A Polynomial Algorithm for Uniqueness of Normal Forms of Linear Shallow Term Rewrite Systems[1]

Julian Zinn[2] and Rakesh Verma

Computer Science Department
University of Houston
Houston, TX, 77204, USA
http://www.cs.uh.edu

UH-CS-10-07
August 25, 2010

**Keywords**: tree automata, polynomial-time algorithm, term rewrite system, uniqueness of normal forms

## Abstract

Term rewrite systems are useful in many areas of computer science. Two especially important areas are decision procedures for the word problem of some algebraic systems and rule-based programming. One of the most studied properties of rewrite systems is confluence, and one of the primary benefits of having a confluent rewrite system is that the system also has uniqueness of normal forms. However, uniqueness of normal forms is an interesting property in its own right and well studied. Also, confluence can be too strong a requirement for applications. In this paper, we study the decidability of uniqueness of normal forms. Uniqueness of normal forms is decidable for ground rewrite systems, but is undecidable in general. This paper shows that the uniqueness of normal forms problem is decidable for the class of linear shallow term rewrite systems, and gives a decision procedure that is polynomial as long as the arities of the function symbols are bounded or the signature is fixed.

[1] Thanks to NSF grants CCF 0306475 and OCI 0453498.
[2] Work done while the author was at the University of Houston.

# A Polynomial Algorithm for Uniqueness of Normal Forms of Linear Shallow Term Rewrite Systems

Julian Zinn
Computer Science Dept.
University of Houston
Houston, TX 77204
jzinn@cs.uh.edu

Rakesh Verma
Computer Science Dept.
University of Houston
Houston, TX 77204
Ph: 713-743-3348/Fax: 713-743-3335
rmverma@cs.uh.edu

June 5, 2008

**Abstract**

Term rewrite systems are useful in many areas of computer science. Two especially important areas are decision procedures for the word problem of some algebraic systems and rule-based programming. One of the most studied properties of rewrite systems is confluence, and one of the primary benefits of having a confluent rewrite system is that the system also has uniqueness of normal forms. However, uniqueness of normal forms is an interesting property in its own right and well studied. Also, confluence can be too strong a requirement for applications. In this paper, we study the decidability of uniqueness of normal forms. Uniqueness of normal forms is decidable for ground rewrite systems, but is undecidable in general. This paper shows that the uniqueness of normal forms problem is decidable for the class of linear shallow term rewrite systems, and gives a decision procedure that is polynomial as long as the arities of the function symbols are bounded or the signature is fixed.

## 1 Introduction

Term rewrite systems (TRSs), which are finite sets of rules, are useful in many areas of computer science. Two especially important areas are decision procedures for the word problem of some algebraic systems and rule-based programming. One of the most studied properties of rewrite systems is confluence, and one of the primary benefits of having a confluent rewrite system is that the system also has uniqueness of normal forms ($\mathrm{UN}^=$). However, uniqueness of normal forms is an interesting property in its own right and well-studied [10]. Also, confluence can be too strong a requirement for some applications such as lazy rule-based programming. Additionally, in the proof-by-consistency approach for inductive proofs, consistency is often ensured by requiring the $\mathrm{UN}^=$ property. Our algorithm may be used as a decidable sufficient condition ensuring $\mathrm{UN}^=$ for left-linear systems using approximation techniques.

The uniqueness of normal forms problem is as follows:

**Input** A TRS $R$.

**Question** For all normal forms $n$ and $m$ such that $n \leftrightarrow_R^* m$ is $n = m$?

In this paper, we study the decidability of uniqueness of normal forms. Uniqueness of normal forms is decidable for ground systems [13], but is undecidable in general [13]. Since the property is undecidable in general, we would like to know for which classes of rewrite systems we can decide $\mathrm{UN}^=$. In this paper, we consider the class of linear shallow systems, and a subset of this class, the linear flat systems. A rewrite system is linear if variables occurs at most once in each side of any rule. A rewrite system is shallow if variables occur only at depth zero or depth one in each side of any rule. And, a rewrite system is flat if the parse trees of both the left- and right-hand sides of all the rules have height zero or one.

An example of a linear flat system that has $\mathrm{UN}^=$ but not confluence is $\{f(c) \rightarrow 1,\, c \rightarrow g(c)\}$. More sophisticated examples can be constructed using a sequential 'or' function in which the second argument gives rise to a nonterminating computation.

This paper shows that the uniqueness of normal forms problem is decidable for the class of linear shallow term rewrite systems, and gives a decision procedure that is polynomial as long as the arities of the function symbols are bounded or the signature is fixed. Even the decidability of $\mathrm{UN}^=$ for this class of systems was not known earlier, to the best of our knowledge.

There are a few related problems that do not quite combine to give a solution to this problem. As is well known, for a given TRS $R$, the problem of determining if a ground term $t$ is a normal form with respect to $R$ is decidable in polynomial time. We show this for shallow linear TRS in this paper. It is also possible to decide, given two ground terms $s$ and $t$, if $s \leftrightarrow_R^* t$, which is also shown in this paper. Additionally, testing if $s = t$ (syntactic identity) is trivial. However, this does not give us a decision procedure for $\mathrm{UN}^=$, because we cannot check for each pair $\langle s, t\rangle$ of terms (there are infinitely many) that if $s$ and $t$ are normal forms and $s \leftrightarrow_R^* t$ then $s = t$.

Our decision procedure works by determining whether there are any witnesses to non-$\mathrm{UN}^=$. A witness to non-$\mathrm{UN}^=$ is a pair of normal forms $\langle n, m\rangle$ such that $n \leftrightarrow_R^* m$ but $n \neq m$. The decision procedure depends on the following results, the first three of which we prove in the course of the paper:

1. If $R$ is a linear shallow TRS, then we can transform $R$ in polynomial time into a linear flat TRS $R'$ such that $R$ is $\mathrm{UN}^=$ if and only if $R'$ is $\mathrm{UN}^=$. We show this in Section 3. Our transformation is necessarily different from the flattening procedures of [5, 13], since those procedures preserve confluence but do not preserve $\mathrm{UN}^=$.

2. If $R$ is a linear flat TRS and $R$ is not $\mathrm{UN}^=$ then there is a witness $\langle n, m\rangle$ to non-$\mathrm{UN}^=$ such that $n$ and $m$ are ground, there is a flat ground term $t$ and ground derivation $n \leftrightarrow_R^* t \leftrightarrow_R^* m$, and $t$ is an instance of a left-hand side of a rule of $R$. We show this in Section 4.

3. If $R$ is a linear flat TRS and $t$ is any ground term, then we can construct in polynomial time a tree automaton $A_t$ that recognizes the ground normal forms that are $R$-equivalent to $t$ by ground derivations. This result is known [3, 4], but we improve the construction and proof, and provide a complexity analysis needed to show that our decision procedure is polynomial.

4. For any tree automaton $A$, we can decide in polynomial time if the language accepted by $A$ contains at most one element [4].

We construct the automaton $A_t$ of item 3 from two simpler automata described in Sections 5 and 6. For a left-linear rewrite system $R$, Section 5 describes a tree automaton $A_{\mathrm{Red}(R)}$ that accepts exactly the normal forms of $R$. For a linear shallow rewrite system $R$, Section 6 describes how to construct, for a ground term $t$, a tree automaton $B$ that accepts the terms that are $R$-equivalent to $t$ by ground derivations. From these two automata, we can create, for a linear flat rewrite system $R$, an intersection machine $A_t$ that accepts normal forms $R$-equivalent to $t$ by ground derivations. Intersection machines are described in [4].

The decision procedure described in item 4 comes from two results described in [4]. The first of these results is that for any tree automaton $A$, we can decide if $A$ has the 'emptiness property'—is the language of $A$ empty. The second result is that for any tree automaton $A$, we can decide if $A$ has the 'singleton set property'—does the language of $A$ contain exactly a single element.

Using these results, we can decide the $\mathrm{UN}^=$ problem with the following algorithm. Given a linear shallow TRS $R$ as input, first flatten $R$ to produce a linear flat TRS $R'$. If for each flat ground instance $t$ of a left-hand side of a rule of $R'$ the automaton $A_t$ accepts at most one term, then $R$ is $\mathrm{UN}^=$. Otherwise $R$ is not $\mathrm{UN}^=$.

The algorithm always terminates because there are only finitely many flat ground instances of left-hand sides of rules of $R'$. The procedure is correct because if for each flat ground instance $t$ of a left-hand side of a rule of $R'$ the automaton $A_t$ accepts at most one term, then there are no ground witnesses to non-UN$^=$ of $R'$ that have a ground derivation including a flat ground instance of a left-hand side of a rule. If this is the case, then $R'$ is UN$^=$. On the other hand, if there is a flat ground instance $t$ of a left-hand side of a rule of $R'$ such that the automaton $A_t$ accepts more than one term, then there is a witness to non-UN$^=$, so $R'$ is not UN$^=$.

The decision procedure for UN$^=$ just described is a polynomial-time algorithm. The number of flat ground instances of left-hand sides of rule is polynomial in the size of $R$. In Sections 5 and 6, we shall see that for each such term $t$, the time to construct the machine $A_t$ is polynomial in the size of $R$ if the arities of function symbols are bounded. The intersection machine $A_t$ has size $|A_{\mathrm{Red}(R)}| \times |B|$ [4]. Thus the size of $A_t$ is also polynomial in the size of $R$ if the arities of function symbols are bounded or the signature is fixed. Finally, we can decide the emptiness property and singleton set property for any tree automaton $A$ in time polynomial in the size of $A$ [4].

### 1.1 Related Work

This paper extends results from three main articles and applies them to the uniqueness of normal forms problem.

The key idea of Section 4 is to use a case analysis that depends on whether a term is equivalent to height-zero term or not. This insight comes from Section 5.3 of [7], where it is used to prove a confluence result. Section 4 uses it to show that one of the terms in any equational proof of the equality of two normal forms is a flat instance of a left-hand side of a rule.

The tree automata from Sections 5 and 6 closely follow the automata in [3], but contain some differences and clarifications. Section 5 on tree automata for reducibility contains a complexity analysis for linear flat rewrite systems. Section 6 on tree automata for reachability describes automata that recognize terms reachable from a term $t$ rather than recognize terms from which $t$ is reachable.

Regular flattening, discussed in Section 3, is used in [5] to develop a confluence preserving transformation that turns a shallow TRS into a flat TRS. However, this paper shows that regular flattening cannot be used in the same way for a UN$^=$ (or UN$^\rightarrow$) preserving transformation. After this negative result, this paper provides two variations on flattening that allow us to construct a UN$^=$ and UN$^\rightarrow$ preserving algorithm that transforms a linear shallow TRS into a linear flat TRS.

