1}, in the previous iteration step, then we output the answer 'no' and halt.

Assume that $p \leftrightarrow^*_E q$. Then, at some step during the run of procedure *PRO3*, $p \leftrightarrow^*_{P \cup Q} q$ becomes true, and procedure *PRO3* outputs 'yes' and halt. If $p \leftrightarrow^*_E q$ does not hold, then procedure *PRO3* either outputs 'no' and halts or runs forever.

We give a semi-decision procedure *PRO4* for the ground word problem of TESs. We obtain it generalizing *PRO3* taking into account *PRO2*. The main difference is the following. We define P_{i+1} from R_i by adding all ground instances $l' \approx r'$ of the equations $l \approx r$ in $E \cup E^{-1}$ such that

- $l' \approx r'$ is not in $\leftrightarrow^*_{P_i}$, that
- there exists a term s such that a conversion $p \leftrightarrow^*_{P_i} s$ can be continued applying $l \approx r$ to s, and that
- we substitute some finitely many ground terms depending on i, R_i, and p, for those variables in r that do not appear in l.

We modify the halting condition of the proceedure so that it stops if we did not add ground instances of equations in $E \cup E^{-1}$ to P_i or Q_i in two successive iteration steps. We need two successive steps rather than one. Because, in general, the heights of the substituted terms becomes larger in each step. If we do not add ground equations to P_i in a step, then in the next step we still may add ground equations to P_i.

Procedures *PRO3* and *PRO4* compute in a different way than all versions of the Knuth-Bendix completion procedure. To some instances of the ground word problem of a TES E, procedures *PRO3* and *PRO4* give an answer sooner than all versions of the Knuth-Bendix completion procedure or it is open whether some version of the Knuth-Bendix completion procedure gives an answer at all. Consequently, they may compute efficiently for some instances of the ground word problem of a TES E, when the various versions of the Knuth-Bendix completion procedure does not give an answer to the ground word problem of a TES E at all or at least not in a reasonable time. However, it is still open in which cases are *PRO3* and *PRO4* really efficient.

In Section 2, we present a brief review of the notions, notations, and preliminary results used in the paper. In Section 3 we introduce and study the concept of reading-up reachability for reduced ground term rewriting systems. In Section 4 we present the procedures *PRO1* and *PRO2*. In Section 5, we present the procedure *PRO3*, and show its correctness. We give examples when procedure *PRO3* is more efficient than procedure *PRO1*. In Section 6, we present the procedure *PRO4*, and show its correctness. In Section 7, we compare procedures *PRO3* and *PRO4* with the basic Knuth-Bendix completion procedure (see Section 7.1 in [1]), an improved version of the Knuth-Bendix completion procedure described by a set of inference rules (see Section 7.2 in [1]), the goal-directed completion procedure based on SOUR graphs [13, 14], and the unfailing Knuth-Bendix completion procedure [2]. In Section 8, we sum up our results, and explain the applicability of procedures *PRO3* and *PRO4*.

2 Preliminaries

In this section we present a brief review of the notions, notations and preliminary results used in the paper. For all unexplained notions and notation see [1].

Relations. Let ρ be an equivalence relation on A. Then for every $a \in A$, we denote by a/ρ the ρ-class containing a, i.e. $a/\rho = \{\, b \mid a\rho b \,\}$. For each $B \subseteq A$, let $B/\rho = \{\, b/\rho \mid b \in B \,\}$.

2.1 Abstract Reduction Systems

An abstract reduction system is a pair $(A, \rightarrow)$, where the reduction $\rightarrow$ is a binary relation on the set A. $\rightarrow^{-1}$, $\leftrightarrow$, $\rightarrow^*$, and $\leftrightarrow^*$ denote the inverse, the symmetric closure, the reflexive transitive closure, and the reflexive transitive symmetric closure of the binary relation $\rightarrow$, respectively.

- $x \in A$ is reducible if there is y such that $x \rightarrow y$.
- $x \in A$ is irreducible if it is not reducible.

• $y \in A$ is a normal form of $x \in A$ if $x \to^* y$ and y is irreducible. If $x \in A$ has a unique normal form, the latter is denoted by $x{\downarrow}$.

• $y \in A$ is a descendant of $x \in A$ if $x \to^* y$.

• $x \in A$ and $y \in A$ are joinable if there is a z such that $x \to^* z \leftarrow^* y$, in which case we write $x{\downarrow}y$.

The reduction $\to$ is called

• confluent if for all $x, y_1, y_2 \in A$, if $y_1 \leftarrow^* x \to^* y_2$, then $y_1{\downarrow}y_2$;

• locally confluent if for all $x, y_1, y_2 \in A$, if $y_1 \leftarrow x \to y_2$, then $y_1{\downarrow}y_2$;

• terminating if there is no infinite chain $x_0 \to x_1 \to x_2 \to \cdots$;

• convergent if it is both confluent and terminating.

If $\to$ is convergent, then each $x \in A$ has a unique normal form [1].

Terms. A ranked alphabet Σ is a finite set of symbols in which every element has a unique rank in the set of nonnegative integers. For each integer $m \geq 0$, Σ_m denotes the elements of Σ which have rank m.

Let Y be a set of variables. The set of terms over Σ with variables in Y is denoted by $T_\Sigma(Y)$. The set $T_\Sigma(\emptyset)$ is written simply as T_Σ and called the set of ground terms over Σ. We specify a countably infinite set $X = \{ x_1, x_2, \dots \}$ of variables which will be kept fixed in this paper. Moreover, we put $X_n = \{ x_1, x_2, \dots, x_n \}$, for $n \geq 0$. Hence $X_0 = \emptyset$. For any $i \geq 1$ and $j \geq 0$, let $X_{[i,j]} = \emptyset$ if $i > j$, and let $X_{[i,j]} = \{ x_i, x_{i+1}, \dots, x_j \}$ otherwise.

For a term $t \in T_\Sigma(X)$, the height $height(t) \in N$ is defined by recursion:

(a) if $t \in \Sigma_0 \cup X$, then $height(t) = 0$,

(b) if $t = \sigma(t_1, \dots, t_m)$ with $m \geq 1$ and $\sigma \in \Sigma_m$, then

$$height(t) = 1 + max(height(t_i) \mid 1 \leq i \leq m).$$

For each $k \geq 0$, $HE_{\Sigma,\leq k}(X) = \{ t \in T_\Sigma(X) \mid height(t) \leq k \}$.

Let N be the set of all positive integers. N^* stands for the free monoid generated by N with empty word λ as identity element. For each word $\alpha \in N^*$, $length(\alpha)$ stands for the length of α. Consider the words $\alpha, \beta, \gamma \in N^*$ such that $\alpha = \beta\gamma$. Then we say that β is a prefix of α. Furthermore, if $\alpha \neq \beta$, then β is a proper prefix of α. For a term $t \in T_\Sigma(X)$, the set $Pos(t) \subseteq N^*$ of positions is defined by recursion:

(i) if $t \in \Sigma_0 \cup X$, then $Pos(t) = \{ \lambda \}$, and

(ii) if $t = \sigma(t_1, \dots, t_m)$ with $m \geq 1$ and $\sigma \in \Sigma_m$, then $Pos(t) = \{ \lambda \} \cup \{ i\alpha \mid 1 \leq i \leq m \text{ and } \alpha \in Pos(t_i) \}$.

For each term $t \in T_\Sigma(X)$, $size(t)$ is the cardinality of $Pos(t)$.

For each $t \in T_\Sigma(X)$ and $\alpha \in Pos(t)$, we introduce the subterm $t/\alpha \in T_\Sigma(X)$ of t at α as follows:

(a) for $t \in \Sigma_0 \cup X$, $t/\lambda = t$;

(b) for $t = \sigma(t_1, \dots, t_m)$ with $m \geq 1$ and $f \in \Sigma_m$, if $\alpha = \lambda$ then $t/\alpha = t$, otherwise, if $\alpha = i\beta$ with $1 \leq i \leq m$, then $t/\alpha = t_i/\beta$

For any $t \in T_\Sigma(X)$, $\alpha \in Pos(t)$, and $r \in T_\Sigma(X)$, we define $t[\alpha \leftarrow r] \in T_\Sigma(X)$.

(i) If $\alpha = \lambda$, then $t[\alpha \leftarrow r] = r$.

(ii) If $\alpha = i\beta$, for some integer i, then $t = \sigma(t_1, \ldots, t_m)$ with $f \in \Sigma_m$ and $1 \leq i \leq m$. Then $t[\alpha \leftarrow r] = \sigma(t_1, \ldots, t_{i-1}, t_i[\beta \leftarrow r], t_{i+1}, \ldots, t_m)$.

For a term $t \in T_\Sigma(X)$, the set $sub(t)$ of subterms of t is defined as $sub(t) = \{ t/\alpha \mid \alpha \in Pos(t) \}$.

Given a term $t \in T_\Sigma(X_n)$, $n \geq 0$, and terms $t_1, \ldots, t_n$, we denote by $t[t_1, \ldots, t_n]$ the term which can be obtained from t by replacing each occurrence of x_i in t by t_i for $1 \leq i \leq n$. A context is a term $u \in T_{\Sigma \cup \{ \diamond \}}$, where the nullary symbol $\diamond$ appears exactly once in u. We denote the set of all contexts over Σ by C_Σ. For a context u and a term t, $u[t]$ is defined from u by replacing the occurrence of $\diamond$ with t.

For the sake of simplicity, we may write unary terms as strings. For example, we write $fgh\#$ for the term $f(g(h(\#)))$ and f^3x_1 for $f(f(f(x_1)))$, where f, g, h are unary symbols and $\#$ is a nullary symbol.

Algebras. Let Σ be a ranked alphabet. A Σ algebra is a system $\mathbf{B} = (B, \Sigma^{\mathbf{B}})$, where B is a nonempty set, called the carrier set of $\mathbf{B}$, and $\Sigma^{\mathbf{B}} = \{ f^{\mathbf{B}} \mid f \in \Sigma \}$ is a Σ-indexed family of operations over B such that for every $f \in \Sigma_m$ with $m \geq 0$, $f^{\mathbf{B}}$ is a mapping from B^m to B. An equivalence relation $\rho \subseteq B \times B$ is a congruence on $\mathbf{B}$ if

$$f^{\mathbf{B}}(t_1, \ldots, t_m) \rho f^{\mathbf{B}}(p_1, \ldots, p_m)$$

whenever $f \in \Sigma_m$, $m \geq 0$, and $t_i \rho p_i$, for $1 \leq i \leq m$. For each $B' \subseteq B$, let $[B']_\rho = \{ [b]_\rho \mid b \in B' \}$. In this paper we shall mainly deal with the algebra $\mathbf{TA} = (T_\Sigma, \Sigma)$ of ground terms over Σ, where for any $f \in \Sigma_m$ with $m \geq 0$ and $t_1, \ldots, t_m \in T_\Sigma$, we have

$$f^{\mathbf{TA}}(t_1, \ldots, t_m) = f(t_1, \ldots, t_m)\,.$$

We now recall the concept of a set of representatives for a congruence ρ and a set of ρ-classes.

Definition 1. [6] Let ρ be a congruence on **TA** and let A be a set of ρ-classes. A set REP of ground terms is called a set of *representatives* for A if

- $REP \subseteq \bigcup A$,
- $\bigcup(sub(t) \mid t \in REP) \subseteq REP$, and
- each class $Z \in A$ contains exactly one term $t \in REP$.

Term equation systems. Let Σ be a ranked alphabet. A term equation system (TES for short) E over Σ is a finite subset of $T_\Sigma(X) \times T_\Sigma(X)$. Elements (l, r) of E are called equations and are denoted by $l \approx r$. The reduction relation $\rightarrow_E \subseteq T_\Sigma(X) \times T_\Sigma(X)$ is defined as follows. For any terms $s, t \in T_\Sigma(X)$, $s \rightarrow_E t$ if there is a pair $l \approx r$ in E and a context $u \in C_\Sigma(X_1)$ and a substition δ such that $s = u[\delta(l)]$ and $t = u[\delta(r)]$. When we apply an arbitrary equation $l \approx r \in E \cup E^{-1}$, we rename the variables of l and r such that $l \in T_\Sigma(X_{k+m})$ and $r \in T_\Sigma(X_k \cup X_{[k+m+1, k+m+\ell]})$ for some $k, m, \ell \geq 0$.

The word problem for a TES E is the problem of deciding for arbitrary $p, q \in T_\Sigma(X)$ whether $p \leftrightarrow_E^* q$. The ground word problem for E is the word problem restricted to ground terms p and q.

For the notion of a term rewriting system (TRS), see Section 4.2 in [1]

Knuth-Bendix completion procedure. We now briefly recall the basic Knuth-Bendix completion procedure, see Section 7.1 in [1]. The basic Knuth-Bendix completion procedure starts with a TES E and tries to find a convergent TRS R that is equivalent to E. A reduction order $>$ is provided as an input for the procedure. Since the word problem is not decidable in general, a finite convergent TRS cannot always be obtained. In the basic Knuth-Bendix completion procedure this could be due to failure or to non-termination of completion. In the initialization phase, the basic completion procedure removes trivial identities of the form $s = s$ and tries to orient the remaining nontrivial identities. If this succeeds, then it computes all critical pairs of the TRS obtained. The terms in each critical pair $\langle s, t\rangle$ are reduced to their normal forms $\hat{s}$ and $\hat{t}$. If the normal forms are identical, then this critical pair is joinable, and nothing needs to be done for it. Otherwise, the procedure tries to orient the terms $\hat{s}$ and $\hat{t}$ into the rewrite rule $\hat{s} \to \hat{t}$ with $\hat{s} > \hat{t}$ or $\hat{t} \to \hat{s}$ with $\hat{t} > \hat{s}$. In this way the procedure orients all instances of the terms $\hat{s}$ and $\hat{t}$ as well. If this succeeds, then the new rule is added to the current rewrite system. This process is iterated until failure occurs or the rewrite system is not changed during a step of the iteration, that is, the system does not have non-joinable critical pairs.

If the basic completion procedure applied to $(E, >)$ terminates succesfully with output R, then R is a finite convergent TRS that is equivalent to E. In this case, R yields a decision procedure for the word problem for E. If the basic completion procedure applied to $(E, >)$ does not terminate, then it outputs an infinite convergent TRS that is equivalent to E. In this case, the completion procedure can be used as a semidecision procedure for the word problem for E.