Many other decidability results are known about the class of linear shallow rewrite systems. The word problem for shallow rewrite systems is decidable in polynomial time [9]. Also, confluence, reachability, and joinability are decidable for the linear, shallow class [7]. Decidability of termination for the linear, shallow class follows from a more general result [6].

## 2 Preliminaries

### 2.1 Terms

A *signature* is a set $\mathcal{F}$ along with an function $arity\colon \mathcal{F} \to \mathbb{N}$. Members of $\mathcal{F}$ are called *function symbols*, and $arity(f)$ is called the *arity* of a function symbol $f$. Function symbols of arity zero are called *constants*. Let $X$ be a countable set disjoint from $\mathcal{F}$ that we shall call the set of *variables*. The set $\mathcal{T}(\mathcal{F}, X)$ of *$\mathcal{F}$-terms over $X$* is defined to be the smallest set that contains $X$ and has the property that $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}, X)$ whenever $f \in \mathcal{F}$, $n = arity(f)$, and $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, X)$. A term is called *ground* if no variable occurs in it. The set of ground terms over signature $\mathcal{F}$ is denoted $\mathcal{T}(\mathcal{F})$. A term is called *linear* if no variable occurs more than once in it.

The *size* $|t|$ of a term $t$ is the number of occurrences of variables and function symbols in $t$. Thus $|t| = 1$ if $t$ is a variable, and $|t| = 1 + |t_1| + \cdots + |t_n|$ if $t = f(t_1, \ldots, t_n)$. In particular, the size of a constant is 1. The *height* of a term $t$ is 0 if $t$ is a constant or variable, and $1 + \max\{\mathrm{height}(t_1), \ldots, \mathrm{height}(t_n)\}$ if $t = f(t_1, \ldots, t_n)$. Because of this definition, variables and constants are also known as *height-zero terms*. Terms that have height zero or one are called *flat*.

A *position* of a term $t$ is a sequence of natural numbers that is used to identify the locations of subterms of $t$. The empty sequence $\lambda$ is a position that identifies the subterm $t$ itself. The set $\mathrm{Pos}(t)$ of positions of $t$ is defined by $\mathrm{Pos}(t) = \{\lambda\}$ if $t$ is a variable, and $\mathrm{Pos}(t) = \{\lambda\} \cup \{1.p \mid p \in \mathrm{Pos}(t_1)\} \cup \cdots \cup \{n.p \mid p \in \mathrm{Pos}(t_n)\}$ if $t = f(t_1, \ldots, t_n)$. We can define a partial order $\leq$ on $\mathrm{Pos}(t)$ by $p \leq q$ if and only if $p$ is a prefix of $q$, i.e. there is a sequence $p'$ such that $q = pp'$. We say that $p$ is *above* $q$ if $p \leq q$, and we say that $p$ is *below* $q$ if $p \geq q$. We say that positions $p$ and $q$ are *parallel* if they are incomparable with respect to $\leq$. If $t$ is a term and $p$ is a position, then $t|_p$ is the subterm of $t$ at position $p$. More formally defined, $t|_\lambda = t$ and $f(t_1, \ldots, t_n)|_{i.p} = t_i|_p$.

If $s$ is a subterm of $t$ that occurs at a position $p$ that has length $d$, then the *depth* of $s$ in $t$ is $d$. A term is called *shallow* if no variable occurs at depth greater than one. Thus, every flat term is shallow, but not vice versa.

We denote by $t[s]_p$ the term that is like $t$ except that the subterm $t|_p$ is replaced by $s$. More formally defined, $t[s]_\lambda = s$ and $f(t_1, \ldots, t_n)[s]_{i.p} = f(t_1, \ldots, t_i[s]_p, \ldots, t_n)$. For example, if $t = f(g(a), b, g(h(c, b)))$ then $t|_{3.1.2} = b$ and $t[c]_3 = f(g(a), b, c)$.

A notational device called a *context* is useful when performing replacements. Intuitively, a context is a term with one or more 'holes' into which terms may be inserted. We can provide a formal definition by considering a context to be a term in an extended signature that includes an extra constant symbol $\square$. If $C$ is a context with one occurrence of $\square$, then we write $C$ as $C[\,]$. If $C$ contains two occurrences of $\square$, then we write $C$ as $C[\,,\,]$, and so on. If $C[\,, \ldots,\,]$ is a context with $n$ occurrences of $\square$, then $C[t_1, \ldots, t_n]$ represents the term that is like $C$ except that the occurrences of $\square$ are replaced with the terms $t_1, \ldots, t_n$. For example, if $C[\,,\,] = f(a, \square, g(\square))$, then $C[g(a), g(b)] = f(a, g(a), g(g(b)))$.

A *substitution* is a mapping $\sigma\colon X \to \mathcal{T}(\mathcal{F}, X)$ that is identified with its homomorphic extension $\hat{\sigma}\colon \mathcal{T}(\mathcal{F}, X) \to \mathcal{T}(\mathcal{F}, X)$, which agrees with $\sigma$ on $X$ and is such that $\hat{\sigma}(f(t_1, \ldots, t_n)) = f(\hat{\sigma}(t_1), \ldots, \hat{\sigma}(t_n))$. The *domain* of a substitution $\sigma$ is the set $\{x \in X \mid x \neq \sigma(x)\}$. We define a substitution $\sigma$ with finite domain $\{x_1, \ldots, x_n\}$ by using the notation $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$. Applications of substitutions to terms are commonly written in postfix notation, i.e. $t\sigma$ rather than $\sigma(t)$. An example of a substitution is $\sigma = \{x \mapsto f(a, b),\, y \mapsto g(b)\}$. Application of $\sigma$ to a term $t = f(g(y), x)$ results in $t\sigma = f(g(g(b)), f(a, b))$. For terms $s$ and $t$, if $t = s\sigma$ for some substitution $\sigma$, then $t$ is said to be an *instance* of $s$. The composition of substitutions $\sigma$ and $\tau$ is denoted by $\sigma\tau$ and is defined by $t(\sigma\tau) = (t\sigma)\tau$. If $\sigma$ and $\tau$ are substitutions and there is a substitution $\sigma'$ such that $\tau = \sigma\sigma'$, then $\sigma$ is said to be more general than $\tau$.

Two terms $s$ and $t$ are *unifiable* if there is a substitution $\sigma$ such that $s\sigma = t\sigma$. In this case $\sigma$ is called a *unifier* of $s$ and $t$, and $s\sigma$ is called a *most general instance* of $s$ and $t$. If two terms $s$ and $t$ are unifiable, then they have a most general unifier $\sigma$ in the sense that for any unifier $\tau$ of $s$ and $t$ there is a substitution $\sigma'$ such that $\tau = \sigma\sigma'$.

## 2.2 Term Rewrite Systems

A *rewrite rule* is a pair $\langle l, r\rangle$ of terms typically written $l \to r$. For the rule $l \to r$, the *left-hand side* is $l$ and the *right-hand side* is $r$. A rule $l \to r$ is *flat* (*shallow*, *linear*, *ground*) if both $l$ and $r$ are flat (shallow, linear, ground). A rule $l \to r$ is *left-linear* if $l$ is linear, and it is *right-linear* if $r$ is linear. A rule $l \to r$ is *collapsing* if $r$ is a variable. Variables that occur in both sides of a rule are called *shared* variables. Variables that occur in one side of a rule but not the other are called *non-shared* variables.

A *term rewrite system* (TRS) is a pair $\langle \mathcal{T}, R\rangle$ where $R$ is a set of rules and $\mathcal{T}$ is the set of terms over a particular signature. We only treat finite rewrite systems in this paper. We require the standard restrictions that a left-hand side of a rule may not be a variable, but not that a variable may occur in the right-hand side of a rule only if it occurs in the left-hand side. Usually only the rules are emphasized and the terms are assumed to be those that can be built from the symbols occurring in the rules. For a set $R$ of rules, the *rewrite relation* $\to_R$ is a binary relation on terms defined by $s \to_R t$ if and only if there is a position $p \in \mathrm{Pos}(s)$, a rule $l \to r \in R$, and a substitution $\sigma$ such that $s|_p = l\sigma$ and $t = s[r\sigma]_p$. Here, $p$ is called the position of the rewrite application, and $s|_p$ is called a *redex* (reducible expression) of $s$. If a variable $x$ occurs at position $p'$ of $l$, then the subterm $l\sigma|_{p'}$ (which is also the subterm $s|_{p.p'}$) is called the substitution part of $x$. Using contexts, we can say that $s \to_R t$ if and only if there is a context $C[\,]$, rule $l \to r \in R$, and substitution $\sigma$ such that $s = C[l\sigma]$ and $t = C[r\sigma]$. A TRS $R$ is *flat* (*shallow*, *linear*, *ground*) if each rule is flat (shallow, linear, ground). A TRS $R$ is *left-linear* (*right-linear*) if each rule is left-linear (right-linear). If $R$ is a set of rules, then we define $R^-$ by $R^- = \{r \to l \mid l \to r \in R\}$.

If $\to$ is a binary relation, then its inverse is denoted $\leftarrow$, its reflexive closure $\to^=$, its symmetric closure $\leftrightarrow$, its transitive closure $\to^+$, and its reflexive transitive closure $\to^*$ or $\twoheadrightarrow$. The $n$-fold composition of $\to$ is denoted by $\to^n$. The reflexive symmetric transitive closure is denoted $\leftrightarrow^*$, which is also called the equivalence closure. If $R$ is a rewrite system, then terms related by $\leftrightarrow_R^*$ are called *R-equivalent* or *convertible*.

A term $s$ is *reachable* by $R$ from a term $t$ if $t \to_R^* s$. Terms $s$ and $t$ are called *joinable* by $R$, denoted $s \downarrow_R t$, if there is a term $u$ such that $s \to_R^* u$ and $t \to_R^* u$. In other words, $\downarrow_R = \to_R^* \circ \leftarrow_R^*$. We also define a relation $\uparrow_R$ by $s \uparrow_R t$ if and only if there is a term $u$ such that $u \to_R^* s$ and $u \to_R^* t$. In other words, $\uparrow_R = \leftarrow_R^* \circ \to_R^*$. A term $t$ is called a *normal form* if there is no term $s$ for which $t \to_R s$. The set of normal forms of $R$ is denoted $\mathrm{nf}(R)$. Terms that are not normal forms are called *reducible*.

A TRS $R$ is *terminating* if for every term $t$ there is no infinite reduction sequence $t = t_0 \to_R t_1 \to_R t_2 \to_R \cdots$. A rewrite system is *confluent* if two rewrite sequences from the same term can always be extended to end with the same term. Formally, a TRS $R$ is confluent if for every term $u$, if $u \to_R^* t_1$ and $u \to_R^* t_2$ for terms $t_1$ and $t_2$ then $t_1 \to_R^* v$ and $t_2 \to_R^* v$ for some term $v$. In other words, $R$ is *confluent* when $\leftarrow_R^* \circ \to_R^* \subseteq \to_R^* \circ \leftarrow_R^*$.

We say that a TRS $R$ has *uniqueness of normal forms*, or that $R$ is $\mathrm{UN}^=$, if for every term $s$, if $s \leftrightarrow_R^* n$ and $s \leftrightarrow_R^* m$ where $n$ and $m$ are normal forms, then $n = m$. Another way of stating this is that $R$ is $\mathrm{UN}^=$ if and only if for all normal forms $n$ and $m$, if $n \leftrightarrow_R^* m$ then $n = m$. We say that $R$ is *uniquely normalizing*, or that $R$ is $\mathrm{UN}^\to$, if for every term $s$, if $s \to_R^* n$ and $s \to_R^* m$ where $n$ and $m$ are normal forms, then $n = m$. Another way of stating this is that $R$ is $\mathrm{UN}^\to$ if and only if for all normal forms $n$ and $m$, if $n \uparrow_R m$ then $n = m$. Note that if a term rewrite system is $\mathrm{UN}^=$, then it is $\mathrm{UN}^\to$.

If $\to$ is a binary relation on terms, then a *derivation* $s \to^* t$ is a finite sequence $s_0, \ldots, s_k$ of terms such that $s = s_0$, $s_k = t$, and $s_i \to s_{i+1}$ for each $i$. Each pair $s_i \to s_{i+1}$ is called a *step* of the derivation, and the position of the rewrite, the rule, and substitution used are assumed to be known for each step, although this information may not be written down. Derivations are often written as $s_0 \to \cdots \to s_k$. The length of a derivation is the number of steps in it, so the derivation $s_0 \to \cdots \to s_k$ has length $k$. A *ground derivation* of $s \to^* t$ is a derivation $s_0, \ldots, s_k$ in which each $s_i$ is a ground term. When the binary relation $\to$ is a rewrite relation $\to_R$, a *root rewrite step* is a step $s_i \to_R s_{i+1}$ such that the position of the rewrite is $\lambda$. If $s_i \to_R s_{i+1}$ is a root rewrite step, it may be notated by $s_i \xrightarrow{\mathrm{r}}_R s_{i+1}$.

## 2.3 Tree Automata

A *nondeterministic (bottom-up) tree automaton* $A$ is a tuple $\langle \mathcal{F}, Q, Q_f, \Delta \rangle$ where $\mathcal{F}$ is a signature, $Q$ is a set of states, $Q_f \subseteq Q$ is set of final states, and $\Delta$ is a set of transition rules of the form

$$f(q_1, \ldots, q_n) \to q\,,$$

where $f \in \mathcal{F}$, the arity of $f$ is $n$, and $q, q_1, \ldots, q_n \in Q$.

Tree automata take ground terms over $\mathcal{F}$ as input. Starting at the leaves and proceeding upward, an automaton associates a state with subterms of the input until finally no transition rules are applicable. An automaton may or may not be able to associate a state with every subterm of the input. Tree automata have no start states. The leaves of the input term are constants, and for a constant $a$ that is a leaf of the input, automata rely on transition rules of the form $a() \to q$ to get started. Then, if a term $t = f(t_1, \ldots, t_n)$ is a subterm of the input, and there are states $q_1, \ldots, q_n$ such that for each $i$ the automaton has associated state $q_i$ with term $t_i$, and there is a transition rule $f(q_1, \ldots, q_n) \to q$, then the automaton may associate state $q$ with subterm $t$. There may be other transition rules with the same left-hand side but different right-hand side, so the automaton may associate a different state with $t$. If the automaton finally associates a state $q$ with the input term, then the input term is said to *reach* $q$. The automaton *accepts* the input if the input reaches some state in $Q_f$. The *language* $\mathcal{L}(A)$ of a tree automaton $A$ is the set of terms that $A$ accepts. A tree automaton is *deterministic* if no two distinct transition rules have the same left-hand side. A tree automaton is *complete* if for every term $t$ there is a state that $t$ reaches. For more information on tree automata, consult [4].

Once a subterm has been associated with a state, all that matters is the state, and we may forget the subterm. A *configuration* of a tree automaton $A$ is a term in which some of the subterms have been replaced by states. We may consider the set of configurations to be the set of terms over the signature $\mathcal{F} \cup Q$, where the states are treated as constants in the signature. We define a *move relation* $\vdash_A$ on configurations by $s \vdash_A t$ if and only if there is a context $C[\,]$ and transition rule $f(q_1, \ldots, q_n) \to q$ such that $s = C[f(q_1, \ldots, q_n)]$ and $t = C[q]$. $\Delta$ can be modeled as a rewriting System over $\mathcal{F} \cup Q$, so $\vdash_A$ is the same as $\to_\Delta$. Using this terminology, an automaton $A$ accepts a term $t$ if and only if $t \vdash_A^* q$ for some final state $q$. A *computation* of a tree automaton $A$ is a sequence $s_0, \ldots, s_n$ of configurations such that $s_i \vdash_A s_{i+1}$ for each $i$.

# 3 Flattening Shallow Rewrite Systems

This section describes a polynomial-time algorithm that transforms an arbitrary shallow TRS into a flat TRS while preserving $\mathrm{UN}^=$ and linearity. For a property $P$ of term rewrite systems, a transformation of a TRS $R$ into a TRS $R'$ is called $P$-preserving when $R$ has property $P$ if and only if $R'$ has property $P$.

Additionally, we will show that the transformation preserves $\mathrm{UN}^\to$. $\mathrm{UN}^\to$ is similar to $\mathrm{UN}^=$, and we would eventually like to extend the result of this paper to show that $\mathrm{UN}^\to$ is decidable for linear shallow rewrite systems.

Godoy et al. [5] introduced a flattening algorithm, called *regular flattening* here, that transforms a shallow TRS into a flat TRS while preserving confluence. Each iteration of the regular flattening algorithm chooses a non-constant ground term $t$ and replaces all of its occurrences in the rules of $R$ by a new constant $c$ and adds the rules $c \to t$ and $t \to c$ to $R$.

Unfortunately, regular flattening does not preserve $\mathrm{UN}^=$ or $\mathrm{UN}^\to$. After providing examples of this, we describe two alternate kinds of rule flattening—*flattening on the right* and *flattening on the left*—that each preserve both $\mathrm{UN}^=$ and $\mathrm{UN}^\to$. In flattening on the right, we choose one occurrence of a non-constant flat ground term $r_0$ in the right-hand side of a rule, replace it with a new constant $c$, and add the rule $c \to r_0$. In flattening on the left, we choose every occurrence of a non-constant flat ground term $l_0$ in the left-hand side of a rule, replace it with a new constant $c$, add the rule $l_0 \to c$, and add the rule $c \to c$ if $l_0$ is reducible. It is easier to show preservation of $\mathrm{UN}^=$ and $\mathrm{UN}^\to$ for flattening on the right than flattening on the left because flattening the right-hand side of a rule does not affect the set of normal forms (other than is caused by adding the new constant $c$).

We can iteratively apply flattening on the left and flattening on the right to any arbitrary TRS, and each step will preserve $\mathrm{UN}^=$, linearity, and $\mathrm{UN}^\to$. The transformation preserves linearity because no variable is touched during any step. This process terminates, and the size of the final TRS is polynomial in the size of the original TRS. If we start with a shallow TRS, then we will eventually end up with a flat TRS.

Each step in the flattening process requires that we introduce a new constant $c$ to the signature. Thus we must be mindful of the signature of $R$. We denote by $\mathcal{T}$ the set of terms over the original signature $\mathcal{F}$, and by $\mathcal{T}_c$ the set of terms over the signature $\mathcal{F} \cup \{c\}$, where $c$ is a new constant not in $\mathcal{F}$. We write $\langle \mathcal{T}, R \rangle$ to specify that we are talking about rules $R$ over terms $\mathcal{T}$.

In the following proofs, we will need to perform multiple simultaneous replacement of ground terms with other ground terms. The notation we use for this is $u[t \Rightarrow c]$ to represent the term that results from simultaneously replacing all instances of the ground term $t$ in $u$ with $c$. Similarly, $u[c \Rightarrow t]$ represents the term that results from replacing all instances of the constant $c$ with the ground term $t$. Note that if $u \in \mathcal{T}_c$, then $u$, $u[c \Rightarrow t]$, and $u[t \Rightarrow c]$ may be three distinct terms, and we may have $u \neq (u[t \Rightarrow c])[c \Rightarrow t]$ and $u \neq (u[c \Rightarrow t])[t \Rightarrow c]$.

## 3.1 Regular Flattening preserves neither $\mathrm{UN}^=$ nor $\mathrm{UN}^\to$

For this section, let $R$ be a rewrite system over $\mathcal{T}$ and $t$ be a ground term that is a subterm of a side of some rule of $R$. Also let $\text{b-flat}\, R = R^b \cup \{c \to t,\, t \to c\}$, where $c$ is a new constant not in $\mathcal{F}$, and $R^b$ is $R$ with every occurrence of $t$ in both sides of every rule replaced by $c$.

To show that regular flattening does not preserve $\mathrm{UN}^=$, we can exhibit one of the following

1. A TRS $R$ such that $R$ is $\mathrm{UN}^=$ but b-flat $R$ is not $\mathrm{UN}^=$.
2. A TRS $R$ such that $R$ is not $\mathrm{UN}^=$ but b-flat $R$ is $\mathrm{UN}^=$.

A similar statement holds for $\mathrm{UN}^\rightarrow$. As it turns out, there are no counterexamples of type 1 for either $\mathrm{UN}^=$ or $\mathrm{UN}^\rightarrow$, because regular flattening destroys normal forms without creating any new ones. The following examples show how regular flattening destroys normal forms.

**Example 1.** Let $R = \{l \to g(f(a))\}$ where $l$ is any term. Then b-flat $R = \{l \to g(c),\ f(a) \to c,\ c \to f(a)\}$. The term $f(a)$ is a normal form of $R$ but is not a normal form of b-flat $R$.

**Example 2.** Let $R = \{h(f(a)) \to r\}$ for some term $r$. Then b-flat $R = \{h(c) \to r,\ f(a) \to c,\ c \to f(a)\}$. The term $f(a)$ is a normal form of $R$ but is not a normal form of b-flat $R$.

We state the following two theorems without proof, because their proofs are similar to the $\Leftarrow$ directions of proofs in Sections 3.2 and 3.3.