Assume that we want to decide for given terms $p, q \in T_\Sigma(X)$, whether $p \leftrightarrow_E^* q$ holds. We call the pair (p, q) the *goal*. The basic Knuth-Bendix completion procedure is independent of the goal. Hence, if $p \leftrightarrow_E^* q$ does not hold, and the set E of equations has no finite convergent system, then the basic Knuth-Bendix completion will run forever. In the light of this observation, Lynch and Strogova [13, 14] presented a goal-directed completion procedure based on SOUR graphs. Similarly to the basic Knuth-Bendix completion procedure, the goal-directed completion procedure uses a reduction order $>$. Unlike the basic Knuth-Bendix completion procedure, it uses some inference rules. The main difference, described in an intuitive simplified way, is the following. Along the completion procedure, we try to construct a rewrite system R and a conversion

$$p = r_1 \underset{R}{\leftrightarrow} r_2 \underset{R}{\leftrightarrow} \cdots \underset{R}{\leftrightarrow} r_n = q,\ n \geq 1 \tag{1}$$

in a nondeterministic way. We compute and orient critical pairs and control the completion process keeping in our mind that the rules of R should be applicable along a conversion (1). When orienting the equations into rules along the completion process, we do not put a rule in R if it is not applicable along a conversion (1). If we do not find a conversion (1), the goal-directed completion procedure detects that $(p, q) \notin \leftrightarrow_E^*$, outputs 'no' and halts. Consider the following example. Let ranked alphabet Σ consist of the unary symbols f, g and the nullary symbols \$, #. Consider the variable preserving TES $E = \{ ffx \approx gfx \}$. We raise the problem whether $\$ \leftrightarrow_E^* \#$. The basic Knuth-Bendix completion procedure runs forever on this example [13]. Along the goal oriented completion procedure, we find no rewrite rule such that it is applicable along a conversion $\$ = r_1 \leftrightarrow_R r_2 \leftrightarrow_R \cdots \leftrightarrow_R r_n = \#$, $n \geq 1$. Therefore, the goal-directed completion procedure detects that $(\$, \#) \notin \leftrightarrow_E^*$, outputs 'no', and halts [13].

We now adopt a more detailed description of the goal-directed completion procedure. [14] The goal-directed completion procedure uses a reduction order $>$ and computes critical pairs equipped with equational and ordering constraints, and constructs a graph. "The goal-directed completion procedure has two phases. The first phase is the compilation phase. In this phase, all the edges and the recursive constraints labelling each edge are created. This phase also takes into account the goal to be solved. Importantly, this phase takes only polynomial time, because there are only polynomially many edges in the graph. The result of this phase is a constrained tree automaton representing a schematized version of the completed system, and a set of constraints representing potential solutions to the goal. The constraints that are generated are the equational constraints representing the unification problems, and ordering constraints arising from the critical pair inferences.

The second phase is the goal solving (or constraint solving) phase. In this phase, the potential solutions to the goal are solved in order to determine whether they are actual solutions of the goal. This phase can take infinitely long, since the constraints are recursive. Step by step a constraint is rolled back, based on which edges it is created from, and the equational and ordering constraints are solved along the way. In some cases, the ordering constraints cause the recursion to halt, and therefore the constraints are completely solved. The procedure is truly goal oriented, because only a polynomial amount of time is spent compiling the set of equations. The rest of the time is spent working backwards from the goal to solve the constraints. If the procedure is examined more closely, we see that the second phase of the procedure is exactly a backwards process of completion. A schematization of an equation in the completed system is applied to the goal, step by step until it rewrites to an identity. At the same time, the schematized equation that is selected is worked backwards until we reach the original equations from which it is formed." [14]

See Section 7.2 in [1] for an improved version of the Knuth-Bendix completion procedure described by a set of inference rules. A detailed description of the unfailing Knuth-Bendix completion procedure can be found in [2].

Ground term equation systems and rewriting systems. A ground term equation system (GTES) E over a ranked alphabet Σ is a finite binary relation on T_Σ. Elements (l, r) of E are called equations and are denoted by $l \approx r$. The reduction relation $\rightarrow_E \subseteq T_\Sigma(X) \times T_\Sigma(X)$ is defined as follows. For any ground terms $s, t \in T_\Sigma$, $s \rightarrow_E t$ if there is a pair $l \approx r$ in E and a context $u \in C_\Sigma(X_1)$

such that $s = u[l]$ and $t = u[r]$. It is well known that the relation $\leftrightarrow_E^*$ is a congruence on the term algebra **TA** [18]. We call $\leftrightarrow_E^*$ the congruence induced by E. The size of E is defined as the number of occurrences of symbols in the set. $sub(E) = \{ sub(l) \mid l \approx r \in E \cup E^{-1} \}$. Clearly, $\leftrightarrow_E^* \cap (sub(E) \times sub(E))$ is an equivalence relation on $sub(E)$. The word problem for a GTES E is the problem of deciding for arbitrary $p, q \in T_\Sigma$ whether $p \leftrightarrow_E^* q$.

A ground term rewrite system (GTRS) over a ranked alphabet Σ is a finite subset R of $T_\Sigma \times T_\Sigma$. The elements of R are called rules and a rule $(l, r) \in R$ is written in the form $l \to r$ as well. Moreover, we say that l is the left-hand side and r is the right-hand side of the rule $l \to r$. $lhs(R) = \{l \mid l \to r \in R\}$, $rhs(R) = \{r \mid l \to r \in R\}$. $sub(R) = \{ sub(l) \mid l \in lhs(R) \} \cup \{ sub(r) \mid r \in lhs(R) \}$.

The reduction relation $\to_R \subseteq T_\Sigma(X) \times T_\Sigma(X)$ is defined as follows. For any ground terms $s, t \in T_\Sigma$, $s \to_R t$ if there is a pair $l \approx r$ in E and a context $u \in C_\Sigma(X_1)$ such that $s = u[l]$ and $t = u[r]$. Here we say that R rewrites s to t applying the rule $l \to r$. A GTRS R is *equivalent to* a GTRS E, if $\leftrightarrow_R^* = \leftrightarrow_E^*$ holds.

$IRR(R)$ denotes the set of all ground terms irreducible by R. A GTRS R is reduced if for every rule $u \to v$ in R, u is irreducible with respect to $R - \{ u \to v \}$ and v is irreducible with respect to R. For a reduced GTRS R, $IRR(R) \cap sub(R) = sub(R) - lhs(R)$, and $sub(R) - lhs(R)$ is a set of representatives for $sub(R)/\leftrightarrow_R^*$, see Theorem 3.14 on page 162 in [17].

We say that a GTRS R is confluent, locally confluent, terminating, or convergent, if $\to_R$ has the corresponding property.

We recall the following important result.

Proposition 1. [19] Any reduced GTRS R is convergent.

Proposition 2. For a reduced GTRS R, one can reduce a ground term $t \in T_\Sigma$ to its normal form in linear time of $size(t)$. We traverse the term t in postorder. When visiting a position α, we reduce the subterm t/α of t at α to is normal form $t/\alpha\downarrow_R$.

We say that a GTRS R is equivalent to a GTES E if $\leftrightarrow_R^* = \leftrightarrow_E^*$.

Proposition 3. [19] *For a GTES E one can effectively construct an equivalent reduced GTRS R in $O(n \log n)$ time. Here n is the size of E.*

Proof. We briefly recall Snyder's [19] fast ground completion algorithm. We run a congruence closure algorithm for E over the subterm graph of E [4, 15]. In this way we get the representation of the equivalence relation $\leftrightarrow_E^* \cap (sub(E) \times sub(E))$. We compute a set REP of representatives for $sub(E)/\leftrightarrow_E^*$. Then we construct a reduced GTRS R over Σ as follows. We put the rewrite rule $l \to r$ in R if

- $l = f(p_1, \ldots, p_m)$ for some $f \in \Sigma_m$, $m \geq 0$, and $p_1, \ldots, p_m \in REP$,
- $r \in REP$,
- $l \neq r$ and $l \leftrightarrow_E^* r$.

□

We can decide the word problem of a GTES E applying a congruence closure algorithm [4, 15] for the GTES $E_1 = E \cup \{ p \approx p, q \approx q \}$ and then examine whether

p, q are in the same class of the equivalence relation $\leftrightarrow^*_{E_1} \cap (sub(E_1) \times sub(E_1))$. Assume that we want to solve the word problem of a fixed GTES E for varying terms p, q. Then we compute a convergent GTRS over Σ equivalent to E [8, 14, 16, 19]. We compute $p\downarrow_R$ and $q\downarrow_R$, and compare them. If $p\downarrow_R = q\downarrow_R$, then $p \leftrightarrow^*_E q$. Otherwise, $(p, q) \notin \leftrightarrow^*_E$. By Proposition 2, we can decide the word problem of E in linear time. We can also extend the signature. We introduce constants for the equivalence classes of $\leftrightarrow^*_E \cap (sub(E) \times sub(E))$. Then we can construct in $O(n \log n)$ time a reduced GTRS over the extended signature such that $p\downarrow_R = q\downarrow_R$ if and only if $p \leftrightarrow^*_E q$. By Proposition 2, we can decide the word problem of E in linear time. Finally, assume that we want to solve the word problem of a fixed GTES E for a fixed term p and varying term q. Then we can construct in $O(n \log n)$ time a deterministic tree automaton recognizing the $\leftrightarrow^*_E$-class of p [17].
For other completion algorithms on GTRSs see [5, 16]. For further results on GTRSs see [18]. Proposition 1 and Proposition 3 imply the following well known result.

Proposition 4. [19] *For a GTES E and ground terms p, q, one can decide whether $p \leftrightarrow^*_E q$.*

3 Reachability starting from a term attached to a context

Let R be a reduced GTRS over Σ and $s, t \in IRR(R)$. We say that R reaches t starting from s attached to some context, if there is a $u \in C_\Sigma$ such that $u[s] \to^*_R t$. Let $RAC(s)$ denote the set of all terms $t \in IRR(R)$ which are reachable by R starting from s attached to some context.

Example 1. Let $\Sigma = \Sigma_0 \cup \Sigma_1$, $\Sigma_0 = \{0, 1\}$, and $\Sigma_2 = \{f\}$. Let GTRS R consist of the equations $f(0, 0) \to 0$ and $f(0, 1) \to 1$. Clearly R is reduced. Then each element of $IRR(R)$ containing 0 is in $RAC(0)$. For example, $f(f(1, 0), 1) \in RAC(0)$, because $f(f(1, \diamond), 1) \in C_\Sigma$ and

$f(f(1, \diamond), 1)[0] = f(f(1, 0), 1) \to^*_R f(f(1, 0), 1)$.

Furthermore, $1 \in RAC(0)$, because

$f(\diamond, 1)[0] = f(0, 1) \to_R 1$.

Thus each element of $IRR(R)$ containing 1 is in $RAC(0)$. Consequently, $IRR(R) = RAC(0)$.

Lemma 1. *Let R be a reduced GTRS over Σ. For any $s \in sub(R) - lhs(R)$, we can effectively compute $RAC(s) \cap (sub(R) - lhs(R))$.*

Proof. Let $RAC_0 = \{s\}$. For each $i \geq 0$, let RAC_{i+1} consists of all elements t, where

- $t \in RAC_i$ or
- $t \in sub(R) - lhs(R)$ and there is a rule $f(t_1, \ldots, t_m) \to t$ in R for some $f \in \Sigma_m$, $t_1, \ldots, t_m \in sub(R) - lhs(R)$, such that $t_j \in RAC_i$ for some $1 \leq j \leq m$, or

• $t \in sub(R) - lhs(R)$ and $t = f(t_1, \ldots, t_m)$ for some $f \in \Sigma_m$, $t_1, \ldots, t_m \in sub(R) - lhs(R)$, and $t_j \in RAC_i$ for some $1 \leq j \leq m$. Then

$$RAC_i \subseteq RAC_{i+1} \subseteq RAC(s) \cap (sub(R) - lhs(R)) \text{ for } i \geq 0\,. \tag{2}$$

Hence there is an integer $0 \leq \ell \leq card(sub(R) - lhs(R))$ such that $RAC_\ell = RAC_{\ell+1}$. Then

$$RAC_\ell = RAC_{\ell+k} \text{ for } k \geq 1\,. \tag{3}$$

Hence

$$RAC_\ell \subseteq RAC(s) \cap (sub(R) - lhs(R))\,. \tag{4}$$

To show the reverse inclusion, we need the following.

Claim 1. *For any $u \in C_\Sigma$ of height $n \geq 0$ and $t \in sub(R) - lhs(R)$, if $u(s) \to_R^* t$, then $t \in RAC_n$.*

Proof. By induction on n.

□

By (2), (3), and Claim 1, $RAC(s) \cap (sub(R) - lhs(R)) \subseteq RAC_\ell$. By (4),

$$RAC(s) \cap (sub(R) - lhs(R)) = RAC_\ell\,.$$

We compute the sets $RAC_0, RAC_1, \ldots, R_{card(sub(R)-lhs(R))}$. In this way we obtain the integer ℓ and $RAC(s) \cap (sub(R) - lhs(R))$.

□

Lemma 2. *For any reduced GTRS R and $s, t \in IRR(R)$, R reaches t starting from s attached to some context if and only if*

(i) $t = u[s]$ for some $u \in C_\Sigma$ or

(ii) $s \in (sub(R) - lhs(R))$, and there are $u \in C_\Sigma$ and $r \in rhs(R)$ such that $t = u[r]$ and R reaches r starting from s attached to some context.

Proof. ($\Rightarrow$) Assume that R reaches t starting from s attached to some context. Then there is $u \in C_\Sigma$ such that $u[s] \to_R^* t$. If $u[s] = t$, then (i) holds. Otherwise, $u[s] \to_R^+ t$. Hence there are $v_1, v_2, z \in C_\Sigma$ and a rule $l \to r$ in R such that $u[s] = v_1[z[s]] \to_R^* v_1[l] \to_R v_1[r] \to_R^* v_2[r] = t$, where

(a) $u = v_1[z]$,

(b) $z[s] \to_R^* l$,

(c) $l \to r \in R$,

(d) $v_1 \to_R^* v_2$ over the ranked alphabet $\Sigma \cup \diamond$.

Hence $t = v_2[r]$, $v_2 \in C_\Sigma$, $r \in rhs(R)$. By (b), $s \in sub(l)$ or $s \in sub(l_1)$ for some $l_1 \in LHS(R)$. Recall that $s \in IRR(R)$. Hence $s \in (sub(R) - lhs(R))$.

($\Leftarrow$) If (i) holds, then R reaches t starting from s attached to some context.

Assume that (ii) holds. Then there is $z \in C_\Sigma$ such that $z[s] \to_R^* r$. Consequently $(u[z])[s] = u[z[s]] \to_R^* u[r] = t$. Hence R reaches t starting from s attached to some context.

□

Lemma 1 and Lemma 2 imply the following result.

Proposition 5. *For any $s, t \in IRR(R)$, we can decide whether R reaches t starting from s attached to some context.*

4 Two trivial semi-decision procedures

We present the well known trivial semi-decision procedure *PRO1* for the ground word problem of variable preserving TESs. We give examples when *PRO1* is efficient. Then we present the trivial semi-decision procedure *PRO2* for the ground word problem of TESs. Note that *PRO2* is a straightforward generalization of *PRO1*.

Procedure *PRO1* Input: A variable preserving TES E over the ranked alphabet Σ and ground terms $p, q \in T_\Sigma$.
Output: 'yes' if $p \leftrightarrow_E^* q$, 'no' or undefined otherwise.
Let $U_0 = \{ p \}$, $V_0 = \{ q \}$, $i = 0$.
repeat
  $i := i + 1$;
  $U_i := U_{i-1} \cup \{ s |$ there is $u \in U_{i-1}$ such that $u \leftrightarrow_E s \}$;
  $V_i := V_{i-1} \cup \{ s |$ there is $u \in V_{i-1}$ such that $u \leftrightarrow_E s \}$;
until ($U_i = U_{i-1}$ or $V_i = V_{i-1}$) or $U_i \cap V_i$ is not empty;
  if $U_i \cap V_i$ is not empty
  then begin output 'yes'; halt end;
output 'no';
halt

For any variable preserving TES E and ground term u, the set $\{ s | u \leftrightarrow_E s \}$ is finite and then effectively computable. Thus for every $i \geq 0$, U_i and V_i, are finite and can be computed effectively. Hence the above procedure can be implemented. Clearly, *PRO1* outputs 'yes' and halts if and only if $p \leftrightarrow_E^* q$. If *PRO1* outputs 'no' and halts, then $(p, q) \notin \leftrightarrow_E^*$.