**Theorem 3.** *If $\langle \mathcal{T}, R\rangle$ is $\mathrm{UN}^=$, then $\langle \mathcal{T}_c, \text{b-flat}\,R\rangle$ is $\mathrm{UN}^=$.*

**Theorem 4.** *If $\langle \mathcal{T}, R\rangle$ is $\mathrm{UN}^\rightarrow$, then $\langle \mathcal{T}_c, \text{b-flat}\,R\rangle$ is $\mathrm{UN}^\rightarrow$.*

Thus we will try to show counterexamples of type 2. In fact, a single counterexample will suffice for both $\mathrm{UN}^=$ and $\mathrm{UN}^\rightarrow$. We use the fact that any rewrite system that is $\mathrm{UN}^=$ is also $\mathrm{UN}^\rightarrow$. Our counterexample is a TRS $R$ such that $R$ is not $\mathrm{UN}^\rightarrow$ but b-flat $R$ is $\mathrm{UN}^=$. Thus, $R$ is not $\mathrm{UN}^=$ but b-flat $R$ is $\mathrm{UN}^=$. Also, $R$ is not $\mathrm{UN}^\rightarrow$ but b-flat $R$ is $\mathrm{UN}^\rightarrow$. This implies that regular flattening preserves neither $\mathrm{UN}^=$ nor $\mathrm{UN}^\rightarrow$.

**Example 5.** Let $R = \{a \to f(f(b)),\ a \to c\}$. $R$ is not $\mathrm{UN}^\rightarrow$ because $f(f(b)) \neq c$. We now show that b-flat $R = \{a \to f(d),\ a \to c,\ d \to f(b),\ f(b) \to d\}$ is $\mathrm{UN}^=$. First we note that b-flat $R$ is a rewrite system over the set of terms $\mathcal{T}_c = \{f^n(t) \mid n \geq 0 \text{ and } t \in \{a, b, c\} \cup X\}$. Of these terms, the normal forms are $\{b\} \cup \{f^n(c) \mid n \geq 0\} \cup \{f^n(x) \mid n \geq 0 \text{ and } x \text{ is a variable}\}$. To show that b-flat $R$ is $\mathrm{UN}^=$, we need to show that each equivalence class of $\leftrightarrow^*_{\text{b-flat}\,R}$ contains at most one normal form. The equivalence class of $b$ is $\{b\}$, and for each $n$ and variable $x$ the equivalence class of $f^n(x)$ is $\{f^n(x)\}$. It remains to show that for each $n$ the equivalence class of $f^n(c)$ does not contain $f^m(c)$ for some $m \neq n$. The only derivation that exists from $f^n(c)$ is $f^n(c) \leftarrow_{\text{b-flat}\,R} f^n(a) \to_{\text{b-flat}\,R} f^{n+1}(d) \leftrightarrow_{\text{b-flat}\,R} f^{n+2}(b)$. Thus, the equivalence class of $f^n(c)$ is $\{f^n(c), f^n(a), f^{n+1}(d), f^{n+2}(b)\}$. Therefore b-flat $R$ is $\mathrm{UN}^=$.

### 3.2 Flattening on the Right

For this section, let $R = R_0 \cup \{l \to r[r_0]_p\}$ be a rewrite system over $\mathcal{T}$, where $R_0$ is a set of rules, $l$ and $r$ are terms, and $r_0$ is a non-constant flat ground term. The term $r[r_0]_p$ is just the right-hand side of the rule, and we can consider it to be just the term $r$ which has $r_0$ as a subterm at position $p$. We write the right-hand side of the rule as $r[r_0]_p$ because we will be replacing $r_0$ with a new constant $c$. Let r-flat $R = R_0 \cup \{l \to r[c]_p,\ c \to r_0\}$, where $c$ is a new constant not in $\mathcal{F}$. The position $p$ will be omitted from now on. Because $r_0$ is flat, the rule $c \to r_0$ is a flat rule.

**Proposition 6.** *For all $u, v \in \mathcal{T}_c$, if $u \to v$ in $\langle \mathcal{T}_c, \text{r-flat}\,R\rangle$ then $u[c \Rightarrow r_0] \to^* v[c \Rightarrow r_0]$ in $\langle \mathcal{T}, R\rangle$.*

*Proof.* There are three cases. First, if $u \to_{R_0} v$, then $u[c \Rightarrow r_0] \to_{R_0} v[c \Rightarrow r_0]$. Second, if $u \to_{l \to r[c]} v$, then $u[c \Rightarrow r_0] \to_{l \to r[r_0]} v[c \Rightarrow r_0]$. Finally, if $u \to_{c \to r_0} v$, then $u[c \Rightarrow r_0] = v[c \Rightarrow r_0]$. ∎

**Theorem 7 (Flattening on the right preserves $\mathrm{UN}^=$).** *$\langle \mathcal{T}_c, \text{r-flat}\,R\rangle$ is $\mathrm{UN}^=$ if and only if $\langle \mathcal{T}, R\rangle$ is $\mathrm{UN}^=$.*

*Proof.* ⇒: Assume $\langle \mathcal{T}_c, \text{r-flat}\, R\rangle$ is $\text{UN}^=$. Pick normal forms $n_1$ and $n_2$ of $\langle \mathcal{T}, R\rangle$ such that $n_1 \leftrightarrow^* n_2$ in $\langle \mathcal{T}, R\rangle$ and show that $n_1 = n_2$. The left-hand sides of the rules are the same in both $R$ and r-flat $R$, so $n_1$ and $n_2$ must be normal forms of $\langle \mathcal{T}_c, \text{r-flat}\, R\rangle$. We need to show that $n_1 \leftrightarrow^* n_2$ in $\langle \mathcal{T}_c, \text{r-flat}\, R\rangle$. This follows from the fact that for terms $u, v \in \mathcal{T}$ if $u \to_{l\to r[r_0]} v$, then there is a term $w \in \mathcal{T}_c$ such that $u \to_{l\to r[c]} w \to_{c\to r_0} v$. Thus $n_1 = n_2$ because $\langle \mathcal{T}_c, \text{r-flat}\, R\rangle$ is $\text{UN}^=$.

⇐: Assume $\langle \mathcal{T}, R\rangle$ is $\text{UN}^=$. Pick normal forms $n_1$ and $n_2$ of $\langle \mathcal{T}_c, \text{r-flat}\, R\rangle$ such that $n_1 \leftrightarrow^* n_2$ in $\langle \mathcal{T}_c, \text{r-flat}\, R\rangle$ and show that $n_1 = n_2$. Because of the $c \to r_0$ rule, both $n_1$ and $n_2$ must be in $\mathcal{T}$. Again, the left-hand sides of the rules are the same in both $R$ and r-flat $R$, so $n_1$ and $n_2$ must be normal forms of $\langle \mathcal{T}, R\rangle$. We need to show that $n_1 \leftrightarrow^* n_2$ in $\langle \mathcal{T}, R\rangle$. This follows from Proposition 6 because $n_1[c \Rightarrow r_0] = n_1$ and $n_2[c \Rightarrow r_0] = n_2$. Thus $n_1 = n_2$ because $\langle \mathcal{T}, R\rangle$ is $\text{UN}^=$. ■

**Theorem 8 (Flattening on the right preserves $\text{UN}^\to$).** *$\langle \mathcal{T}_c, \text{r-flat}\, R\rangle$ is $\text{UN}^\to$ if and only if $\langle \mathcal{T}, R\rangle$ is $\text{UN}^\to$.*

*Proof.* Similar to Theorem 7, but note that in the ⇐ direction, we have a term $s$ in $\mathcal{T}_c$ such that $n_1 \leftarrow^* s \to^* n_2$ in $\langle \mathcal{T}_c, \text{r-flat}\, R\rangle$. We can then show that $n_1 \leftarrow^* s[c \Rightarrow r_0] \to^* n_2$ in $\langle \mathcal{T}, R\rangle$ using Proposition 6. ■

For the TRS $R$ described in Example 1, we have r-flat $R = \{l \to g(c),\, c \to f(a)\}$. The term $f(a)$, which is a normal form of $R$ is not a normal form of b-flat $R$, but remains a normal form of r-flat $R$.

## 3.3 Flattening on the Left

For this section, let $R$ be a rewrite system over $\mathcal{T}$ and let $l_0$ be a non-constant flat ground term that is a subterm of a left-hand side of a rule of $R$. Define l-flat $R$ by

$$\text{l-flat}\, R = \begin{cases} R^l \cup \{l_0 \to c\} & \text{if } l_0 \text{ is a normal form of } \langle \mathcal{T}, R\rangle, \\ R^l \cup \{l_0 \to c,\, c \to c\} & \text{otherwise,} \end{cases}$$

where $c$ is a new constant not in $\mathcal{F}$, and $R^l$ is $R$ with every occurrence of $l_0$ in the left-hand sides of the rules of $R$ replaced by $c$. The position $p$ will be omitted from now on. Because $l_0$ is flat, the rule $l_0 \to c$ is a flat rule.

**Proposition 9.** *For all $u, v \in \mathcal{T}_c$, if $u \to v$ in $\langle \mathcal{T}_c, \text{l-flat}\, R\rangle$ then $u[c \Rightarrow l_0] \to^* v[c \Rightarrow l_0]$ in $\langle \mathcal{T}, R\rangle$.*

*Proof.* There are three cases. First, if $u \to_{R^l} v$, then $u[c \Rightarrow l_0] \to_R v[c \Rightarrow l_0]$. Second, if $u \to_{l_0\to c} v$, then $u[c \Rightarrow l_0] = v[c \Rightarrow l_0]$. Finally, if $u \to_{c\to c} v$, then $u[c \Rightarrow l_0] = v[c \Rightarrow l_0]$. ■

**Theorem 10 (Flattening on the left preserves $\text{UN}^=$).** *$\langle \mathcal{T}_c, \text{l-flat}\, R\rangle$ is $\text{UN}^=$ if and only if $\langle \mathcal{T}, R\rangle$ is $\text{UN}^=$.*

*Proof.* ⇒: Assume $\langle \mathcal{T}_c, \text{l-flat}\, R\rangle$ is $\text{UN}^=$. Pick normal forms $n_1$ and $n_2$ of $\langle \mathcal{T}, R\rangle$ such that $n_1 \leftrightarrow^* n_2$ in $\langle \mathcal{T}, R\rangle$ and show that $n_1 = n_2$. We know that $n_1 \leftrightarrow^* n_2$ in $\langle \mathcal{T}_c, \text{l-flat}\, R\rangle$, because for terms $u, v \in \mathcal{T}$, if $u \to_R v$, then there is a term $w \in \mathcal{T}_c$ such that $u \to^*_{l_0\to c} w \to_{R^l} v$. If $n_1$ and $n_2$ are normal forms of $\langle \mathcal{T}_c, \text{l-flat}\, R\rangle$, then $n_1 = n_2$. So assume at least one of $n_1$ and $n_2$ is not a normal form of $\langle \mathcal{T}_c, \text{l-flat}\, R\rangle$. This means that one of the terms has an occurrence of $l_0$ and $l_0$ is a normal form of $\langle \mathcal{T}, R\rangle$. Then the terms $n_1[l_0 \Rightarrow c]$ and $n_2[l_0 \Rightarrow c]$ are normal forms of $\langle \mathcal{T}_c, \text{l-flat}\, R\rangle$, and we have $n_1[l_0 \Rightarrow c] \leftarrow^*_{l_0\to c} n_1 \leftrightarrow^* n_2 \to^*_{l_0\to c} n_2[l_0 \Rightarrow c]$ in $\langle \mathcal{T}_c, \text{l-flat}\, R\rangle$. This means that $n_1[l_0 \Rightarrow c] = n_2[l_0 \Rightarrow c]$, which implies that $n_1 = n_2$.

⇐: Assume $\langle \mathcal{T}, R\rangle$ is $\text{UN}^=$. Pick normal forms $n_1$ and $n_2$ of $\langle \mathcal{T}_c, \text{l-flat}\, R\rangle$ such that $n_1 \leftrightarrow^* n_2$ in $\langle \mathcal{T}_c, \text{l-flat}\, R\rangle$ and show that $n_1 = n_2$.

We claim that $n_1[c \Rightarrow l_0]$ is a normal form of $\langle \mathcal{T}, R\rangle$. For, if it is not, then there is a rule $l \to r \in R$ such that $n_1[c \Rightarrow l_0]$ has a subterm $l\sigma$ for some substitution $\sigma$. But then $n_1$ would have a subterm $(l[l_0 \Rightarrow c])\tau$ where $\tau$ is defined by $x\tau = (x\sigma)[l_0 \Rightarrow c]$. And, since $l[l_0 \Rightarrow c] \to r \in R^l$, this would make $n_1$ reducible in $\langle \mathcal{T}_c, \text{l-flat}\, R\rangle$, which is a contradiction. Similarly, $n_2[c \Rightarrow l_0]$ is a normal form of $\langle \mathcal{T}, R\rangle$.

Proposition 9 implies that $n_1[c \Rightarrow l_0] \leftrightarrow^* n_2[c \Rightarrow l_0]$ in $\langle \mathcal{T}, R\rangle$. Because $\langle \mathcal{T}, R\rangle$ is $\text{UN}^=$, we know that $n_1[c \Rightarrow l_0] = n_2[c \Rightarrow l_0]$. This implies that $n_1 = n_2$. ■

**Theorem 11 (Flattening on the left preserves $\mathrm{UN}^{\rightarrow}$).** *$\langle \mathcal{T}_c, \text{l-flat}\, R\rangle$ is $\mathrm{UN}^{\rightarrow}$ if and only if $\langle \mathcal{T}, R\rangle$ is $\mathrm{UN}^{\rightarrow}$.*

*Proof.* Similar to Theorem 10, but note that in the $\Leftarrow$ direction, we have a term $s$ in $\mathcal{T}_c$ such that $n_1 \leftarrow^* s \rightarrow^* n_2$ in $\langle \mathcal{T}_c, \text{l-flat}\, R\rangle$. We can then show that $n_1[c \Rightarrow l_0] \leftarrow^* s[c \Rightarrow l_0] \rightarrow^* n_2[c \Rightarrow l_0]$ in $\langle \mathcal{T}, R\rangle$ using Proposition 9. ■

For the TRS $R$ described in Example 2, we have l-flat $R = \{h(c) \rightarrow r,\ f(a) \rightarrow c\}$. The term $f(a)$, which is a normal form of $R$ is not a normal form of b-flat $R$, nor is it a normal form of l-flat $R$. However, it is uniquely replaced by the normal form $c$ of l-flat $R$.

# 4 Witnesses to Non-$\mathrm{UN}^=$

We want to show that if $R$ is a linear flat TRS and $R$ is not $\mathrm{UN}^=$ then there is a witness $\langle n, m\rangle$ to non-$\mathrm{UN}^=$ such that $n$ and $m$ are ground, there is a flat ground term $t$ and ground derivation $n \leftrightarrow_R^* t \leftrightarrow_R^* m$, and $t$ is an instance of a left-hand side of a rule of $R$.

A *witness to non-*$\mathrm{UN}^=$ is a pair $\langle n, m\rangle$ of normal forms such that $n \leftrightarrow_R^* m$ and $n \neq m$. The *size* of a witness $\langle n, m\rangle$ is $|n| + |m|$. A *ground witness to non-*$\mathrm{UN}^=$ is a pair $\langle n, m\rangle$ of ground normal forms such that there is a ground derivation of $n \leftrightarrow_R^* m$ and $n \neq m$.

For any rewrite system $R$, if $R$ is not $\mathrm{UN}^=$, then there will be a witness to non-$\mathrm{UN}^=$. And, if there are witnesses to non-$\mathrm{UN}^=$, then we can examine witnesses to non-$\mathrm{UN}^=$ that are minimal in size. Our first step will be to show that if $\langle n, m\rangle$ is a minimal witness to non-$\mathrm{UN}^=$, then in any derivation of $n \leftrightarrow_R^* m$ there is a term $t$ such that $n \leftrightarrow_R^* t \leftrightarrow_R^* m$, and $t$ is an instance (not necessarily flat or ground) of a left-hand side of a rule of $R$. This is accomplished by the following proposition.

**Proposition 12.** *If $R$ is any rewrite system and $\langle n, m\rangle$ is a minimal witness to non-*$\mathrm{UN}^=$*, then there must be a root rewrite step in any derivation of $n \leftrightarrow_R^* m$.*

*Proof.* If there is a derivation from $n$ to $m$ with no root rewrite step, then there is a $k$-ary function symbol $f$ such that $n = f(n_1, \ldots, n_k)$ and $m = f(m_1, \ldots, m_k)$ and $n_i \leftrightarrow_R^* m_i$ for each $i$ and there is some $j$ for which $n_j \neq m_j$. Because $n_j$ and $m_j$ are both normal forms, $\langle n_j, m_j\rangle$ constitutes a witness to non-$\mathrm{UN}^=$ that is smaller than $\langle n, m\rangle$, contradicting the minimality of $\langle n, m\rangle$. Thus there must be a root rewrite step in any derivation of $n \leftrightarrow_R^* m$. ■

For the rest of this section, let $R$ be a linear flat rewrite system, and let $\langle n, m\rangle$ be a minimal witness to non-$\mathrm{UN}^=$ of $R$. In any derivation of $n \leftrightarrow_R^* m$, Proposition 12 says that there is a left-hand side $l$ of a rule of $R$ and substitution $\sigma$ such that $n \leftrightarrow_R^* l\sigma \leftrightarrow_R^* m$. Now we shall show that there is a substitution $\sigma'$ such that $l\sigma'$ is flat (but not necessarily ground) and $n \leftrightarrow_R^* l\sigma' \leftrightarrow_R^* m$. Assume that $l\sigma$ is a non-flat term, so $l\sigma = f(s_1, \ldots, s_k)$ for some $k$-ary function symbol $f$ and terms $s_1, \ldots, s_k$, where at least one $s_i$ is not a height-zero term. Because $l$ is flat and no left-hand side of a rule is a variable (see the Preliminaries section), we know that $l = f(l_1, \ldots, l_k)$ where $l_1, \ldots, l_k$ are constants and variables. Also because $l$ is flat, if $s_i$ is not a height-zero term then $l_i$ must be a variable. We shall show how to replace a non height-zero $s_i$ with a height-zero term to create an instance $l\sigma'$ of $l$ that is flat. Let $s_i$ be a non-height-zero term. We exploit the fact that $s_i$ is either equivalent to some height-zero term, or it is not.

First we deal with the case when $s_i$ is equivalent to a height-zero term $a$.

**Proposition 13.** *If $s_i$ is equivalent to a height-zero term $a$, then $n \leftrightarrow_R^* l\sigma' \leftrightarrow_R^* m$, where $\sigma'$ is defined by $x\sigma' = a$ if $x = l_i$ and $x\sigma' = x\sigma$ if $x \neq l_i$.*

*Proof.* By Proposition 12, we can assume the derivation looks like $n \leftrightarrow_R^* l\sigma \xrightarrow{\text{r}} r\sigma \leftrightarrow_R^* m$, where $l \rightarrow r \in R$. Because $s_i \leftrightarrow_R^* a$, we have both $l\sigma \leftrightarrow_R^* l\sigma'$ and $r\sigma' \leftrightarrow_R^* r\sigma$. By linearity of $l \rightarrow r$ we get $l\sigma' \xrightarrow{\text{r}} r\sigma'$. This yields the claim as shown in Figure 1. In fact, linearity is not required since we can rewrite multiple occurrences of $s_i$ in parallel to multiple occurrences of $a$. ■

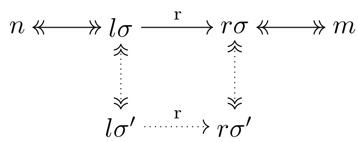


Figure 1: Replacing non-height-zero subterms with equivalent height-zero subterms

We can use Proposition 13 to replace any non height-zero $s_i$ that is equivalent to a height-zero term $a$ by $a$.

Now we deal with the case when $s_i$ is not equivalent to any height-zero term. By following what happens to $s_i$ in a particular derivation of $n \leftrightarrow_R^* l\sigma \leftrightarrow_R^* m$, we will see that we can replace $s_i$ with a new variable $y$. Then we will have a derivation of $n \leftrightarrow_R^* l\sigma' \leftrightarrow_R^* m$ (for the same $n$ and $m$), where $\sigma'$ is defined by $x\sigma' = y$ if $x = l_i$ and $x\sigma' = x\sigma$ if $x \neq l_i$. The key to this result is that for any term $s'_i$ that is $R$-equivalent to $s_i$, $s'_i$ does not have a proper overlap with the pattern of any side of a rule, since sides of rules are flat and $s'_i$ is not height-zero.

**Definition 14 (The Cousin Relation).** Let $v_0 \leftrightarrow_R \cdots \leftrightarrow_R v_k$ be a derivation, and let $v_0 = U_0[u_0]$ for some context $U_0[\,]$ and term $u_0$. We define *cousins* of $u_0$ in the derivation as follows. First, $u_0$ is a cousin of itself. Second, if $v_j = U_j[u_j]$, and $u_j$ is a cousin of $u_0$, and $v_{j+1} = U_{j+1}[u_{j+1}]$, then $u_{j+1}$ is a cousin of $u_0$ if any of the following cases hold:

1. The rewrite occurs at or below the position of $u_j$, in which case $U_j = U_{j+1}$.
2. The rewrite occurs parallel to the position of $u_j$, in which case $u_j = u_{j+1}$ and the position of $u_j$ in $U_j$ is the same as the position of $u_{j+1}$ in $U_{j+1}$.
3. The rewrite occurs above the position of $u_j$, and $u_j$ is at position $pp_{x_1}q$, and $u_{j+1}$ is at position $pp_{x_2}q$, where
   (a) $p$ is the position of the rewrite,
   (b) $p_{x_1}$ is the position of an occurrence of a variable $x$ in one side of the rewrite rule,
   (c) $p_{x_2}$ is the position of an occurrence of $x$ in the other side of the rewrite rule,
   (d) $q$ is the position of $u_j$ in $x\tau$, where $\tau$ is the substitution of the rewrite.