We adopt the following example of Lynch [13].

Example 2. Let $\Sigma = \Sigma_0 \cup \Sigma_1$, $\Sigma_0 = \{ \$, \# \}$, $\Sigma_1 = \{ f, g \}$. Consider the TES $E = \{ ffx \approx gfx \}$. We raise the problem whether $\$ \leftrightarrow_E^* \#$. On the one hand, the basic Knuth-Bendix completion procedure runs forever on this example [13]. On the other hand, the goal-directed completion procedure outputs 'no' and halts [13]. It is still open whether the goal-directed completion procedure halts on the TES E and any goal [13].

Observe that for each $u \in T_\Sigma$, the set $\{ s \mid u \leftrightarrow_E^* s \}$ is finite. Hence for any $p, q \in T_\Sigma$, *PRO1* outputs the correct answer and halts. For this example, *PRO1* is more efficient than the basic Knuth-Bendix completion procedure, and is at least as efficient as the goal-directed completion procedure [13, 14].

Example 3. Let $\Sigma = \Sigma_0 \cup \Sigma_2$, $\Sigma_0 = \{ \star, \$, \# \}$, and $\Sigma_2 = \{ f \}$. We define the terms $comb_i \in T_\Sigma(X_i)$, $i \geq 1$, as follows. Let $comb_1 = f(x_1, \star)$, $comb_{i+1} = f(x_1, comb_i[x_2, \ldots, x_{i+1}])$ for $i \geq 1$. For example, $comb_3 = f(x_1, f(x_2, f(x_3, \star)))$. Let $n \geq 1$, $p = comb_{2n}[\#, \ldots, \#]$, and $q = comb_{2n}[\$, \ldots, \$]$. We run procedure *PRO1* on the TES $E = \{ \# \approx \$ \}$ and the ground terms p and q. Then

$$card(U_i) = card(V_i) = \binom{2n}{i} + \binom{2n}{i-1} + \cdots + \binom{2n}{1} \text{ for } i = 1, \ldots n,$$

$U_i \cap V_i = \emptyset$ for $i = 0, 1, \ldots n-1$, and
$comb_{2n}[\#, \ldots, \#, \$, \ldots, \$] \in U_n \cap V_n$.
Hence in the nth step, *PRO1* outputs 'yes' and halts.

Example 4. We present Ceitin's [3, 11] semi-Thue system as a TES. Let $\Sigma = \Sigma_0 \cup \Sigma_1$, $\Sigma_0 = \{ \$ \}$, and $\Sigma_1 = \{ a, b, c, d, e \}$. E consists of the equations
$acx_1 \approx cax_1$, $adx_1 \approx dax_1$, $bcx_1 \approx cbx_1$, $bdx_1 \approx dbx_1$,
$ecax_1 \approx cex_1$, $edbx_1 \approx dex_1$,
$cdcax_1 \approx cdcaex_1$, $caaax_1 \approx aaax_1$, $daaax_1 \approx aaax_1$.

Proposition 6. [3, 11] *It is undecidable for an arbitrary given ground term $t \in T_\Sigma$ whether $t \leftrightarrow^*_E a^3\$$.*

We run procedure *PRO1* on the TES E and the ground terms $p = a^3\$$ and $q = edb\$$. We compute as follows.
$U_0 = \{ p \}$, $V_0 = \{ q \}$,
$U_1 = \{ a^3\$, ca^3\$, da^3\$ \}$, $V_1 = \{ edb\$, ebd\$, de\$ \}$,
$U_2 = \{ a^3\$, ca^3\$, da^3\$, cca^3\$, cda^3\$, dca^3\$, dda^3\$, acaa\$, adaa\$ \}$, $V_2 = V_1$.
Now procedure *PRO1* outputs 'no' and halts.

Let $n \geq 1$, $p = (bd)^{2n}\$$, and $q = (db)^{2n}\$$. We apply procedure *PRO1* to TES E and ground terms p and q. We compute as follows.
$U_0 = \{ p \}$, $V_0 = \{ q \}$,
$U_1 = \{ p, db(bd)^{2n-1}\$, \ldots, (bd)^{2n-1}db\$ \}$,
$V_1 = \{ q, bd(db)^{2n-1}\$, \ldots, (db)^{2n-1}bd\$, \}$,
$U_2 = U_1 \cup \{ dbdb(bd)^{2n-2}\$, dbbddb(bd)^{2n-3}\$, \ldots, (bd)^{2n-2}dbdb\$ \}$,
$V_2 = V_1 \cup \{ bdbd(db)^{2n-2}\$, bddbbd(db)^{2n-3}\$, \ldots, (db)^{2n-2}bdbd\$ \}$,
$\ldots$.
Observe that $U_i \cap V_i = \emptyset$ for $i = 0, 1, \ldots, n-1$. Clearly, $(bd)^n(db)^n\$ \in U_n \cap V_n$. After computing U_n and V_n, procedure *PRO1* outputs 'yes' and halts.

Example 5. We continue Example 4. Let $p \in T_\Sigma$ be arbitrary such that symbols a or c appear in p. Let $q \in T_\Sigma$ such that a, c do not appear in q. That is, only the constant \$ and the symbols b, d, or e appear in q.

Observe that the left-hand side and the right-hand side of the fourth and sixth rules do not contain a or c. Both sides of all other rules contain a or c. Hence for any reduction sequence
$p \rightarrow_R p_1 \rightarrow_R p_2 \rightarrow_{\ldots} \rightarrow_R p_n$, $n \geq 1$, for any $1 \leq i \leq n$, the term p_i contains the constant \$ and at least one a or c. Furthermore, along any reduction sequence $q \rightarrow_R q_1 \rightarrow_R q_2 \rightarrow_{\ldots} \rightarrow_R q_n$, $n \geq 1$, we only use the fourth and sixth equations. Consequently, the set $\{ v \in T_\Sigma \mid q \leftrightarrow^*_E v \}$ is finite. Furthermore neither a nor c appears in any element of the set $\{ v \in T_\Sigma \mid q \leftrightarrow^*_E v \}$. Thus

$$(p, q) \notin \underset{E}{\overset{*}{\leftrightarrow}}, \tag{5}$$

and $U_i \cap V_i = \emptyset$ for $i \geq 0$. Thus procedure *PRO1* outputs 'no' and halts on the input E, p, q.

Example 6. Let $\Sigma = \Sigma_0 \cup \Sigma_1$, $\Sigma_0 = \{ a \}$, and $\Sigma_1 = \{ f \}$. TES E consists of the equation $ffx \approx x$. We run procedure *PRO1* on TES E and ground terms $p = a$ and $q = fa$. We compute as follows.
$U_0 = \{ a \}$, $V_0 = \{ fa \}$,
$U_1 = \{ a, ffa \}$, $V_1 = \{ fa, f^3a \}$,
$U_2 = \{ a, ffa, f^4a \}$, $V_2 = \{ fa, f^3a, f^5a \}, \ldots$.
$U_0 \subset U_1 \subset U_2 \subset \cdots$,
$V_0 \subset V_1 \subset V_2 \subset \cdots$, and
$U_i \cap V_i = \emptyset$ for $i \geq 0$.
Hence procedure *PRO1* does not halt.

To present the semi-decision procedure *PRO2*, we define the sets $U_i \subseteq T_\Sigma$, $i \geq 0$, by recursion. Let $U_0 = \{ p \}$. Let $i \geq 1$. We put all elements of U_{i-1} in U_i. Moreover, we put in U_i all $s \in T_\Sigma$ such that

- $l' \approx r'$ is a ground instance of some equation $l \approx r$ in $E \cup E^{-1}$ obtained by substituting arbitrary ground terms of height less than or equal to $i - 1$ for all variables that do not appear in l,
- $v \in C_\Sigma$,
- $v[l'] \in U_{i-1}$ and $s = v[r']$.

We define $V_i \subseteq T_\Sigma$, $i \geq 0$, symmetrically to U_i, $i \geq 0$. Clearly for every $i \geq 0$, U_i and V_i are finite and can be computed effectively. Note that there may be an $i \geq 1$ such that $U_i = U_{i+1}$ and $U_{i+1} \subset U_{i+2}$.

Example 7. Let $\Sigma = \Sigma_0 \cup \Sigma_1$, $\Sigma_0 = \{ 0, 1 \}$, and $\Sigma_2 = \{ f \}$. Let TES E consist of the equations
$f(x_1, x_1) \approx 0$, $f(0, x_1) \approx x_1$.
Let $p = f(1, 0)$ and $q = f(1, f(1, 1))$. Then
$U_0 = \{ f(1, 0) \}$, $V_0 = \{ f(1, f(1, 1)) \}$,
$U_1 = \{ f(1, 0), f(f(0, 1), 0), f(1, f(0, 0)), f(1, f(1, 1)) \}$,
$V_1 = \{ f(1, f(1, 1)), f(f(0, 1), f(1, 1)), f(1, 0), f(1, f(f(0, 1), 1)),$
$f(1, f(1, f(0, 1))) \}$.

Procedure *PRO2* Input: A TES E over the ranked alphabet Σ and ground terms $p, q \in T_\Sigma$.
Output: 'yes' if $p \leftrightarrow_E^* q$, undefined otherwise.

```
1   i := i + 1;
    compute U_i and V_i;
    if U_i ∩ V_i is not empty then begin output 'yes'; halt end;
    goto 1
```

PRO2 outputs 'yes' and halts if and only if $p \leftrightarrow_E^* q$.

Example 8. We continue Example 7. We run procedure *PRO2* on TES E and ground terms p, q. We compute as follows. We compute U_0 and V_0. We observe that $U_0 \cap V_0$ is empty. Then we compute U_1 and V_1. We observe that $U_1 \cap V_1$ is not empty. Procedure *PRO2* outputs 'yes' and halts.

5 Semi-decision procedure for the ground word problem of variable preserving TESs

We present the semi-decision procedure *PRO3* for the ground word problem of variable preserving TESs, and show its correctness. *PRO3* is an improvement of *PRO1.* The starting idea is the following. For each $i \geq 1$, we construct the GTES P_i using those instances of equations in $E \cup E^{-1}$ which are applied to compute the set U_i. We improve this construction by defining P_i, $i \geq 2$, as the set of all instances of equations in $E \cup E^{-1}$ which can be applied to elements of $\{ s \in T_\Sigma \mid p \leftrightarrow^*_{P_{i-1}} s \}$ rather than to the elements of U_{i-1}. Furthermore, we define the GTES Q_i symmetrically. We give examples when procedure *PRO3* is more efficient than procedure *PRO1.*

Let E be a variable preserving TES over Σ, and let $p, q \in T_\Sigma$. We define the GTESs P_i and the reduced GTRSs R_i, $i \geq 1$, over Σ as follows.

For each equation $l \approx r$ of $E \cup E^{-1}$ with $l, r \in T_\Sigma(X_m)$, $m \geq 0$, and for any $u \in C_\Sigma$, $u_1, \ldots, u_m \in T_\Sigma$, if $p = u[l[u_1, \ldots, u_m]]$ then we put the equation $l[u_1, \ldots, u_m] \approx r[u_1, \ldots, u_m]$ in P_1. Applying Snyder's algorithm we compute a reduced GTRS R_1 equivalent to the GTES P_1, see Proposition 3.

Let $i \geq 1$. (a) We put each element of R_i into P_{i+1}.

(b) For each equation $l \approx r$ of $E \cup E^{-1}$, $l, r \in T_\Sigma(X_m)$, $m \geq 0$, for any $u_1, \ldots, u_m \in (sub(R_i) - lhs(R_i)) \cup sub(p\downarrow_{R_i})$, if R_i reaches $p\downarrow_{R_i}$ starting from $l[u_1, \ldots, u_m]\downarrow_{R_i}$ attached to some context, and $l[u_1, \ldots, u_m]\downarrow_{R_i} \neq r[u_1, \ldots, u_m]\downarrow_{R_i}$, then we put the equation $l[u_1, \ldots, u_m] \approx r[u_1, \ldots, u_m]$ in P_{i+1}.

If $P_{i+1} = R_i$, then let $R_{i+1} = R_i$. Otherwise, applying Snyder's algorithm, we compute a reduced GTRS R_{i+1} equivalent to the GTES P_{i+1}.

When misunderstanding may arise, we denote R_i as R_{P_i}. We define the GTESs Q_i, $i \geq 1$, symmetrically to the GTESs P_i, $i \geq 1$. Applying Snyder's algorithm, we compute a reduced GTRS $R_{P_i \cup Q_i}$ equivalent to the GTRS $R_{P_i} \cup R_{Q_i}$ for $i \geq 1$.

We illustrate our concepts and results by the following example.

Example 9. Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, $\Sigma_0 = \{ \$, \# \}$, $\Sigma_1 = \{ e, f, g, h \}$, and $\Sigma_2 = \{ d \}$. Let the TES E consist of the equations

$\# \approx \$, \quad g\$ \approx h\$, \quad d(hx_1, hx_1) \approx hx_1, \quad efhx_1 \approx hx_1.$

Observe that E is variable preserving. Let $p = efg\#$, $q = d(h\#, h\#)$.

First we compute the GTES P_i, $i \geq 1$. GTES P_1 consists of the equation $\# \approx \$$. Let Θ stand for $\leftrightarrow^*_{P_1} \cap (sub(P_1) \times sub(P_1))$. Then $sub(P_1)/\Theta = \{ \{ \#, \$ \} \}$ and $\{ \$ \}$ is a set of representatives for $sub(P_1)/\leftrightarrow^*_{P_1}$. GTRS R_1 consists of the rule $\# \to \$$.

GTES P_2 consists of the equations $\# \approx \$$, $\quad g\$ \approx h\$$. Let Θ stand for $\leftrightarrow^*_{P_2} \cap (sub(P_2) \times sub(P_2))$. Then $sub(P_2)/\Theta = \{ \{ \#, \$ \}, \{ g\#, g\$, h\#, h\$ \} \}$ and $\{ \$, h\$ \}$ is a set of representatives for $sub(P_2)/\leftrightarrow^*_{P_2}$. GTRS R_2 consists of the rules $\# \to \$$, $g\$ \to h\$$.