A couple of things are clear about cousins. First, cousins are $R$-equivalent. Second, since $R$ is linear, there can be at most one cousin of $u_0$ in each term of the derivation.

**Proposition 15.** *Let $U[u] \leftrightarrow_R^* V[v]$ where $v$ is a cousin of $u$. Then $U[w] \leftrightarrow_R^* V[w]$ for any term $w$.*

**Proposition 16.** *Let $u = U_0[u_0]$ be a term where $u_0$ is not equivalent to a height-zero term, and let $v$ be a term such that $u \leftrightarrow_R^* v$. If $v$ does not contain a cousin of $u_0$, then $U_0[w] \leftrightarrow_R^* v$ for any term $w$.*

*Proof.* In a derivation of $u \leftrightarrow_R^* v$ there are terms $u'$ and $v'$ such that $u \leftrightarrow_R^* u' \leftrightarrow_R v' \leftrightarrow_R^* v$ where every term in $u \leftrightarrow_R^* u'$ has a cousin of $u_0$ while no term in $v' \leftrightarrow_R^* v$ has a cousin of $u_0$. Let $u' = U'[u'_0]$ where $u'_0$ is the cousin of $u_0$. We know that $U_0[w] \leftrightarrow_R^* U'[w]$. We need to show that $U'[w] \leftrightarrow_R v'$. In the rewrite step $U'[u'_0] \leftrightarrow_R v'$, the rewrite can occur at, below, parallel to, or above the position of $u'_0$. If it occurs at, below, or parallel to the position of $u'_0$, then $v'$ would have a cousin of $u_0$. Therefore, the rewrite occurs above $u'_0$. Because $u'_0$ is not a height-zero term and $R$ is flat, $u'_0$ must lie underneath a variable, say $x$, in the substitution part of the rule application. Because $u'_0$ does not occur in $v'$, the other side of the rule does not contain $x$. Therefore, combined with the linearity of $R$, we have $U'[w] \leftrightarrow_R v'$. ■

**Lemma 17.** *If $s_i$ is not equivalent to any height-zero term, then $n \leftrightarrow_R^* l\sigma' \leftrightarrow_R^* m$, where $\sigma'$ is defined by $l_i\sigma' = y$ for a new variable $y$ and $x\sigma' = x\sigma$ if $x \neq l_i$.*

*Proof.* Pick a derivation of $n \leftrightarrow_R^* l\sigma \leftrightarrow_R^* m$ that is as short as possible. There are three cases to consider.

In the first case neither $n$ nor $m$ contains a cousin of $s_i$. Then by Proposition 16, $n \leftrightarrow_R^* l\sigma' \leftrightarrow_R^* m$.

In the second case, one of the normal forms contains a cousin of $s_i$ but the other does not. Without loss of generality, let $n$ contain no cousin of $s_i$ and let $m = M[m_0]$ where $m_0$ is a cousin of $s_i$. Then $n \leftrightarrow_R^* l\sigma' \leftrightarrow_R^* M[y]$ by Propositions 15 and 16, so $\langle n, M[y]\rangle$ is a smaller witness to non-UN$^=$ than $\langle n, m\rangle$ (note that $n \neq M[y]$ since $y$ is a new variable). Thus the second case cannot happen.

In the third case, $s_i$ has both a cousin $n_0$ in $n$ and a cousin $m_0$ in $m$, so that $n = N[n_0]$ and $m = M[m_0]$. If $N[\,] \neq M[\,]$ then $N[y] \neq M[y]$. Since $N[y] \leftrightarrow_R^* l\sigma' \leftrightarrow_R^* M[y]$ by Proposition 15, then $\langle N[y], M[y]\rangle$ is a smaller witness to non-UN$^=$ than $\langle n, m\rangle$ (note that $n_0$ and $m_0$ can not be height-zero because they are equivalent to $s_i$). If $N[\,] = M[\,]$ then $n_0 \neq m_0$ so $\langle n_0, m_0\rangle$ is a witness to non-UN$^=$. It is a smaller witness to non-UN$^=$ than $\langle n, m\rangle$, because if $\langle n_0, m_0\rangle = \langle n, m\rangle$ then we have a derivation of $n \leftrightarrow_R^* s_i \leftrightarrow_R^* m$ that is shorter than the derivation of $n \leftrightarrow_R^* l\sigma \leftrightarrow_R^* m$, which we picked to be as short as possible. Thus, the third case cannot happen either. ∎

We can use Lemma 17 to replace all non height-zero $s_i$ that are not equivalent to height-zero terms by new variables, since replacing such an $s_i$ does not increase the length of the derivation of $n \leftrightarrow_R^* l\sigma' \leftrightarrow_R^* m$. In fact, we can now assume that $l\sigma$ does not contain any non height-zero $s_i$ that are not equivalent to height-zero terms.

Thus, we have proved Theorem 18.

**Theorem 18.** *If $R$ is a linear flat TRS that is not* UN$^=$*, then there is a witness $\langle n, m\rangle$ to non-*UN$^=$ *with a derivation of $n \leftrightarrow_R^* t \leftrightarrow_R^* m$, where $t$ is a flat instance of a left-hand side of a rule of $R$.*

## 4.1 Ground Witnesses

So far, we have shown that if $\langle \mathcal{T}, R\rangle$ is a linear flat TRS that is not UN$^=$, then there is a witness $\langle n, m\rangle$ to non-UN$^=$ with a derivation of $n \leftrightarrow_R^* t \leftrightarrow_R^* m$, where $t$ is a flat instance of a left-hand side of a rule of $R$. In this section we shall show that for any linear flat TRS $\langle \mathcal{T}, R\rangle$, we can add a finite number of constants to the signature of $\mathcal{T}$ to get a rewrite system $\langle \mathcal{T}', R\rangle$ that is UN$^=$ if and only if $\langle \mathcal{T}, R\rangle$ is UN$^=$, and if they are not UN$^=$, then $\langle \mathcal{T}', R\rangle$ has a ground witness $\langle n, m\rangle$ to non-UN$^=$ with a ground derivation $n \leftrightarrow_R^* t \leftrightarrow_R^* m$, where $t$ is a flat ground instance of a left-hand side of a rule of $R$.

**Proposition 19.** *Let $\mathcal{T}$ be the set of terms over a signature $\mathcal{F}$, and let $\mathcal{T}_c$ be the set of terms over the signature $\mathcal{F} \cup \{c\}$, where $c$ is a new constant not in $\mathcal{F}$. A TRS $\langle \mathcal{T}, R\rangle$ is* UN$^=$ *if and only if $\langle \mathcal{T}_c, R\rangle$ is* UN$^=$.

*Proof.* The $\Leftarrow$ direction is obvious. For the $\Rightarrow$ direction, assume $\langle \mathcal{T}, R\rangle$ is UN$^=$. To show $\langle \mathcal{T}_c, R\rangle$ is UN$^=$, pick normal forms $n, m \in \mathcal{T}_c$ such that $n \leftrightarrow_R^* m$ in $\mathcal{T}_c$. In any derivation of $n \leftrightarrow_R^* m$, we can replace all occurrences of $c$ with a new variable $x$ to get a derivation of $\bar{n} \leftrightarrow_R^* \bar{m}$ in $\mathcal{T}$. Since $\bar{n}$ and $\bar{m}$ are normal forms of $\langle \mathcal{T}, R\rangle$, we have $\bar{n} = \bar{m}$. This implies $n = m$. ∎

**Lemma 20.** *Let $\langle \mathcal{T}, R\rangle$ be a linear flat TRS. Then we can add a finite number of constants to the signature of $\mathcal{T}$ to get a rewrite system $\langle \mathcal{T}', R\rangle$ that is* UN$^=$ *if and only if $\langle \mathcal{T}, R\rangle$ is* UN$^=$*, and if they are not* UN$^=$*, then $\langle \mathcal{T}', R\rangle$ has a ground witness $\langle n, m\rangle$ to non-*UN$^=$ *with a ground derivation $n \leftrightarrow_R^* t \leftrightarrow_R^* m$, where $t$ is a flat ground instance of a left-hand side of a rule of $R$.*

*Proof.* Let $\mathcal{T}$ be the set of terms over the signature $\mathcal{F}$. Let $\alpha$ be the maximum arity of the function symbols in $\mathcal{F}$. Let $\mathcal{T}'$ be the set of terms over the signature $\mathcal{F} \cup \{c_1, \ldots, c_{3\alpha}\}$, where $c_1, \ldots, c_{3\alpha}$ are distinct new constants not in $\mathcal{F}$. Proposition 19 ensures that $\langle \mathcal{T}', R\rangle$ is UN$^=$ if and only if $\langle \mathcal{T}, R\rangle$ is UN$^=$.

Assume $\langle \mathcal{T}', R\rangle$ and $\langle \mathcal{T}, R\rangle$ are not UN$^=$. Let $\langle n, m\rangle$ be a minimal witness to non-UN$^=$ of $\langle \mathcal{T}, R\rangle$. By Theorem 18, there is an instance $l\sigma$ of a left-hand side $l$ of a rule such that $l\sigma$ is flat and $n \leftrightarrow_R^* l\sigma \leftrightarrow_R^* m$. Pick a derivation of $n \leftrightarrow_R^* l\sigma \leftrightarrow_R^* m$. Let $x_1, \ldots, x_j$ be the variables that occur in $n$, $l\sigma$, and $m$, and let $x_{j+1}, \ldots, x_k$ be the rest of the variables in the derivation. Note that $j \leq 3\alpha$.