GTES P_3 consists of the equations

$\# \approx \$, \quad g\$ \approx h\$, \quad h\$ \approx d(h\$, h\$), \quad h\$ \approx efh\$.$

Let Θ stand for $\leftrightarrow^*_{P_3} \cap (sub(P_3) \times sub(P_3))$. Then

$sub(P_3)/\Theta = \{\,\{\,\#, \$\,\}, \{\,g\#, g\$, h\#, h\$, d(h\$, h\$), efh\$\}, \{\,fh\$\,\}\,\}$
and $\{\,\$, h\$, fh\$\,\}$ is a set of representatives for $sub(P_3)/\leftrightarrow^*_{P_3}$. R_3 consists of the rules

$\# \rightarrow \$, \quad g\$ \rightarrow h\$, \quad d(h\$, h\$) \rightarrow h\$, \quad efh\$ \rightarrow h\$.$

$P_4 = R_3$ and $R_4 = R_3$. Furthermore, $P_i = R_3$ and $R_i = R_3$ for $i \geq 4$.

Second, we compute the GTESs Q_i, $i \geq 1$. GTES Q_1 consists of the equations $\# \approx \$$, $d(h\#, h\#) \approx h\#$. GTRS R_{Q_1} consists of the rules $\# \rightarrow \$$, $d(h\$, h\$) \rightarrow h\$$.

GTES Q_2 consists of the equations $\# \approx \$$, $d(h\$, h\$) \approx h\$$, $efh\$ \approx h\$$.

GTRS R_{Q_2} consists of the rules $\# \rightarrow \$$, $d(h\$, h\$) \rightarrow h\$$, $efh\$ \rightarrow h\$$.

Observe that $R_{Q_2} = Q_i = R_{Q_i}$ for $i \geq 3$.

$R_{P_1 \cup Q_1} = R_{P_1}$, $R_{P_2 \cup Q_2} = R_{P_2} \cup R_{Q_2}$, and $R_{P_3 \cup Q_3} = R_{P_3}$. Then

$p{\downarrow}_{R_{P_1 \cup Q_1}} = efg\$$, $q{\downarrow}_{R_{P_1 \cup Q_1}} = h\$$,

$p{\downarrow}_{R_{P_2 \cup Q_2}} = h\$$, $q{\downarrow}_{R_{P_2 \cup Q_2}} = h\$$.

We get the following result by direct inspection of the definition of the GTES P_i, $i \geq 1$.

Lemma 3. *(a) For each* $i \geq 1$, $\leftrightarrow^*_{P_i} = \leftrightarrow^*_{R_i} \subseteq \leftrightarrow^*_{P_{i+1}} \subseteq \leftrightarrow^*_E$.
(b) If $R_i = P_{i+1}$ *for some* $i \geq 1$, *then* $R_i = P_j = R_j$ *for* $j \geq i + 1$.

Lemma 4. *For each* $i \geq 1$, *we can effectively construct the GTES* P_i.

Proof. By induction on i.

Base Case: $i = 1$. Clearly, we can construct P_1.

Induction Step: Let $i \geq 1$. Assume that we have constructed P_i. By Proposition 3, we can construct R_i. Consider item (b) in the definition of P_i. By Proposition 5, we can effectively decide whether R_i reaches $p{\downarrow}_{R_i}$ starting from $l[u_1 \ldots, u_m]{\downarrow}_{R_i}$ attached to some context. Hence we can construct P_{i+1} as well.

□

We now present our semi-decision procedure.

Procedure *PRO3 Input:* A variable preserving TES E over the ranked alphabet Σ and ground terms $p, q \in T_\Sigma$.
Output: • 'yes' if $p \leftrightarrow^*_E q$,
• 'no' if $(p, q) \notin \leftrightarrow^*_E$ and the procedure halts,
• undefined if the procedure does not halt.

compute P_1, R_{P_1}, Q_1, R_{Q_1}, and $R_{P_1 \cup Q_1}$;
if $p{\downarrow}_{R_{P_1 \cup Q_1}} = q{\downarrow}_{R_{P_1 \cup Q_1}}$, then begin output 'yes'; halt end;
$i := 1$;
1: $i := i + 1$;
compute P_i, R_{P_i}, Q_i, R_{Q_i}, and $R_{P_i \cup Q_i}$;
if $p{\downarrow}_{R_{P_i \cup Q_i}} = q{\downarrow}_{R_{P_i \cup Q_i}}$, then begin output 'yes'; halt end;
if $R_{P_{i-1}} = P_i$ or $R_{Q_{i-1}} = Q_i$,
then begin output 'no'; halt end;
goto 1

Example 10. We continue Example 9. Note that $p{\downarrow}_{R_{P_1\cup Q_1}} \neq q{\downarrow}_{R_{P_1\cup Q_1}}$. Hence procedure *PRO3* does not output anything and does not halt in the first step. Observe that $p{\downarrow}_{R_{P_2\cup Q_2}} = q{\downarrow}_{R_{P_2\cup Q_2}}$. Hence procedure *PRO3* outputs 'yes' and halts in the second step.

Example 11. We continue Example 5. Let $n \geq 1$. We run procedure *PRO3* on the TES E and the ground terms $p = (bd)^{2n}\$$, and $q = (db)^{2n}\$$. We compute as follows. GTES P_1 consists of the equation $bd\$ \approx db\$$. Let Θ stand for $\leftrightarrow^*_{P_1} \cap (sub(P_1) \times sub(P_1))$. Then $sub(P_1)/\Theta = \{\, \{\, b\$\,\}, \{\, d\$\,\}, \{\, bd\$\,\}\,\}$ and $\{\, bd\$\,\}$ is a set of representatives for $sub(P_1)/\leftrightarrow^*_{P_1}$. GTRS R_{P_1} consists of the rule $bd\$ \rightarrow db\$$.

Symmetrically, GTES Q_1 consists of the equation $db\$ \approx bd\$$. GTRS R_{Q_1} consists of the rule $db\$ \rightarrow bd\$$. It is not hard to see, that GTRS $R_{P_1\cup Q_1}$ is equal to GTRS R_{P_1}. Observe that $p{\downarrow}_{R_{P_1\cup Q_1}} = q{\downarrow}_{R_{P_1\cup Q_1}}$, Hence procedure *PRO3* outputs 'yes' and halts in the first step.

We run procedure *PRO3* on the TES E and the ground terms $p = aaa\$$ and $q = bedb\$$. By our arguments in Example 5,

$$p{\downarrow}_{R_{P_i\cup Q_i}} \neq q{\downarrow}_{R_{P_i\cup Q_i}} \quad \text{for } i \geq 1\,.$$

Furthermore, *PRO3* computes as follows.
$R_{Q_1} = \{\, db\$ \rightarrow bd\$, edb\$ \rightarrow de\$\,\}$,
$R_{Q_2} = \{\, db\$ \rightarrow bd\$, edb\$ \rightarrow de\$, bdde\$ \rightarrow dbde\$\,\}$, and
$R_{Q_2} = R_{Q_{n+2}}$ for $n \geq 1$.
Consequently, Procedure *PRO3* outputs 'no' and then halts. Generalizing our arguments, we can show the following.

Statement 1. *Let $p \in T_\Sigma$ be arbitrary such that symbols a or c appear in p. Let $q \in T_\Sigma$ such that a, c do not appear in q. Then procedure PRO3 outputs 'no' and halts on the input E, p, q.*

By Propositon 6, for an arbitrary ground term $q' \in T_\Sigma$, the goal-directed completion procedure [13] may fail or may not halt on the TES E and the goal $(aaa\$, q')$. The following problem is open. For each goal $(aaa\$, q)$ such that $q \in T_\Sigma$, and a, c do not appear in q, is it true that the the goal-directed completion procedure does not fail and halts on the TES E and the goal $(aaa\$, q)$.

It is open whether the goal-directed completion procedure does not fail and halts on the TES E and any goal $(aaa\$, q)$ such that $q \in T_\Sigma$, a, c do not appear in q.

We now show the correctness of Procedure *PRO3*.

Lemma 5. *For any i, n with $1 \leq n \leq i$, and any $t_1, \ldots, t_n \in T_\Sigma$, if $p \leftrightarrow_E t_1 \leftrightarrow_E t_2 \leftrightarrow_E \cdots \leftrightarrow_E t_n$, then $p \leftrightarrow^*_{P_i} t_1 \leftrightarrow^*_{P_i} t_2 \leftrightarrow^*_{P_i} \cdots \leftrightarrow^*_{P_i} t_n$.*

Proof. We proceed by induction on i.
Base Case: $i = 1$. Then $n = 1$. By the definition of P_1, we have $p \leftrightarrow_{P_1} t_1$.

Induction Step: Let $i \geq 1$, and assume that the statement holds for $1, 2, \ldots, i$. We now show that the statement holds for $i+1$. To this end, assume that

$$p \underset{E}{\leftrightarrow} t_1 \underset{E}{\leftrightarrow} t_2 \underset{E}{\leftrightarrow} \cdots \underset{E}{\leftrightarrow} t_n \text{ for some } 0 \leq n \leq i+1\,. \tag{6}$$

By the induction hypothesis,

$$p \overset{*}{\underset{P_i}{\leftrightarrow}} t_1 \overset{*}{\underset{P_i}{\leftrightarrow}} t_2 \overset{*}{\underset{P_i}{\leftrightarrow}} \cdots \overset{*}{\underset{P_i}{\leftrightarrow}} t_{n-1}\,. \tag{7}$$

Hence

$$t_{n-1} \overset{*}{\underset{R_i}{\rightarrow}} p{\downarrow_{R_i}}\,. \tag{8}$$

By (6), there is an equation $l \approx r$ in $E \cup E^{-1}$ with $l, r \in T_\Sigma(X_m)$, $m \geq 0$ and there are $u \in C_\Sigma$, $u_1, \ldots, u_m \in T_\Sigma$ such that

$$t_{n-1} = u[l[u_1, \ldots, u_m]] \text{ and } t_n = u[r[u_1, \ldots, u_m]]\,. \tag{9}$$

As R_i is convergent, by (8) and (9), $u[l[u_1, \ldots, u_m]\downarrow_{R_i}] \rightarrow^*_{R_i} p\downarrow_{R_i}$. That is, R_i reaches $p\downarrow_{R_i}$ starting from $l[u_1, \ldots, u_m]\downarrow_{R_i}$ attached to some context. By the definition of P_{i+1},

$$l[u_1, \ldots, u_m] \approx r[u_1, \ldots, u_m] \text{ is in } \overset{*}{\underset{P_i}{\leftrightarrow}} \text{ or } P_{i+1}\,. \tag{10}$$

By Lemma 3, (7), (9), and (10),

$$p \overset{*}{\underset{P_{i+1}}{\leftrightarrow}} t_1 \overset{*}{\underset{P_{i+1}}{\leftrightarrow}} t_2 \overset{*}{\underset{P_{i+1}}{\leftrightarrow}} \cdots \overset{*}{\underset{P_{i+1}}{\leftrightarrow}} t_{n-1} \overset{*}{\underset{P_{i+1}}{\leftrightarrow}} t_n\,.$$

□

By Lemma 3 and Lemma 5 we have the following result.

Lemma 6. *Assume that $R_i = P_{i+1}$ for some $i \geq 1$. Then $p \leftrightarrow^*_{P_{i+1}} q$ if and only if $p \leftrightarrow^*_E q$.*

Theorem 1. *If $p \leftrightarrow^*_E q$, then procedure PRO3 outputs 'yes' and halts.*

Proof. Assume that $p = t_1 \leftrightarrow_E t_2 \leftrightarrow_E \cdots \leftrightarrow_E t_n = q$ for some $n \geq 1$ and $t_1, \ldots, t_n \in T_\Sigma$. By Lemma 5, $p \leftrightarrow^*_{P_n} q$. Let k be the least integer such that $p \leftrightarrow^*_{P_k \cup Q_k} q$.

First assume that $k = 1$. Then $p \leftrightarrow^*_{P_1 \cup Q_1} q$. Hence $p\downarrow_{R_{P_1 \cup Q_1}} = q\downarrow_{R_{P_1 \cup Q_1}}$. Consequently, procedure *PRO3* outputs 'yes' and halts in the first step.

Second assume that $k \geq 2$. Then by the definition of k, $(p, q) \notin \leftrightarrow^*_{P_i \cup Q_i}$ for $2 \leq i \leq k-1$. Then by Lemma 6, $R_{P_{i-1}} \subset P_i$ and $R_{Q_{i-1}} \subset Q_i$ for $2 \leq i \leq k-1$. Hence procedure *PRO3* does not halt in the first $k-1$ steps. By the definition of the integer k, in the kth step procedure *PRO3* outputs 'yes' and halts.

□

Theorem 2. *If procedure PRO3 outputs 'yes' and halts, then* $p \leftrightarrow^*_E q$. *If procedure PRO3 outputs 'no' and halts, then* $(p,q) \notin \leftrightarrow^*_E$.

Proof. Assume that procedure *PRO3* outputs 'yes' and halts in the kth step. Then $p \leftrightarrow^*_{P_k \cup Q_k} q$. By Lemma 3, $p \leftrightarrow^*_E q$.

Assume that procedure *PRO3* outputs 'no' and halts in the kth step. Then

(a) $(p,q) \notin \leftrightarrow^*_{P_k \cup Q_k}$ and

(b) $P_k = R_{P_{k-1}}$ or $Q_k = R_{Q_{k-1}}$.

We now distinguish two cases.

Case 1: $P_k = R_{P_{k-1}}$. By (a) and by Lemma 6, $(p,q) \notin \leftrightarrow^*_E$.

Case 2: $Q_k = R_{Q_{k-1}}$. This case is symmetric to Case 2.

□

Theorems 1 and 2 imply the following.

Theorem 3. *If* $p \leftrightarrow^*_E q$*, then procedure PRO3 outputs 'yes' and halts. Otherwise, either PRO3 outputs 'no' and halts, or PRO3 does not halt.*

Example 12. We continue Example 3. We now run procedure *PRO3* on the TES E and the ground terms p, q. Then $P_1 = Q_1 = \{\# \approx \$\}$, $R_{P_1} = R_{Q_1} = P_1$, and $R_{P_1 \cup Q_1} = P_1$. Observe that $p{\downarrow}_{R_{P_1 \cup Q_1}} = q{\downarrow}_{R_{P_1 \cup Q_1}}$. Hence procedure *PRO3* outputs 'yes' and halts in the first step. By Proposition 2, we compute $p{\downarrow}_{R_{P_1 \cup Q_1}}$ and $q{\downarrow}_{R_{P_1 \cup Q_1}}$ in linear time. We apply the rules of $R_{P_1 \cup Q_1}$ n times. For this example, *PRO3* is faster than *PRO1*.

Example 13. We continue Example 6. We now run procedure *PRO3* on the TES E and the ground terms p and q. Then $\{a \approx ffa\} = P_1 = R_{P_1} = P_{1+i} = R_{P_{1+i}}$ for $i \geq 1$. Furthermore, $Q_1 = \{a \approx ffa,\ fa \approx fffa\}$, $R_{Q_1} = P_1 = Q_2 = R_{Q_2} = Q_{1+i} = R_{Q_{1+i}}$ for $i \geq 1$.

Observe that $p{\downarrow}_{R_{P_2 \cup Q_2}} \neq q{\downarrow}_{R_{P_2 \cup Q_2}}$. Hence procedure *PRO3* outputs 'no' and halts in the second step.

It should be clear that for all ground terms p and q, *PRO3* halts. It outputs 'yes' if $p \leftrightarrow^*_E q$. Otherwise it outputs 'no'.

Example 14. We now continue Example 2. We apply procedure *PRO3* to the TES $E = \{ffx \approx gfx\}$ and any terms $p, q \in T_\Sigma$. Observe that $height(ffx) = 2 = height(gfx)$.

Statement 2. *For each* $i \geq 0$*, and for each pair of terms,* $s,t \in T_\Sigma(X)$*, if* $(s,t) \in P_i$*, then* $height(s) = height(t) \leq height(p)$.

Proof. We proceed by induction on n.

Base Case: $i = 1$. By the definition of P_1, for each equation $s \approx t$ in P_1, $height(s) = height(t) \leq height(p)$. Hence our statement holds.

Induction Step: Let $n \geq 1$, and assume that the satement holds for $1, 2, \ldots, n$. We now show that the satement holds for $n+1$. Consider an equation

$l[u_1, \ldots, u_m] \approx r[u_1, \ldots, u_m]$ in P_{i+1}. Then there exist

- an equation $l \approx r$ of $E \cup E^{-1}$, where $l, r \in T_\Sigma(X_m)$, $m \geq 0$.

- $u_1, \ldots, u_m \in (sub(R_i) - lhs(R_i)) \cup sub(p{\downarrow}_{R_i})$.

such that R_i reaches $p{\downarrow}_{R_i}$ starting from $l[u_1, \ldots, u_m]{\downarrow}_{R_i}$ attached to some context, and that
$l[u_1, \ldots, u_m]{\downarrow}_{R_i} \neq r[u_1, \ldots, u_m]{\downarrow}_{R_i}$.
Consequently, there is a $u \in C_\Sigma$ such that $u[l[u_1, \ldots, u_m]] \rightarrow^*_{R_i} p$. By (a) in Lemma 3 and the induction hypothesis, $height(u[l[u_1, \ldots, u_m]]) = height(p)$. Thus $height(l[u_1, \ldots, u_m]) \leq height(p)$. By (a) in Lemma 3 and the induction hypothesis, $height(l) = height(r)$. Hence $height(l[u_1, \ldots, u_m]) = height(r[u_1, \ldots, u_m])$.
□

Observe that the set $\{ (s,t) \in T_\Sigma \times T_\Sigma \mid height(s) = height(t) \leq height(p) \}$ is finite. By Lemma 3 and Statement 2, procedure *PRO3* halts on E and any terms $p, q \in T_\Sigma$ in finitely many steps.

The following result can be shown by generalizing the proof appearing in Example 14.

Theorem 4. *Let E be a variable preserving TES such that*

- *for any equation $s \approx t$ in E, $height(s) = height(t)$, or*
- *for any equation $s \approx t$ in E, $size(s) = size(t)$ and each variable appears the same times in s and t.*

Let $p, q \in T_\Sigma$ be arbitrary. Then procedure PRO3 halts on E and terms p, q.

6 Semi-decision procedure for the ground word problem of TESs

We present the semi-decision procedure *PRO4* for the ground word problem of TESs, and show its correctness. We obtain it generalizing *PRO3* taking into account *PRO2*. The starting point to the definition of the GTESs P_i, $i \geq 1$, is the same as in Section 5. We define P_1 as the set of all instances $l' \rightarrow r'$ of equations $l \approx r$ in $E \cup E^{-1}$ which can be applied to p. We define P_{i+1}, $i \geq 1$, as the set of all instances $l' \rightarrow r'$ of equations $l \approx r$ in $E \cup E^{-1}$ which can be applied to elements of $\{ s \in T_\Sigma \mid p \leftrightarrow^*_{P_i} s \}$. The question is what should we substitute for those variables in the right-hand side r that do not appear in the left-hand side l. We now give a simplified answer to this question. Applying Snyder's algorithm we compute a reduced GTRS R_i equivalent to the GTES P_i. When constructing the instance $l' \rightarrow r'$ of $l \approx r$, we substitute any term in $(sub(R_i) - lhs(R_i)) \cup sub(p{\downarrow}_{R_i})$ or the R_i normal form of any ground term of height less than or equal to i for each variable in the right-hand side r that does not appear on the left-hand side l. Furthermore, we define the GTESs Q_i, $i \geq 1$, symmetrically.

Let E be a TES over Σ, and let $p, q \in T_\Sigma$. We now define the GTESs P_i and the reduced GTRSs R_i, $i \geq 1$, over Σ.

Let $NORM_0 = \Sigma_0 \cup sub(p)$. For each equation $l \approx r$ of $E \cup E^{-1}$ with $l \in T_\Sigma(X_{k+m})$, $r \in T_\Sigma(X_k \cup X_{[k+m+1,k+m+\ell]})$ for some $k, m, \ell \geq 0$, if $p = u[l[u_1, \ldots, u_{k+m}]]$ for some $u \in C_\Sigma$, $u_1, \ldots, u_{k+m} \in T_\Sigma$, then for all

$v_{k+m+1}, \ldots, v_{k+m+\ell} \in NORM_0$, we put the equation

$$l[u_1, \ldots, u_{k+m}] \approx r[u_1, \ldots, u_k, v_{k+m+1}, \ldots, v_{k+m+\ell}]$$

in P_1. Applying Snyder's algorithm we compute a reduced GTRS R_1 equivalent to the GTES P_1, see Proposition 3.

Let $i \geq 1$. Let
$NORM_i = sub(p{\downarrow_{R_i}}) \cup (sub(R_i) - lhs(R_i)) \cup$
$\{\, t{\downarrow_{R_i}} \mid t \in NORM_{i-1}$ or $t = f(t_1, \ldots, t_m)$ for some $f \in \Sigma_m$ and $t_1, \ldots, t_m \in NORM_{i-1} \,\}$.

(a) We put each rule of R_i into P_{i+1}.

(b) For each equation $l \approx r$ of $E \cup E^{-1}$ with $l \in T_\Sigma(X_{k+m})$, $r \in T_\Sigma(X_k \cup X_{[k+m+1,k+m+\ell]})$ for some $k, m, \ell \geq 0$, for any $u_1, \ldots, u_{k+m} \in (sub(R_i) - lhs(R_i)) \cup sub(p{\downarrow_{R_i}})$ and $v_{k+m+1}, \ldots, v_{k+m+\ell} \in NORM_i$, if R_i reaches $p{\downarrow_{R_i}}$ starting from $l[u_1, \ldots, u_{k+m}]{\downarrow_{R_i}}$ attached to some context, and
$l[u_1, \ldots, u_{k+m}]{\downarrow_{R_i}} \neq r[u_1, \ldots, u_m, v_{k+m+1}, \ldots, v_{k+m+\ell}]{\downarrow_{R_i}}$,
then we put the equation

$$l[u_1, \ldots, u_{k+m}] \approx r[u_1, \ldots, u_m, v_{k+m+1}, \ldots, v_{k+m+\ell}]$$

in P_{i+1}.

If we do not put equations in P_{i+1} in item (b), i.e. $P_{i+1} = R_i$, then let $R_{i+1} = R_i$. Otherwise, applying Snyder's algorithm, we compute a reduced GTRS R_{i+1} equivalent to the GTES P_{i+1}.

When misunderstanding may arise, we denote R_i as R_{P_i}. We define the GTESs Q_i, $i \geq 1$, symmetrically to the GTESs P_i, $i \geq 1$. Applying Snyder's algorithm, we compute a reduced GTRS $R_{P_i \cup Q_i}$ equivalent to the GTRS $R_{P_i} \cup R_{Q_i}$ for $i \geq 1$.

By Proposition 1 GTRSs R_{P_i}, R_{Q_i}, and $R_{P_i \cup Q_i}$ are convergent.

We illustrate our concepts and results by two running examples, each of them is presented as a series of examples.

Example 15. We continue Example 7. Let $p = f(0,1)$ and $q = f(f(0,1),1)$. Observe that for any $u, v \in T_\Sigma$, if $u \leftrightarrow_E^* v$, then the parity of the number of 1's in u equals to that in v. Hence

$$(p, q) \notin \overset{*}{\underset{E}{\leftrightarrow}} . \tag{11}$$

We now construct the GTESs P_1, P_2, and P_3. Then $NORM_0 = \{\, 0, 1, f(0,1) \,\}$. P_1 consists of the equations
$0 \approx f(0,0)$, $0 \approx f(1,1)$, $0 \approx f(f(0,1), f(0,1))$,
$1 \approx f(0,1)$, $f(0,1) \approx 1$, $f(0,1) \approx f(0, f(0,1))$.
R_1 consists of the rules
$f(0,0) \to 0$, $f(1,1) \to 0$, $f(0,1) \to 1$.
$NORM_1 = \{\, 0, 1, f(1,0) \,\}$. P_2 consists of the equations
$f(0,0) \approx 0$, $f(1,1) \approx 0$, $f(0,1) \approx 1$, $0 \approx f(f(1,0), f(1,0))$.
R_2 consists of the rules
$f(0,0) \to 0$, $f(1,1) \to 0$, $f(0,1) \to 1$, $f(f(1,0), f(1,0)) \to 0$.

$NORM_2 = \{\, 0,\ 1,\ f(1,0),\ f(0,f(1,0)),\ f(1,f(1,0)),\ f(f(1,0),0),\ f(f(1,0),1) \,\}$.
P_3 consists of the equations
$f(0,0) \approx 0, \quad f(1,1) \approx 0, \quad f(0,1) \approx 1, \quad f(f(1,0),f(1,0)) \approx 0,$
$0 \approx f(f(0,f(1,0)),f(0,f(1,0))),$
$0 \approx f(f(1,f(1,0)),f(1,f(1,0))),$
$0 \approx f(f(f(1,0),0),f(f(1,0),0)),$
$0 \approx f(f(f(1,0),1),f(f(1,0),1)).$
R_3 consists of the rules
$f(0,0) \to 0, \quad f(1,1) \to 0, \quad f(0,1) \to 1, \quad f(f(1,0),f(1,0)) \to 0,$
$f(f(0,f(1,0)),f(0,f(1,0))) \to 0,$
$f(f(1,f(1,0)),f(1,f(1,0))) \to 0,$
$f(f(f(1,0),0),f(f(1,0),0)) \to 0,$
$f(f(f(1,0),1),f(f(1,0),1)) \to 0.$
Continuing in this manner we get that

$$R_{P_i} \subset R_{P_{i+1}} \text{ for } i \geq 1\,. \tag{12}$$

We now compute the GTESs Q_1, Q_2, and Q_3.
$NORM_0 = \{\, 0,1,f(0,1),f(f(0,1),1) \,\}$.
Q_1 consists of the equations
$0 \approx f(0,0), \quad 0 \approx f(1,1), \quad 0 \approx f(f(0,1),f(0,1)),$
$0 \approx f(f(f(0,1),1),f(f(0,1),1)),$
$1 \approx f(0,1), \quad f(0,1) \approx f(0,f(0,1)), \quad f(f(0,1),1) \approx f(0,f(f(0,1),1)).$
R_{Q_1} consists of the rules
$f(0,0) \to 0, \quad f(1,1) \to 0, \quad f(0,1) \to 1.$
$NORM_1 = \{\, 0,1,f(1,0) \,\}$.
Q_2 consists of the equations
$f(0,0) \approx 0, \quad f(1,1) \approx 0, \quad f(0,1) \approx 1, \quad 0 \approx f(f(1,0),f(1,0)).$
R_{Q_2} consists of the rules
$f(0,0) \to 0, \quad f(1,1) \to 0, \quad f(0,1) \to 1, \quad f(f(1,0),f(1,0)) \to 0.$
$NORM_2 = \{\, 0,1,f(1,0),f(0,f(1,0)),f(1,f(1,0)),f(f(1,0),0),f(f(1,0),1) \,\}$.
Q_3 consists of the equations
$f(0,0) \approx 0, \quad f(1,1) \approx 0, \quad f(0,1) \approx 1, \quad 0 \approx f(f(1,0),f(1,0)),$
$0 \approx f(f(0,f(1,0)),f(0,f(1,0))), \quad 0 \approx f(f(1,f(1,0)),f(1,f(1,0))),$
$0 \approx f(f(f(1,0),0),f(f(1,0),0)), \quad 0 \approx f(f(f(1,0),1),f(f(1,0),1)).$
R_{Q_3} consists of the rules
$f(0,0) \to 0, \quad f(1,1) \to 0, \quad f(0,1) \to 1, \quad f(f(1,0),f(1,0)) \to 0,$
$f(f(0,f(1,0)),f(0,f(1,0))) \to 0, \quad f(f(1,f(1,0)),f(1,f(1,0))) \to 0,$
$f(f(f(1,0),0),f(f(1,0),0)) \to 0, \quad f(f(f(1,0),1),f(f(1,0),1)) \to 0.$
Continuing in this manner we get that

$$R_{Q_i} \subset R_{Q_{i+1}} \text{ for } i \geq 1\,. \tag{13}$$

Let $R_{P_1 \cup Q_1} = R_{P_1}$, $R_{P_2 \cup Q_2} = R_{P_2}$, and $R_{P_3 \cup Q_3} = R_{P_3} \cup R_{Q_3}$.

Example 16. Let $\Sigma = \Sigma_0 \cup \Sigma_1$, $\Sigma_0 = \{0, 1\}$, and $\Sigma_1 = \{g, h\}$. Let TES E consist of the equations

$gx_1 \approx x_1, \quad hx_1 \approx hx_2.$

Let $p = 0$ and $q = 1$.

We now construct the GTESs P_1, P_2, and P_3. Then $NORM_0 = \{0, 1\}$. P_1 consists of the equation $0 \approx g0$.

R_1 consists of the rule $g0 \to 0$.

$NORM_1 = \{0, 1, g1, h0, h1\}$.

$P_2 = R_1$ and $R_2 = P_2$.

$NORM_2 = \{0, 1, g1, h0, h1, gg1, hg1, gh0, hh0, gh1, hh1\}$.

$P_3 = R_2$ and $R_3 = P_3$.

We now construct the GTESs Q_1, Q_2, and Q_3. Then $NORM_0 = \{0, 1\}$. Q_1 consists of the equation $1 \approx g1$.

R_{Q_1} consists of the rule $g1 \to 1$.

$NORM_1 = \{0, 1, g0, h0, h1\}$.

$Q_2 = R_{Q_1}$ and $R_{Q_2} = Q_2$.

$NORM_2 = \{0, 1, g0, h0, h1, gg0, hg0, gh0, hh0, gh1, hh1\}$.

$Q_3 = R_{Q_2}$ and $R_{Q_3} = Q_3$.

$R_{P_1} \cup R_{Q_1} = R_{P_1 \cup Q_1} = R_{P_2 \cup Q_2} = R_{P_3 \cup Q_3}$.

We get the following result by direct inspection of the definition of the GTES P_i and GTRS R_i, $i \geq 1$.

Statement 3. *For each $i \geq 1$, $\leftrightarrow^*_{P_i} \subseteq \leftrightarrow^*_{P_{i+1}} \subseteq \leftrightarrow^*_E$.*

We can show the following result similarly to Lemma 4.

Lemma 7. *For each $i \geq 1$, we can effectively construct the GTES P_i.*

Lemma 8. *For each $i \geq 1$, $sub(p\downarrow_{R_{P_i}}) \cup (sub(R_{P_i}) - lhs(R_{P_i})) \cup \{ t\downarrow_{R_{P_i}} \mid height(t) \leq i \} \subseteq NORM_i$.*

Proof. By induction on i.