Define a substitution $\tau$ by $x_i\tau = c_i$ if $i \leq j$ and $x_i\tau$ is any ground term otherwise. We can apply $\tau$ to each term in the derivation to get a ground derivation of $n\tau \leftrightarrow_R^* l\sigma\tau \leftrightarrow_R^* m\tau$. Then $\langle n\tau, m\tau\rangle$ is a ground witness to non-UN$^=$ of $\langle \mathcal{T}', R\rangle$. Note that $n\tau \neq m\tau$ since different variables in $n$ and $m$ are mapped to different constants. Additionally, $l\sigma\tau$ is flat and ground. ∎

# 5 Tree Automata for Reducible Terms

Let $R$ be a left-linear TRS over the set $\mathcal{T}$ of terms. We want a tree automaton $A_{\mathrm{NF}(R)}$ that recognizes the language of ground normal forms of $R$ over $\mathcal{T}$. To ensure a polynomial time algorithm for UN$^=$, we need $A_{\mathrm{NF}(R)}$ to be polynomial in the size of $R$, but we don't require determinism. It is easier to construct a tree automaton $A_{\mathrm{Red}(R)}$ that recognizes the complement language—the language of $R$-reducible terms. Once we have $A_{\mathrm{Red}(R)}$, we can construct $A_{\mathrm{NF}(R)}$ as follows. If $A_{\mathrm{Red}(R)}$ is deterministic, then we define $A_{\mathrm{NF}(R)}$ to be like $A_{\mathrm{Red}(R)}$ but with a complemented set of final states. If $A_{\mathrm{Red}(R)}$ is nondeterministic, then we may use [4] to find a deterministic automaton $A'_{\mathrm{Red}(R)}$ such that $\mathcal{L}(A'_{\mathrm{Red}(R)}) = \mathcal{L}(A_{\mathrm{Red}(R)})$. Then we could define $A_{\mathrm{NF}(R)}$ to be like $A'_{\mathrm{Red}(R)}$ but with a complemented set of final states, as before. Unfortunately, $A'_{\mathrm{Red}(R)}$ may be exponential in the size of $A_{\mathrm{Red}(R)}$.

We give two automata that accept the language of $R$-reducible terms. The first is a simple nondeterministic automaton that is described in Appendix A. The second is a more complicated deterministic automaton $A_{\mathrm{Red}(R)}$ that is polynomial in the size of $R$ provided the maximum arity of the function symbols is bounded or the signature is fixed.

Comon already does something similar in [3]. This section clarifies some points and provides a complexity analysis specific for our purposes with linear flat rewrite systems.

## 5.1 Deterministic Tree Automata for Reducible Terms

The next couple of examples show how to deal with instances and nondeterminism. We give only the definition of the automata here. That they in fact accept the $R$-reducible terms is shown by Theorem 26.

Similarly to the nondeterministic case, the examples will have one state $q_t$ per subterm $t$ of a left-hand side of a rule that is not an instance of a left-hand side of a rule. To be complete and deterministic, we need to have for each function symbol $f$ and states $q_1, \ldots, q_n$ exactly one state $q$ such that $f(q_1, \ldots, q_n) \to q$ is a transition rule. If any of the $q_i$ equals $q_r$, then $q$ should be $q_r$. If no $q_i$ is $q_r$, then $f(q_1, \ldots, q_n) = f(q_{t_1}, \ldots, q_{t_n})$ for some terms $t_1, \ldots, t_n$. If $f(t_1, \ldots, t_n)$ is an instance of a left-hand side of a rule, then we want $q$ to be $q_r$. If $f(t_1, \ldots, t_n)$ is an instance of a term $t$ for which there is a state $q_t$, then we may want $q$ to be $q_t$. This leads to a complete deterministic automaton in some cases, for instance in the next example.

**Example 21.** Let $R = \{g(f(x), y) \to f(a)\}$. We shall define a complete deterministic automaton $A$ to recognize $R$-reducible terms. Define the set $Q$ of states by $Q = \{q_x, q_{f(x)}, q_r\}$ and the set $Q_f$ of final states by $Q_f = \{q_r\}$. Define the set $\Delta$ of transition rules by

$$\begin{array}{rrl}
a \to q_x & g(q_r, q) \to q_r & \forall q \in Q \\
f(q_x) \to q_{f(x)} & g(q, q_r) \to q_r & \forall q \in Q \\
f(q_{f(x)}) \to q_{f(x)} & g(q_{f(x)}, q) \to q_r & \forall q \in Q \\
f(q_r) \to q_r & g(q_1, q_2) \to q_x & \text{if } q_1 \neq q_r \text{ and } q_2 \neq q_r \text{ and } q_1 \neq q_{f(x)}
\end{array}$$

When there are two states $q_t$ and $q_{t'}$ such that $f(t_1, \ldots, t_n)$ is an instance of $t$ and $t'$, then there can be conflicts. If $t$ is an instance of $t'$, then we want $q$ to be $q_{t'}$. We implicitly did this in the last example, since $f(a)$ is an instance of both $f(x)$ and $x$, and we chose transition rule $f(q_x) \to q_{f(x)}$. However, when it is not the case that one of $t$ and $t'$ is an instance of the other, then we have a problem. The solution is to add a new state to $Q$. The next example shows how to do this.

**Example 22.** Let $R = \{g(f(a, x)) \to g(a),\ g(f(x, a)) \to f(a, x)\}$. If we try to construct an automaton as in the previous example, then we will need a transition rule with left-hand side $f(q_a, q_a)$. Since $f(a, a)$ is an instance of both $f(a, x)$ and $f(x, a)$, we would end up with two transition rules $f(q_a, q_a) \to q_{f(a,x)}$ and

$f(q_a, q_a) \to q_{f(x,a)}$ since $f(a, x)$ is not an instance of $f(x, a)$ and vice versa. This would make the automaton nondeterministic. To compensate, we introduce an extra state $q_{f(a,a)}$, where $f(a, a)$ is the most general instance of $f(a, x)$ and $f(y, a)$. Then the transition rule is $f(q_a, q_a) \to q_{f(a,a)}$ (marked (∗) below).

Define the set $Q$ of states by $Q = \{q_x, q_a, q_{f(a,x)}, q_{f(x,a)}, q_{f(a,a)}, q_r\}$ and set $Q_f$ of final states by $Q_f = \{q_r\}$. Define the set $\Delta$ of transition rules by

$$
\begin{array}{rlr}
a \to q_a & & \\
f(q_x, q_x) \to q_x & & f(q_{f(a,x)}, q_x) \to q_x \\
f(q_x, q_a) \to q_{f(x,a)} & & f(q_{f(a,x)}, q_a) \to q_{f(x,a)} \\
f(q_x, q_{f(a,x)}) \to q_x & & f(q_{f(a,x)}, q_{f(a,x)}) \to q_x \\
f(q_x, q_{f(x,a)}) \to q_x & & f(q_{f(a,x)}, q_{f(x,a)}) \to q_x \\
f(q_x, q_{f(a,a)}) \to q_x & & f(q_{f(a,x)}, q_{f(a,a)}) \to q_x \\
f(q_a, q_x) \to q_{f(a,x)} & & f(q_{f(a,a)}, q_x) \to q_x \\
f(q_a, q_a) \to q_{f(a,a)} & (*) & f(q_{f(a,a)}, q_a) \to q_{f(x,a)} \\
f(q_a, q_{f(a,x)}) \to q_{f(a,x)} & & f(q_{f(a,a)}, q_{f(a,x)}) \to q_x \\
f(q_a, q_{f(x,a)}) \to q_{f(a,x)} & & f(q_{f(a,a)}, q_{f(x,a)}) \to q_x \\
f(q_a, q_{f(a,a)}) \to q_{f(a,x)} & & f(q_{f(a,a)}, q_{f(a,a)}) \to q_x \\
f(q_{f(x,a)}, q_x) \to q_x & & g(q_x) \to q_x \\
f(q_{f(x,a)}, q_a) \to q_{f(x,a)} & & g(q_a) \to q_x \\
f(q_{f(x,a)}, q_{f(a,x)}) \to q_x & & g(q_{f(a,x)}) \to q_r \\
f(q_{f(x,a)}, q_{f(x,a)}) \to q_x & & g(q_{f(x,a)}) \to q_r \\
f(q_{f(x,a)}, q_{f(a,a)}) \to q_x & & g(q_{f(a,a)}) \to q_r
\end{array}
$$

plus

$$
\begin{array}{l}
f(q_1, q_2) \to q_r \quad \text{if } q_1 = q_r \text{ or } q_2 = q_r \\
g(q_r) \to q_r \, .
\end{array}
$$

Now, given a left-linear TRS $R$, we show how (based on [3]) to construct a deterministic bottom-up tree automaton $A_{\mathrm{Red}(R)}$ that accepts the language of terms reducible by $R$. Verma [12] covers the case when $R$ is ground, so here we assume that there is a non-ground left-hand side of a rewrite rule. Also remember that no left-hand side of a rule of $R$ is a variable. Therefore, some variable occurs as a strict subterm of a left-hand side of a rewrite rule.

We define a partial operation $\Downarrow$ on terms by defining $s \Downarrow t$ to be a most general instance of $s$ and $t'$ if $s$ and $t'$ are unifiable, where $t'$ is the term $t$ with variables renamed so that $t'$ does not share any variables with $s$. The term $s \Downarrow t$ is unique up to variable renaming, in the sense that if $u = s \Downarrow t$ and $v = s \Downarrow t$ then $u$ is a renaming of $v$. If $s$ and $t$ are linear, then $s \Downarrow t$ is linear (when it exists). If a term $u$ is an instance of a term $t_1$ and $u$ is also an instance of another term $t_2$, then $u$ is an instance of $t_1 \Downarrow t_2$. Additionally, the binary operation $\Downarrow$ is associative, commutative, and idempotent. That is, for any terms $s$, $t$, and $u$ we have the following properties: $s \Downarrow t = t \Downarrow s$, $(s \Downarrow t) \Downarrow u = s \Downarrow (t \Downarrow u)$, and $s \Downarrow s = s$.

Let $S_0(R)$ be the set of subterms of left-hand sides of rules of $R$ that are not instances of left-hand sides of $R$. Note that $x \in S_0(R)$. Also, the size of $S_0(R)$ is a polynomial in the size of $R$. Let $S_1(R)$ be the smallest set containing $S_0(R)$ that is closed under the inference rule

$$
\frac{s, t \in S_1(R)}{s \Downarrow t \in S_1(R) \quad \text{if } s \Downarrow t \text{ exists}} \, .
$$

For example, there may be a term $t \in S_1(R)$ that we can write as $t = (t_1 \Downarrow t_2) \Downarrow ((t_3 \Downarrow t_2) \Downarrow t_4)$ where $t_1$, $t_2$, $t_3$, $t_4 \in S_0(R)$. Due to the associativity, commutativity, and idempotency of $\Downarrow$, we can transform this into $t = (((t_1 \Downarrow t_2) \Downarrow t_3) \Downarrow t_4)$. In fact, any term $t \in S_1(R)$ can be written as $t = t_1 \Downarrow t_2 \Downarrow \cdots \Downarrow t_n$ where $t_i = t_j$ only if $i = j$ for a set $\{t_1, t_2, \ldots, t_n\} \subseteq S_0(R)$. Thus there is at most one term in $S_1(R)$ for each subset of $S_0(R)$, which gives us the inequality $|S_1(R)| \leq 2^{|S_0(R)|}$. So in the worst case for a left-linear $R$, the size of $S_1(R)$ could be exponential in the size of $S_0(R)$. However, if $R$ is additionally shallow, as it is in our case, then $S_0$ contains only ground terms plus the variable $x$, since the only subterms of left-hand sides of rules of $R$ that contain variables are the left-hand sides of rules themselves and the variable $x$. In this case $S_1(R) = S_0(R)$.