□

We now present our semi-decision procedure.

Procedure *PRO4 Input:* A variable preserving TES E over the ranked alphabet Σ and ground terms $p, q \in T_\Sigma$.

Output: • 'yes' if $p \leftrightarrow^*_E q$,

• 'no' if $(p, q) \notin \leftrightarrow^*_E$ and the procedure halts,

• undefined if the procedure does not halt.

compute P_1, R_{P_1}, Q_1, R_{Q_1}, and $R_{P_1 \cup Q_1}$;

if $p\downarrow_{R_{P_1 \cup Q_1}} = q\downarrow_{R_{P_1 \cup Q_1}}$, then begin output 'yes'; halt end;

$i := 1$;

1: $i := i + 1$;

compute P_i, R_{P_i}, Q_i, R_{Q_i}, and $R_{P_i \cup Q_i}$;

if $p\downarrow_{R_{P_i \cup Q_i}} = q\downarrow_{R_{P_i \cup Q_i}}$, then begin output 'yes'; halt end;

if $i = 2$, then goto 1;
if $R_{P_{i-2}} = R_{P_{i-1}} = P_i$, or $R_{Q_{i-2}} = R_{Q_{i-1}} = Q_i$,
then begin output 'no'; halt end;
goto 1

Example 17. We continue Example 15. By Statement 3 and (11), $p{\downarrow}_{R_{P_i \cup Q_i}} \neq q{\downarrow}_{R_{P_i \cup Q_i}}$ for $i \geq 1$. Hence procedure *PRO4* does not output 'yes'. By (12) and (13), procedure *PRO4* does not output 'no'. Hence procedure *PRO4* does not output anything and does not halt at all.

Example 18. We continue Example 16. Observe that
$p{\downarrow}_{R_{P_1 \cup Q_1}} = 0 \neq 1 = q{\downarrow}_{R_{P_1 \cup Q_1}}$,
$p{\downarrow}_{R_{P_2 \cup Q_2}} = 0 \neq 1 = q{\downarrow}_{R_{P_2 \cup Q_2}}$,
$p{\downarrow}_{R_{P_3 \cup Q_3}} = 0 \neq 1 = q{\downarrow}_{R_{P_3 \cup Q_3}}$, and
$R_{P_1} = R_{P_2} = P_3$.
Hence procedure *PRO4* outputs 'no' and halts in the third step.

Example 19. Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, $\Sigma_0 = \{\,\$, \#\,\}$, $\Sigma_1 = \{\,f, g\,\}$, $\Sigma_2 = \{\,h\,\}$. Consider the TES $E = \{\, ffx_1 \approx gfx_1,\ h(x_1, x_1) \approx \$\,\}$. As in Example 2, we can show that the basic Knuth-Bendix completion procedure runs forever on this example. Moreover, it is still open whether the goal-directed completion procedure halts on the TES E and any goal.

Let $n \geq 1$. Let $p = h(f^n\$, gf^{n-1}\$)$ and $q = \$$. We raise the problem whether $p \leftrightarrow_E^* q$. We now apply procedure *PRO4* to the TES E and the terms p, q.
GTRS R_{P_1} consists of the rules
$f^i\$ \to gf^{i-1}\$$ for $2 \leq i \leq n$,
$h(\$, \$) \to \$$,
$h(\#, \#) \to \$$.
GTRS R_{Q_1} consists of the rules
$h(\$, \$) \to \$$,
$h(\#, \#) \to \$$.
GTRS R_{P_2} consists of the rules
$f^2gf\$ \to gf\$$,
$h(\$, \$) \to \$$,
$h(\#, \#) \to \$$.
$h(f\$, f\$) \to \$$,
$h(f\#, f\#) \to \$$.
$h(g\$, g\$) \to \$$,
$h(g\#, g\#) \to \$$.
GTRS R_{Q_2} consists of the rules
$h(\$, \$) \to \$$,
$h(\#, \#) \to \$$,
$h(f\$, f\$) \to \$$,
$h(f\#, f\#) \to \$$.
$h(g\$, g\$) \to \$$,
$h(g\#, g\#) \to \$$.

Clearly,
$p{\downarrow_{R_{P_2\cup Q_2}}} = q{\downarrow_{R_{P_2\cup Q_2}}}$.
Hence procedure *PRO4* outputs 'yes' and halts in the second step.

We now show the correctness of Procedure *PRO4*.

Lemma 9. *Assume that* $R_{i-1} = R_i = P_{i+1}$ *and* $NORM_{i-1} \subset NORM_i$ *for some* $i \geq 2$. *Then for each equation* $l \approx r$ *of* $E \cup E^{-1}$ *with* $l \in T_\Sigma(X_{k+m})$, $r \in T_\Sigma(X_k \cup X_{[k+m+1,k+m+\ell]})$, $k, m \geq 0$, $\ell \geq 1$, *and for any* $u_1, \ldots, u_{k+m} \in sub(p{\downarrow_{R_i}}) \cup (sub(R_i) - lhs(R_i))$, R_i *does not reach* $p{\downarrow_{R_i}}$ *starting from* $l[u_1, \ldots, u_{k+m}]{\downarrow_{R_i}}$ *attached to some context.*

Proof. By contradiction. Assume that there is an equation $l \approx r$ of $E \cup E^{-1}$ with $l \in T_\Sigma(X_{k+m})$, $r \in T_\Sigma(X_k \cup X_{[k+m+1,k+m+\ell]})$, $k, m \geq 0$, $\ell \geq 1$, and there are $u_1, \ldots, u_{k+m} \in sub(p{\downarrow_{R_i}}) \cup (sub(R_i) - lhs(R_i))$ such that R_i reaches $p{\downarrow_{R_i}}$ starting from $l[u_1, \ldots, u_{k+m}]{\downarrow_{R_i}}$ attached to some context. By $R_i = P_{i+1}$, we do not put equations in P_{i+1} in item (b) of its definition. Consequently, for any $v_{k+m+1}, \ldots, v_{k+m+\ell} \in NORM_i$,

$$l[u_1, \ldots, u_{k+m}]{\downarrow_{R_i}} = r[u_1, \ldots, u_m, v_{k+m+1}, \ldots, v_{k+m+\ell}]{\downarrow_{R_i}} \ .$$

Hence by our indirect assumption, R_i reaches $p{\downarrow_{R_i}}$ starting from $r[u_1, \ldots, u_m, v_{k+m+1}, \ldots, v_{k+m+\ell}]{\downarrow_{R_i}}$ attached to some context. Hence there is a $u \in C_\Sigma$ such that

$$u[r[u_1, \ldots, u_m, v_{k+m+1}, \ldots, v_{k+m+\ell}]{\downarrow_{R_i}}] \xrightarrow[R]{*} p{\downarrow_{R_i}} \ .$$

Then $u[r[u_1, \ldots, u_m, v_{k+m+1}{\downarrow_{R_i}}, v_{k+m+2}, \ldots, v_{k+m+\ell}]] \to^*_{R_i}$
$u[r[u_1, \ldots, u_m, v_{k+m+1}, \ldots, v_{k+m+\ell}]{\downarrow_{R_i}}] \to^*_{R_i} p{\downarrow_{R_i}}$. By Lemma 2, $v_{k+m+1}{\downarrow_{R_i}} \in sub(p{\downarrow_{R_i}}) \cup (sub(R_i) - lhs(R_i))$. Since $R_{i-1} = R_i$,
$v_{k+m+1}{\downarrow_{R_{i-1}}} \in sub(p{\downarrow_{R_{i-1}}}) \cup (sub(R_{i-1}) - lhs(R_{i-1})) \subseteq NORM_{i-1}$.
By definition, v_{k+m+1} is an arbitrary element of $NORM_i$. Consequently, we have $NORM_i \subseteq NORM_{i-1}$. This is a contradiction.

□

Lemma 10. *Let* $i \geq 2$. *If* $R_{i-1} = R_i = R_{i+1}$ *and* $NORM_{i-1} = NORM_i$, *then* $NORM_i = NORM_{i+1}$.

Proof. First we show that $NORM_i \subseteq NORM_{i+1}$. Let $s \in NORM_i$ be arbitrary. If $s \in sub(p{\downarrow_{R_i}}) \cup (sub(R_i) - lhs(R_i)) \cup \{\, t{\downarrow_{R_i}} \mid t \in NORM_{i-1} \,\}$, then $s \in sub(p{\downarrow_{R_{i+1}}}) \cup (sub(R_{i+1}) - lhs(R_{i+1})) \cup \{\, t{\downarrow_{R_{i+1}}} \mid t \in NORM_i \,\}$. Hence $t \in NORM_{i+1}$. If $s = f(t_1, \ldots, t_m){\downarrow_{R_i}}$ for some $f \in \Sigma_m$ and $t_1, \ldots, t_m \in NORM_{i-1}$, then $s = f(t_1, \ldots, t_m){\downarrow_{R_{i+1}}}$ with $f \in \Sigma_m$ and $t_1, \ldots, t_m \in NORM_i$. Hence $t \in NORM_{i+1}$.

We now show that $NORM_{i+1} \subseteq NORM_i$. Let $s \in NORM_{i+1}$ be arbitrary. If $s \in sub(p{\downarrow_{R_{i+1}}}) \cup (sub(R_{i+1}) - lhs(R_{i+1})) \cup \{\, t{\downarrow_{R_{i+1}}} \mid t \in NORM_i \,\}$, then $s \in sub(p{\downarrow_{R_i}}) \cup (sub(R_i) - lhs(R_i)) \cup \{\, t{\downarrow_{R_i}} \mid t \in NORM_{i-1} \,\}$. Hence $t \in NORM_i$.

If $s = f(t_1, \ldots, t_m)\downarrow_{R_{i+1}}$ for some $f \in \Sigma_m$ and $t_1, \ldots, t_m \in NORM_i$, then $s = f(t_1, \ldots, t_m)\downarrow_{R_i}$ for $f \in \Sigma_m$ and $t_1, \ldots, t_m \in NORM_{i-1}$. Hence $t \in NORM_i$.

□

Lemma 11. *For each $i \geq 2$, if $R_{i-1} = R_i = P_{i+1}$, then $R_i = R_{i+1} = P_{i+2}$.*

Proof. By the assumption $R_i = P_{i+1}$ and the definition of R_{i+1}, we have

$$R_i = R_{i+1}\,. \tag{14}$$

We now distinguish two cases.

Case 1: $NORM_{i-1} = NORM_i$. By Lemma 10,

$$NORM_i = NORM_{i+1}\,. \tag{15}$$

By (14) and (15), $P_{i+1} = P_{i+2}$. By the assumption $R_i = P_{i+1}$ and (14), we have $R_i = R_{i+1} = P_{i+2}$.

Case 2: $NORM_{i-1} \subset NORM_i$. Then by Lemma 9, for each equation $l \approx r$ of $E \cup E^{-1}$ with $l \in T_\Sigma(X_{k+m})$, $r \in T_\Sigma(X_k \cup X_{[k+m+1,k+m+\ell]})$, $k, m \geq 0$, $\ell \geq 1$, and for any $u_1, \ldots, u_{k+m} \in (sub(R_i) - lhs(R_i)) \cup sub(p\downarrow_{R_i})$, R_i does not reach $p\downarrow_{R_i}$ starting from $l[u_1, \ldots, u_{k+m}]\downarrow_{R_i}$ attached to some context. Then by (14), we do not put equations in P_{i+2} in item (b) in the definition of P_{i+2}. Hence $R_{i+1} = P_{i+2}$. By (14) the proof is complete.

□

Lemma 11 implies the following.

Lemma 12. *For each $i \geq 1$, if $R_{i-1} = R_i = P_{i+1}$, then for each $k \geq 1$, $R_i = R_{i+k} = P_{i+k+1}$.*

We now show the correctness of Procedure *PRO4*.

Lemma 13. *For any $n \geq 1$, $t_1, \ldots, t_n \in T_\Sigma$, if $p \leftrightarrow_E t_1 \leftrightarrow_E t_2 \leftrightarrow_E \cdots \leftrightarrow_E t_n$, then there is $i \geq 1$ such that $p \leftrightarrow^*_{P_i} t_1 \leftrightarrow^*_{P_i} t_2 \leftrightarrow^*_{P_i} \cdots \leftrightarrow^*_{P_i} t_n$.*

Proof. We proceed by induction on n.

Base Case: $n = 1$. Assume that $p \leftrightarrow_E t_1$. Then there is an equation $l \approx r$ of $E \cup E^{-1}$ with $l \in T_\Sigma(X_{k+m})$, $r \in T_\Sigma(X_{k+m+\ell})$, $k, m, \ell \geq 0$, and there is $u \in C_\Sigma$, $u_1, \ldots, u_{k+m}, v_{k+m+1}, \ldots, v_{k+m+\ell} \in T_\Sigma$ such that

$$p = u[l[u_1, \ldots, u_{k+m}]] \tag{16}$$

and $t_1 = u[r[u_1, \ldots, u_k, v_{k+m+1}, \ldots, v_{k+m+\ell}]]$.
Let $i = max\{\, height(v_{k+1}), \ldots, height(v_{k+m+\ell})\,\}$. By Lemma 8, $v_{k+m+1}\downarrow_{R_i}, \ldots, v_{k+m+\ell}\downarrow_{R_i}$ are in $NORM_i$. By (16), R_i reaches $p\downarrow_{R_i}$ from $l[u_1\downarrow_{R_i}, \ldots, u_{k+m}\downarrow_{R_i}]\downarrow_{R_i}$ attached to some context. By the definition of P_{i+1}, the equation

$$l[u_1\downarrow_{R_i}, \ldots, u_{k+m}\downarrow_{R_i}] \approx r[u_1\downarrow_{R_i}, \ldots, u_k\downarrow_{R_i}, v_{k+m+1}\downarrow_{R_i}, \ldots, v_{k+m+\ell}\downarrow_{R_i}]$$

is in $\leftrightarrow^*_{P_i}$ or P_{i+1}. Hence, by the definition of R_i and Statement 3,

$p = u[l[u_1, \ldots, u_{k+m}]] \leftrightarrow^*_{P_{i+1}} u[l[u_1{\downarrow}_{R_i}, \ldots, u_{k+m}{\downarrow}_{R_i}]] \leftrightarrow^*_{P_{i+1}}$
$u[r[u_1{\downarrow}_{R_i}, \ldots, u_k{\downarrow}_{R_i}, v_{k+m+1}{\downarrow}_{R_i}, \ldots, v_{k+m+\ell}]{\downarrow}_{R_i}] \leftrightarrow^*_{P_i}$
$u[r[u_1, \ldots, u_k, v_{k+m+1}, \ldots, v_{k+m+\ell}]] = t_1$.
Then we have $p \leftrightarrow^*_{P_{i+1}} t_1$.

Induction Step: Let $n \geq 1$, and assume that the satement holds for $1, 2, \ldots, n$. We now show that the satement holds for $n + 1$. To this end, assume that

$$p \underset{E}{\leftrightarrow} t_1 \underset{E}{\leftrightarrow} t_2 \underset{E}{\leftrightarrow} \cdots \underset{E}{\leftrightarrow} t_{n+1}\,. \tag{17}$$

By the induction hypothesis, there is $j \geq 1$ such that

$$p \overset{*}{\underset{P_j}{\leftrightarrow}} t_1 \overset{*}{\underset{P_j}{\leftrightarrow}} t_2 \overset{*}{\underset{P_j}{\leftrightarrow}} \cdots \overset{*}{\underset{P_j}{\leftrightarrow}} t_n\,. \tag{18}$$

Hence

$$t_n \overset{*}{\underset{R_i}{\rightarrow}} p{\downarrow}_{R_i}\,. \tag{19}$$

By (17), there is an equation $l \approx r$ in $E \cup E^{-1}$ with $l \in T_\Sigma(X_{k+m})$, $r \in T_\Sigma(X_k \cup X_{[k+m+1,k+m+\ell]})$ for some $k, m, \ell \geq 0$, and there are $u \in C_\Sigma$, $u_1, \ldots, u_{k+m}, v_{k+m+1}, \ldots, v_{k+m+\ell} \in T_\Sigma$ such that

$$t_n = u[l[u_1, \ldots, u_{k+m}]] \text{ and } t_{n+1} = u[r[u_1, \ldots, u_k, v_{k+m+1}, \ldots, v_{k+m+\ell}]]\,. \tag{20}$$

Let $i = max\{\, j, height(v_{k+m+1}), \ldots, height(v_{k+m+\ell})\,\}$. By Lemma 8, $v_{k+m+1}{\downarrow}_{R_i}, \ldots, v_{k+m+\ell}{\downarrow}_{R_i}$ are in $NORM_i$. Clearly, $l[u_1{\downarrow}_{R_i}, \ldots, u_{k+m}{\downarrow}_{R_i}] \rightarrow^*_{R_i} l[u_1{\downarrow}_{R_i}, \ldots, u_{k+m}{\downarrow}_{R_i}]{\downarrow}_{R_i}$. Then by (19) and (20), R_i reaches $p{\downarrow}_{R_i}$ starting from $l[u_1{\downarrow}_{R_i}, \ldots, u_{k+m}{\downarrow}_{R_i}]{\downarrow}_{R_i}$ attached to some context. By the definition of P_{i+1}, the equation
$l[u_1{\downarrow}_{R_i}, \ldots, u_{k+m}{\downarrow}_{R_i}] \approx r[u_1{\downarrow}_{R_i}, \ldots, u_k{\downarrow}_{R_i}, v_{k+m+1}{\downarrow}_{R_i}, \ldots, v_{k+m+\ell}{\downarrow}_{R_i}]$
is in $\leftrightarrow^*_{P_i}$ or P_{i+1}. Hence, by the definition of R_i and Statement 3,
$t_n = u[l[u_1, \ldots, u_{k+m}]] \leftrightarrow^*_{P_{i+1}} u[l[u_1{\downarrow}_{R_i}, \ldots, u_{k+m}{\downarrow}_{R_i}]] \leftrightarrow_{P_{i+1}}$
$u[r[u_1{\downarrow}_{R_i}, \ldots, u_k{\downarrow}_{R_i}, v_{k+m+1}{\downarrow}_{R_i}, \ldots, v_{k+m+\ell}{\downarrow}_{R_i}]] \leftrightarrow^*_{P_{i+1}}$
$u[r[u_1, \ldots, u_k, v_{k+m+1}, \ldots, v_{k+m+\ell}]] = t_{n+1}$.
By (18), $p \leftrightarrow^*_{P_{i+1}} t_1 \leftrightarrow^*_{P_{i+1}} t_2 \leftrightarrow^*_{P_{i+1}} \cdots \leftrightarrow^*_{P_{i+1}} t_n \leftrightarrow^*_{P_{i+1}} t_{n+1}$.

□

By Statement 3, Lemma 12, and Lemma 13 we have the following result.

Lemma 14. *For each $i \geq 2$, if $R_{i-1} = R_i = P_{i+1}$, then for each $q' \in T_\Sigma$, $p \leftrightarrow^*_{P_i} q'$ if and only if $p \leftrightarrow^*_E q'$.*

We can show the following in the same way as Theorem 1.

Theorem 5. *If $p \leftrightarrow^*_E q$, then procedure PRO4 outputs 'yes' and halts.*

We can show the following in the same way as Theorem 2.

Theorem 6. *If procedure PRO4 outputs 'yes' and halts, then $p \leftrightarrow^*_E q$. If procedure PRO4 outputs 'no' and halts, then $(p, q) \notin \leftrightarrow^*_E$.*

Theorems 5 and 6 imply the following.

Theorem 7. *If $p \leftrightarrow^*_E q$, then procedure PRO4 outputs 'yes' and halts. Otherwise, either PRO4 outputs 'no' and halts, or PRO4 does not halt at all.*

7 Comparison with the Knuth-Bendix completion procedure

We now compare procedures *PRO3* and *PRO4* with the basic Knuth-Bendix completion procedure (see Section 7.1 in [1]), the improved version of the Knuth-Bendix completion procedure described by a set of inference rules (see Section 7.2 in [1]), the goal-directed completion procedure based on SOUR graphs [13, 14], and the unfailing Knuth-Bendix completion procedure [2]. In contrast to all versions of the Knuth-Bendix procedure, Procedures *PRO3* and *PRO4* do not compute any critical pairs and do not use a reduction order. They do not attempt to construct a convergent TRS equivalent to E. When *PRO3* and *PRO4* run a congruence closure algorithm for the TES E over the subterm graph of E [4, 15], they compute and then process only finitely many ground instances $(\overline{s}, \overline{t})$ of finitely many elements (s, t) of the relation $\leftrightarrow_E^*$, where s, t may contain variables. Here (s, t) need not be a critical pair computed by the basic Knuth-Bendix completion procedure. In fact, the ground instances $(\overline{s}, \overline{t})$ are elements of the equivalence relation $\leftrightarrow_E^* \cap (sub(E) \times sub(E))$. Procedures *PRO3* and *PRO4* compute a representative r of $\overline{s}$ and $\overline{t}$ for the equivalence relation $\leftrightarrow_E^* \cap (sub(E) \times sub(E))$. The representative r becomes the normal form of $\overline{s}$ and $\overline{t}$ for the rewrite relation induced by the constructed reduced GTRS. Hence, *PRO3* and *PRO4* do not compare the normal forms of s and t via any reduction order. In contrast, the basic Knuth-Bendix completion procedure reduces the terms in each critical pair to their normal forms. Then tries to orient the normal forms into a rewrite rule. In this way the procedure orients all instances of these terms as well. The improved version of the Knuth-Bendix completion procedure described by a set of inference rules (see Section 7.2 in [1]) also processes each critical pair and also orients the obtained pair, and hence all of its instances. The unfailing Knuth-Bendix completion procedure [2] applies orientable instances of equations in E with respect to a reduction order $>$.

To illustrate the efficiency of the goal-directed completion procedure, Lynch [13] presented the following example. Let the ranked alphabet Σ consist of the unary symbols f, g and the nullary symbols \$, #. Consider the variable preserving TES $E = \{ ffx \approx gfx \}$. We raise the problem whether $\$ \leftrightarrow_E^* \#$. On the one hand, the basic Knuth-Bendix completion procedure runs forever on this example [13]. On the other hand, the goal-directed completion procedure does not generate any rule applicable to \$ or #. Therefore, the goal-directed completion procedure outputs 'no' and halts [13]. Lynch and Strogova [14] said that "the goal-directed completion procedure compiles the TES E and the goal (p, q). After the compilation is finished, we cannot apply a schematization of an equation in the completed system. Therefore, the goal-directed completion procedure outputs 'no' and halts. This is an example where the goal-directed completion procedure is superior to the basic Knuth-Bendix algorithm." It is still open whether the goal-directed completion procedure halts on the TES E and any goal [13]. As for the above example, *PRO3* gives the correct answer and then halts on the TES E and any terms $p, q \in T_\Sigma$.

We conjecture that there are variable preserving TES E and ground terms p, q

such that Conditions (a)-(c) hold.

(a) The basic Knuth-Bendix completion procedure runs forever on E.

(b) There is a goal (p, q) such that the goal-directed completion procedure does not stop on E and (p, q).

(c) Procedure *PRO3* gives the correct answer and then halts on the TES E and any terms $p, q \in T_\Sigma$.

Let TES E be as in Example 11. We conjecture that there is $q \in T_\Sigma$ such that the symbols a, c do not appear in q and that the goal-directed completion procedure does not halt on the TES E and the goal $(aaa\$, q)$. On the other hand, let $q \in T_\Sigma$ be arbitrary such that the symbols a, c do not appear in q. On the input E, $aaa\$$, q, Procedure *PRO3* outputs 'no', the correct answer, and then halts, see Example 11.

Procedures *PRO3* and *PRO4* attempt to construct the reduced GTRSs R_P and R_Q, rather than a convergent term rewrite system equivalent to E, such that

- $R_P \cup R_Q \subseteq \leftrightarrow_E^*$,
- $p \leftrightarrow_{R_P}^* q$ or $\leftrightarrow_{R_P}^* \cap(\{ p \} \times T_\Sigma) = \leftrightarrow_E^* \cap(\{ p \} \times T_\Sigma)$, and
- $p \leftrightarrow_{R_Q}^* q$ or $\leftrightarrow_{R_Q}^* \cap(\{ q \} \times T_\Sigma) = \leftrightarrow_E^* \cap(\{ q \} \times T_\Sigma)$.

Thus R_P and R_Q need not be equivalent to E. By contrast, all versions of the Knuth-Bendix completion procedure attempt to transform a given TES E into an equivalent convergent term rewrite sytem. Since Snyder's ground completion algorithm does not apply orderings, procedures *PRO3* and *PRO4* do not apply any orderings as well.

We now present three examples where procedures *PRO3* and *PRO4* compute efficiently, probably more efficiently than all versions of the Knuth-Bendix completion procedure.

Example 20. [8, 16] Gallier et al [8] and Plaisted and Sattler-Klein [16] presented the following problem to illustrate that reducing a ground term to its normal form can take exponential time if a proper strategy is not used. Let $\Sigma = \Sigma_0 \cup \Sigma_1$, $\Sigma_0 = \{ \$ \}$, and $\Sigma_1 = \{ f, g \}$. Let $n \geq 2$. Let the GTRS R consist of the following rules:

$f\$ \to g\$$,
$fg\$ \to gf\$$,
$fg^2\$ \to gf^2\$$,
...
$fg^n\$ \to gf^n\$$.

Plaisted and Sattler-Klein observed the following on page 156 in [16]. Although GTRS R is convergent, the right-hand sides can be further rewritten. An unskilful choice of rewrites can lead to an exponential time of process. The straightforward reduction of the term $gf^n\$$ can take a number of rewrite steps exponential in n. However, if we apply the rules in order of size, smallest first, to all other rules, the whole TRS can be rewritten to a reduced GTRS in a polynomial number of steps.

We form the TES E by adding the equation

$$fg^{n+1}x \approx gf^{n+1}x$$

to the set R. We now run procedure *PRO3* on the variable preserving TES E and the ground terms $p = f^{n+2}\$$ and $q = g^{n+2}\$$. Then

$\{ f\$ \approx g\$ \} = P_1 = R_{P_1} = Q_1 = R_{Q_1}$, $p{\downarrow}_{R_{P_1 \cup Q_1}} \neq q{\downarrow}_{R_{P_1 \cup Q_1}} = q$.
$R_1 \cup \{ fg\$ \approx g^2\$ \} = P_2 = R_{P_2} = Q_2 = R_{Q_2}$, $p{\downarrow}_{R_{P_2 \cup Q_2}} \neq q{\downarrow}_{R_{P_2 \cup Q_2}} = q$.
$R_2 \cup \{ fg^2\$ \approx g^3\$ \} = P_3 = R_{P_3} = Q_3 = R_{Q_3}$, $p{\downarrow}_{R_{P_3 \cup Q_3}} \neq q{\downarrow}_{R_{P_3 \cup Q_3}} = q$.
...
$R_{P_n} \cup \{ fg^n\$ \approx g^n\$ \} = P_n = R_{P_n} = Q_n = R_{Q_n}$, $p{\downarrow}_{R_{P_{n+1} \cup Q_{n+1}}} \neq q{\downarrow}_{R_{P_{n+1} \cup Q_{n+1}}}$.
$R_{P_{n+1}} \cup \{ fg^{n+1}\$ \approx g^{n+1}\$ \} = P_{n+2} = R_{P_{n+2}} = Q_{n+2} = R_{Q_{n+2}}$.
$P_{n+2} = R_{P_{n+3}} = Q_{n+2} = R_{Q_{n+3}}$.

Observe that $p{\downarrow}_{R_{P_{n+2} \cup Q_{n+2}}} = q{\downarrow}_{R_{P_{n+2} \cup Q_{n+2}}} = q$. Hence procedure *PRO3* outputs 'yes' and halts in the $(n+2)$nd step. The number of computation steps is polynomial. It should be clear that for all ground terms p and q, *PRO3* halts. It outputs 'yes' if $p \leftrightarrow_E^* q$. Otherwise it outputs 'no'.

Consider the lexicographic path order $>_{lpo}$ induced by the order $f > g > \$$ [1]. We now run the basic Knuth-Bendix completion procedure on the TES E and the reduction order $>_{lpo}$. In the initialization phase, the basic Knuth-Bendix completion procedure orients the equations of E. We obtain the TRS S consisting of the following rules:

$f\$ \to g\$$,
$fg\$ \to gf\$$,
$fg^2\$ \to gf^2\$$,
...
$fg^n\$ \to gf^n\$$,
$fg^{n+1}x \to gf^{n+1}x$.

Similarly to the first part of the example we have the following. The TRS S has no critical pairs. Hence the basic Knuth-Bendix procedure outputs S. The straightforward reduction of the term $f^{n+2}\$$ to $g^{n+2}\$$ by S takes a number of rewrite steps exponential in n. The improved Knuth-Bendix completion procedure reduces the right-hand sides of the first n rules as in the first part of the example. We obtain the TRS S' consisting of the following rules:

$f\$ \to g\$$,
$fg\$ \to gf\$$, $fg\$ \to gg\$$,
$fg^2\$ \to gf^2\$$, $fg^2\$ \to gfg\$$, $fg^2\$ \to g^3\$$,
...
$fg^n\$ \to gf^n\$$, $fg^n\$ \to gf^{n-1}g\$$, ..., $fg^n\$ \to g^{n+1}\$$,
$fg^{n+1}x \to gf^{n+1}x$.

In the best case, the reduction of the term $f^{n+2}\$$ to $g^{n+2}\$$ applies the rules

$f\$ \to g\$$,
$fg\$ \to gg\$$,
$fg^2\$ \to g^3\$$,
...
$fg^n\$ \to g^{n+1}\$$,
$fg^{n+1}x \to gf^{n+1}x$.

In the worst case, S' applies only the rules of S in the reduction of the term $f^{n+2}\$$ to $g^{n+2}\$$. Hence it takes a number of rewrite steps exponential in n as in the first part of the example. The goal-directed completion procedure computes fast on E and the goal (p, q). For experimental results, see the line of the problem Counter5 in Table 1 in Section 7 in [14].

Example 21. We now modify an example of Plaisted and Sattler-Klein [16] and Lynch and Strogova [14].

Let $n \geq 2$, $\Sigma = \Sigma_0 \cup \Sigma_2$, $\Sigma_0 = \{ \$_1, \$_2, \ldots, \$_n, \#_1, \#_2, \ldots, \#_n \}$, and $\Sigma_2 = \{ f, g \}$. Let the TES E consist of the following equations:

$f(\flat, \flat) \approx f(\#_0, \$_0)$,
$\$_0 \approx f(\$_1, \#_1)$,
$\#_0 \approx g(\#_1, \$_1)$,
$\$_1 \approx f(\$_2, \#_2)$,
$\#_1 \approx g(\#_2, \$_2)$,
$\ldots$
$\$_{n-1} \approx f(\$_n, \#_n)$,
$\#_{n-1} \approx g(\#_n, \$_n)$,
$\$_n \approx \#_n$,
$f(x_1, x_1) \approx g(x_1, x_1)$.

We now run procedure *PRO3* on the variable preserving TES E and the ground terms $p = f(\$_0, \#_0)$ and $q = g(\#_0, \#_0)$. Then

$\{ f(\$_1, \#_1) \approx \$_0,\ g(\#_1, \$_1) \approx \#_0 \} = P_1 = R_{P_1}$,
$\{ g(\#_1, \$_1) \approx \#_0,\ f(\#_0, \#_0) \approx g(\#_0, \#_0) \} = Q_1 = R_{Q_1}$,
$R_{P_1} \cup \{ f(\$_2, \#_2) \approx \$_1,\ g(\#_2, \$_2) \approx \$_1 \} = P_2 = R_{P_2}$,
$R_{Q_1} \cup \{ f(\$_1, \#_1) \approx \$_0,\ f(\$_2, \#_2) \approx \$_1,\ g(\#_2, \$_2) \approx \#_1 \} = Q_2 = R_{Q_2}$,
$\ldots$
$R_{P_{n-1}} \cup \{ f(\$_n, \#_n) \approx \$_{n-1},\ g(\#_n, \$_n) \approx \#_{n-1} \} = P_n = R_{P_n}$,
$R_{Q_{n-1}} \cup \{ f(\$_{n-1}, \#_{n-1}) \approx \$_{n-2},\ g(\#_n, \$_n) \approx \#_{n-1} \} = Q_n = R_{Q_n}$.

R_{P_n} consists of the following rules:

$f(\$_1, \#_1) \to \$_0$,
$g(\#_1, \$_1) \to \#_0$,
$f(\$_2, \#_2) \to \$_1$,
$g(\#_2, \$_2) \to \#_1$,
$\ldots$
$f(\$_n, \#_n) \to \$_{n-1}$,
$g(\#_n, \$_n) \to \#_{n-1}$,
$R_{P_n} \cup \{ \$_n \approx \#_n \} = P_{n+1}$.

$R_{P_{n+1}}$ consists of the following rules:

$f(\$_0, \#_0) \to \$_0$,
$f(\$_1, \$_1) \to \$_0$,
$f(\$_2, \$_2) \to \$_1$,
$\ldots$
$g(\$_n, \$_n) \to \$_{n-1}$,
$\#_0 \to \$_0$,

$\#_1 \to \$_1$,
$\#_2 \to \$_2$,
...
$\#_n \to \$_n$.
$R_{Q_n} \cup \{ f(\$_n, \#_n) \approx \$_{n-1}, \$ \approx \# \} = Q_{n+1}$,
$R_{Q_{n+1}} = R_{P_{n+1}} \cup \{ f(\#_0, \#_0) \to g(\#_0, \#_0) \}$.
$P_{n+2} = R_{P_{n+3}} = Q_{n+2} = R_{Q_{n+3}}$.

Clearly, $p{\downarrow}_{R_{P_{n+1}}} = q{\downarrow}_{R_{P_{n+1}}}$. Consequently, procedure *PRO3* outputs 'yes' and halts in the $(n+1)$st step. The number of computation steps is polynomial.

Consider the lexicographic path order $>_{lpo}$ induced by the order

$$\flat > \$_0 > \$_1 > \cdots > \$_n > \#_0 > \#_1 > \cdots > \#_n > f > g.$$

We now run the basic Knuth-Bendix completion procedure on the TES E and the reduction order $>_{lpo}$. In the initialization phase, the basic Knuth-Bendix completion procedure orients the equations of E. We obtain the TRS S consisting of the following rules:

$\$_0 \to f(\$_1, \#_1)$,
$\#_0 \to g(\#_1, \$_1)$,
$\$_1 \to f(\$_2, \#_2)$,
$\#_1 \to g(\#_2, \$_2)$,
...
$\$_{n-1} \to f(\$_n, \#_n)$,
$\#_{n-1} \to g(\#_n, \$_n)$,
$\$_n \to \#_n$,
$f(\flat, \flat) \to f(\#_0, \$_0)$,
$f(x_1, x_1) \to g(x_1, x_1)$.

The last two rules yield the critical pair $\langle f(\#_0, \$_0), g(\flat, \flat)\rangle$. Observe that $f(\#_0, \$_0)$ has a unique $\to_S$ normal form, and that $size(f(\#_0, \$_0){\downarrow}_S) = 2^{n+1}$. Thus the completed system contains a rule with a left-hand side of size 2^{n+1}. The improved Knuth-Bendix completion procedure also yields the TRS S and the above critical pair. Again, the completed system contains a rule with a left-hand side of size 2^{n+1}. The goal-directed completion procedure based on SOUR graphs [13, 14] stores the term $f(\#_0, \$_0){\downarrow}_S$ in linear space in n.

Example 22. Let $\Sigma = \Sigma_0 \cup \Sigma_1$, $\Sigma_0 = \{ \$ \}$, and $\Sigma_1 = \{ a, b \}$. Let the GTES F consist of the equation $abbax_1 \approx x_1$. Furthermore, let the GTES E consist of the equations

$abbax_1 \approx x_1$, $a\$ \approx \$$, $b\$ \approx \$$.

It is well-known that there is no convergent TRS R equivalent to F, see Theorem 4.2.18 in [10]. Hence there is no convergent TRS R equivalent to E either. Consequently, the basic Knuth-Bendix completion procedure (see Section 7.1 in [1]), the improved version of the Knuth-Bendix completion procedure described by a set of inference rules (see Section 7.2 in [1]) cannot produce a convergent TRS R equivalent to E.

Let $p, q \in T_\Sigma$ be arbitrary. First, we run the procedure *PRO3* on the input E, p, q. Procedure *PRO3* outputs 'yes' and halts in the first or second step. The resulting reduced GTRS is a subset of

$\{ a\$ \to \$, b\$ \to \$ \}$.

Second, we run the goal-directed completion procedure on the input E, (p, q). It computes all critical pairs and then processes them. Then it applies the resulting rules. The goal-directed completion procedure takes more time on E and the goal (p, q) than procedure *PRO3* on the input E, p, q.

8 Conclusion

We recalled the well known trivial semi-decision procedure *PRO1* for the ground word problem of variable preserving TESs and its straightforward generalization, the trivial semi-decision procedure *PRO2* for the ground word problem of TESs. On the basis of *PRO1*, we gave the semi-decision procedure *PRO3* for the ground word problem of variable preserving TESs. We gave examples when procedure *PRO3* was more efficient than procedure *PRO1*. Then we presented the semi-decision procedure *PRO4* for the ground word problem of term equation systems. We obtained it generalizing *PRO3* taking into account *PRO2*. We showed the correctness of *PRO3* and *PRO4*. We compared the procedures *PRO3* and *PRO4* with the basic Knuth-Bendix completion procedure and the goal-directed completion procedure based on SOUR graphs [13, 14].

Procedures *PRO3* or *PRO4* compute in a different way than all versions of the Knuth-Bendix completion procedure. To some instances of the ground word problem of a TES E, they give an answer sooner than all versions of the Knuth-Bendix completion procedure or it is open whether some version of the Knuth-Bendix completion procedure gives an answer at all. Assume that, given a TES E and ground terms p, q, we want to decide whether $p \leftrightarrow_E^* q$. The ground word problem is undecidable even for variable-preserving TESs. Consequently, we have no upper bound on the running time of any type of the Knuth-Bendix completion procedure on the input TES E any reduction order $>$ and the ground terms p, q. However, we assume beforehand that the basic Knuth-Bendix completion procedure or the goal-directed completion procedure or the nonfailing Knuth-Bendix completion procedure will stop on E, $>$, and p, q, and estimate its running time. We base our time estimate on the size of the input and the experimental results by the various implementations [7, 9, 12, 20] of all versions of the Knuth-Bendix completion procedure on inputs of similar size. Then we carry out the following steps. Simultaneously, we start all implementations of all versions of the Knuth-Bendix completion procedure on E and p, q. We wait for the estimated running time. If none of the procedures stop within this time, then they do not stop at all, or we underestimated the running time. Then we start the procedure *PRO3* or *PRO4* depending on whether TES E is variable preserving. In some cases *PRO3* or *PRO4* might give an answer sooner than all implementations of all versions of the Knuth-Bendix completion procedure.

We presented ad hoc examples when procedure *PRO3* was probably more efficient than the goal-directed completion procedure [13, 14]. However, to justify the introduction of procedures *PRO3* and *PRO4*, we need further evidence for the efficiency of the procedures *PRO3* and *PRO4*. We should present implementation results and theoretical arguments. We now raise questions on the efficiency of *PRO3* and *PRO4* compared to the various versions of the Knuth-Bendix completion procedure.

Question 1. *Is it true that for most instances of the ground word problem of a TES E, a correctly chosen version of the Knuth-Bendix completion procedure is more efficient than PRO3 or PRO4?*

Question 2. *For which instances of the ground word problem of a TES E, is a correctly chosen version of the Knuth-Bendix completion procedure more efficient than PRO3 or PRO4?*

Question 3. *Is it decidable for an instance of the ground word problem of a TES E, whether a correctly chosen version of the Knuth-Bendix completion procedure is more efficient than PRO3 or PRO4?*

Question 4. *Is there an instance of the ground word problem of a TES E, such that no version of the Knuth-Bendix completion procedure halts, and PRO3 or PRO4 halts?*

We can reduce an instance of the word problem for a TES E to an instance of the ground word problem for E over a larger alphabet Δ. Let E be a TES and p, q arbitrary terms over a ranked alphabet Σ. Assume that exactly the variables $x_1, \ldots, x_m$ appear in p or q. We now define the ranked alphabet Δ. It contains each element of Σ. Furthermore, for each $i = 1, \ldots, m$, we add a new constant $\#_i$ to Δ. We define p' from p and q' from q by replacing each occurrence of x_i with $\#_i$ for $i = 1, \ldots, m$. Then $p \leftrightarrow_E^* q$ over Σ if and only if $p' \leftrightarrow_E^* q'$ over Δ. Thus if we can decide whether $p' \leftrightarrow_E^* q'$ over Δ, then we can also decide whether $p \leftrightarrow_E^* q$ over Σ.

References

[1] F. Baader and T. Nipkow Term Rewriting and All That, Cambridge University Press, Cambridge, United Kingdom, 1998.

[2] L. Bachmair, N. Dershowitz, D. A. Plaisted, Completion without failure. In Resolution of equations in algebraic structures, Vol. 2, Rewriting techniques. Edited by H. Ait-Kaci and M. Nivat. pp. 130, Academic Press, Boston, MA, 1989,

[3] G. C. Ceitin: Associative calculus with an unsolvable equivalence problem. Tr. Mat. Inst. Akad. Nauk 52, 172-189 (1958) (in Russian).

[4] P. J. Downey, R. Sethi, and R. E. Tarjan: Variations on the Common Subexpression Problem. Journal of the ACM, 27 (1980) 758-771.

[5] Z. Fülöp and S. Vágvölgyi, Ground term rewriting rules for the word problem of ground term equations, Bulletin of the EATCS, **45** (1991) 186-201.

[6] Z. Fülöp and S. Vágvölgyi, Minimal Equational Representations of Recognizable Tree Languages, *Acta Informatica*, **34** (1997) 59-84.

[7] J.-M. Gaillourdet, T. Hillenbrand, B. Löchner, H. Spies: The New WALDMEISTER Loop at Work, in Franz Baader (Ed.): Automated Deduction - CADE-19, 19th International Conference on Automated Deduction, Proceedings. Lecture Notes in Computer Science 2741 Springer 2003, I 317-321.

[8] J. Gallier, P. Narendran, D. Plaisted, S. Raatz, and W. Snyder, An Algorithm for Finding Canonical Sets of Ground Rewrite Rules in Polynomial Time, *Journal of the Association for Computing Machinery*, **40** (1993) 1-16.

[9] T. Hillenbrand, Citius altius fortius: Lessons learned from the Theorem Prover WALDMEISTER, Electronic Notes in Theoretical Computer Science 86(1) (2003).

[10] M. Jantzen, Confluent string rewriting, Springer Verlag, Berlin 1988.

[11] Y. Matiyasevich, G. Sénizergues, Decision Problems for Semi-Thue Systems with a Few Rules, Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, 27-30 July 1996. IEEE Computer Society Press, 523-531.

[12] Serge Mechveliani, From a Computer Algebra Library to a System with an Equational Prover, in Bruno Buchberger, John A. Campbell (Eds.): Artificial Intelligence and Symbolic Computation, 7th International Conference, AISC 2004, Linz, Austria, September 22-24, 2004, Proceedings. Lecture Notes in Computer Science 3249 Springer 2004, 281-284.

[13] C. Lynch, Goal-Directed Completion Using SOUR Graphs, in Hubert Comon (Ed.): Rewriting Techniques and Applications, 8th International Conference, RTA-97, Proceedings. Lecture Notes in Computer Science 1232 Springer 1997, 8-22.

[14] C. Lynch, P. Strogova, SOUR graphs for efficient completion, Discrete Mathematics & Theoretical Computer Science **2** (1998) 1-25.

[15]] G. Nelson, D. C. Oppen: Fast Decision Procedures Based on Congruence Closure. J. ACM **27** (1980) 356-364

[16] D. Plaisted and A. Sattler-Klein, Proof lengths for equational completion, Information and Computation, **125** (1996) 154-170.

[17] S. Vágvölgyi, A fast algorithm for constructing a tree automaton recognizing a congruential tree language, Theoret. Comput. Sci, **115** (1993) 391-399.

[18] S. Vágvölgyi, Ground term rewriting. Bull. Eur. Assoc. Theor. Comput. Sci. EATCS No. 102 (2010), 153190.

[19] W. Snyder, A Fast Algorithm for Generating Reduced Ground Rewriting Systems from a set of Ground Equations, Journal of Symbolic Computation, 15 (1993) 415-450.

[20] I. Wehrman, A. Stump, E. M. Westbrook: Slothrop: Knuth-Bendix Completion with a Modern Termination Checker, Frank Pfenning (Ed.): Term Rewriting and Applications, 17th International Conference, RTA 2006, Proceedings. Lecture Notes in Computer Science 4098 Springer 2006, 287-296.

Received 5th June 2015