**Proposition 23.** *For every term $u$ there is a unique term $t \in S_1(R)$ such that $u$ is an instance of $t$ and for any $s \in S_1(R)$, if $u$ is an instance of $s$ then $t$ is an instance of $s$.*

*Proof.* Let $T = \{t \in S_0(R) \mid u \text{ is an instance of } t\}$. $T$ is finite and non-empty (since $x \in T$), so we can write $T$ as $T = \{t_1, t_2, \ldots, t_n\}$. Let $t = t_1 \Downarrow t_2 \Downarrow \cdots \Downarrow t_n$. Then $u$ is an instance of $t$ because $u$ is an instance of each $t_i$. Also for every $s \in S_1(R)$ such that $u$ is an instance of $s$ we have $s \Downarrow t = t$, so $t$ is an instance of $s$. Finally, to show uniqueness, we note that if there are two terms $t$ and $t'$ that satisfy the statement of the proposition, then they are instances of each other, so they are equivalent up to variable renaming. ■

For a term $u$, we denote by $u\Uparrow$ the term $t$ described by the proposition.

Let $S(R)$ be $S_1(R)$ minus the terms that are instances of left-hand sides of rewrite rules. (We must remove instances of left-hand sides of rewrite rules again because the closure procedure might have generated new ones.) Note that $x \in S(R)$ and $S(R)$ contains only linear terms. Also note that if a term $u$ is not an instance of a left-hand side of a rewrite rule, then $u\Uparrow$ is not either, so $u\Uparrow \in S(R)$.

We define the set $Q$ of states of $A_{\mathrm{Red}(R)}$ by $Q = \{q_r\} \cup \{q_t \mid t \in S(R)\}$ and the set $Q_f$ of final states by $Q_f = \{q_r\}$. A term that reaches state $q_r$ will be one that is reducible. Since $x \in S(R)$, there will be a state $q_x \in Q$. A term that reaches state $q_x$ will be one that is not reducible, and is not part of a reducible term except below a variable of some left-hand side of a rule of $R$. The transition rules of $A_{\mathrm{Red}(R)}$ are divided into three groups:

(A1) $f(q_{t_1}, \ldots, q_{t_n}) \to q_{f(t_1,\ldots,t_n)\Uparrow}$

if $f(t_1, \ldots, t_n)$ is not an instance of a left-hand side of a rule of $R$. (Note that $t_1, \ldots, t_n, f(t_1, \ldots, t_n)\Uparrow \in S(R)$ and $q_{t_i} \neq q_r$ for all $i$.)

(A2) $f(q_{t_1}, \ldots, q_{t_n}) \to q_r$

if $f(t_1, \ldots, t_n)$ is an instance of a left-hand side of a rule of $R$. (Note that $t_1, \ldots, t_n \in S(R)$ and $q_{t_i} \neq q_r$ for all $i$.)

(A3) $f(q_1, \ldots, q_n) \to q_r$

if there is some $i$ with $q_i = q_r$.

The rules used in Examples 21 and 22 can be generated by (A1), (A2), and (A3).

**Lemma 24.** *$A_{\mathrm{Red}(R)}$ is deterministic and complete.*

*Proof.* To show determinism, first note that the transition rules generated by (A$i$) and (A$j$) are disjoint when $i \neq j$. Also, neither (A1) nor (A2) nor (A3) generates more than one transition rule with the same left-hand side. Thus no two different transition rules share a left-hand side, so $A_{\mathrm{Red}(R)}$ is deterministic.

We show by induction on the structure of terms that $A_{\mathrm{Red}(R)}$ is complete. Let $u = f(u_1, \ldots, u_n)$ where for each $i$, $u_i$ reaches a state $q_i$, so that $u \vdash^* f(q_1, \ldots, q_n)$. We need to show that there is a state $q$ such that $u \vdash^* q$. If any of the $q_i$ is $q_r$ then $f(q_1, \ldots, q_n) \vdash q_r$ by (A3), so that $u \vdash^* q_r$. So assume that no $q_i$ is $q_r$. Then for each $i$ there is a term $t_i \in S(R)$ such that $q_i = q_{t_i}$. We need to show that there is a state $q$ such that $f(q_{t_1}, \ldots, q_{t_n}) \vdash q$. If $f(t_1, \ldots, t_n)$ is an instance of a left-hand side of a rule of $R$, then $f(q_{t_1}, \ldots, q_{t_n}) \vdash q_r$ so $u \vdash^* q_r$. If $f(t_1, \ldots, t_n)$ is not an instance of a left-hand side of a rule of $R$, then $f(q_{t_1}, \ldots, q_{t_n}) \vdash q_{f(t_1,\ldots,t_n)\Uparrow}$ so $u \vdash^* q_{f(t_1,\ldots,t_n)\Uparrow}$. Thus for every term $u$, there is a state $q$ such that $u \vdash^* q$. ■

If there are $\phi$ number of function symbols in the signature, and the greatest arity of a function symbol is $\alpha$, then the number of rules in $A_{\mathrm{Red}(R)}$ is less than or equal to $\phi|Q|^{\alpha}$. For linear flat rewrite systems, $|Q|$ is a polynomial in the size of $R$, since $S(R) \subseteq S_0(R)$. Also, we assume that $\alpha$ is bounded, so $\alpha$ is independent of the size of $R$. Therefore, for linear flat systems, the size of $A_{\mathrm{Red}(R)}$ is polynomial in the size of $R$.

Now we show that $A_{\mathrm{Red}(R)}$ accepts the terms that are reducible by $R$.

**Lemma 25.** *If a term $u$ reaches a state $q_t$ of $A_{\mathrm{Red}(R)}$, then $u$ is an instance of $t$ and for any $s \in S(R)$, if $u$ is an instance of $s$ then $t$ is an instance of $s$.*

*Proof.* We show this by structural induction. Let $u = f(u_1, \ldots, u_n)$ where for each $i$, if $u_i$ reaches state $q_{t_i}$ then $u_i$ is an instance of $t_i$ and for any $s \in S(R)$, if $u_i$ is an instance of $s$ then $t_i$ is an instance of $s$. Assume $u$ reaches state $q_t$.

First, we show that $u$ is an instance of $t$. By determinism and completeness of $A_{\mathrm{Red}(R)}$, we know that $u \vdash^* f(q_{t_1}, \ldots, q_{t_n}) \vdash q_t$ for some unique sequence of terms $t_1, \ldots, t_n \in S(R)$ where $f(t_1, \ldots, t_n)$ is not an instance of some left-hand side of a rewrite rule and $t = f(t_1, \ldots, t_n)\Uparrow$. By the induction hypothesis, $u_i$ is an instance of $t_i$ for each $i$. Let $t = f(t'_1, \ldots, t'_n)$. Since $f(t_1, \ldots, t_n)$ is an instance of $t$, $t_i$ is an instance of $t'_i$ for each $i$. Thus, for each $i$, $u_i$ is an instance of $t'_i$. Then, because $t$ is linear (remember that all terms in $S(R)$ are linear), $u$ is an instance of $t$.

Now we pick a term $s \in S(R)$ such that $u$ is an instance of $s$ and show that $t$ is an instance of $s$. If $s = x$ then $t$ is an instance of $s$. Otherwise, $s = f(s_1, \ldots, s_n)$ for some $s_1, \ldots, s_n \in S(R)$. For each $i$, $u_i$ is an instance of $s_i$, so by the induction hypothesis $t_i$ is an instance of $s_i$. Then, because $s$ is linear, $f(t_1, \ldots, t_n)$ is an instance of $s$. Finally, $t$ is an instance of $s$ by Proposition 23. ■

**Theorem 26.** *$A_{\mathrm{Red}(R)}$ accepts the language of terms reducible by $R$.*

*Proof.* We show by induction on the structure of terms that a term $u$ reaches state $q_r$ if and only if $u$ is reducible by $R$. Let $u = f(u_1, \ldots, u_n)$ where for each $i$, $u_i$ reaches state $q_r$ if and only if $u_i$ is reducible.

$\Rightarrow$: Assume that $u$ reaches state $q_r$ and show that $u$ is reducible. If $u \vdash^* f(q_1, \ldots, q_n) \vdash q_r$ where $q_i = q_r$ for some $i$, then $u_i$ is reducible so $u$ is reducible. Otherwise $u \vdash^* f(q_{t_1}, \ldots, q_{t_n}) \vdash q_r$ for some terms $t_1, \ldots, t_n \in S(R)$, where $f(t_1, \ldots, t_n)$ is an instance of some left-hand side $f(l_1, \ldots, l_n)$ of a rule of $R$. For all $i$, $u_i$ is an instance of $t_i$ by Lemma 25, and $t_i$ is an instance of $l_i$, so $u_i$ is an instance of $l_i$. Then, because $f(l_1, \ldots, l_n)$ is linear, $u$ is an instance of $f(l_1, \ldots, l_n)$. Thus $u$ is reducible.

$\Leftarrow$: Assume $u$ is reducible and show that $u$ reaches $q_r$. If $u_i$ is reducible for some $i$, then $u_i$ reaches $q_r$ so $u$ reaches $q_r$ by (A3). So assume for each $i$ that $u_i$ is not reducible. Then $u$ must be an instance of a left-hand side $f(l_1, \ldots, l_n)$ of a rule of $R$ and $u \vdash^* f(q_{t_1}, \ldots, q_{t_n})$ for some terms $t_1, \ldots, t_n \in S(R)$. We will show that $f(t_1, \ldots, t_n)$ is an instance of $f(l_1, \ldots, l_n)$ so that there is a transition rule $f(q_{t_1}, \ldots, q_{t_n}) \to q_r$ by (A2). For each $i$ we know that $u_i$ is an instance of $l_i$. Also it must be the case that $l_i \in S(R)$. For, if $l_i \notin S(R)$, then $l_i$ would be an instance of a left-hand side of a rule of $R$, which would make $u_i$ reducible. Now Lemma 25 implies that $t_i$ is an instance of $l_i$, since $u_i$ reaches state $q_{t_i}$. Then, because $f(l_1, \ldots, l_n)$ is linear, $f(t_1, \ldots, t_n)$ is an instance of $f(l_1, \ldots, l_n)$. Thus the transition rule $f(q_{t_1}, \ldots, q_{t_n}) \to q_r$ exists, so $u$ reaches $q_r$. ■

**Corollary 27.** *For a left-linear flat TRS $R$ and a ground term $t$, we can decide in polynomial time if $t$ is a normal form of $R$.*

*Proof.* By [4], membership of $t$ in $A_{\mathrm{Red}(R)}$ can be decided in linear time. ■

# 6 Tree Automata for Reachable Terms

Given a ground term $t$, we want a tree automaton that can recognize ground terms $R$-equivalent to $t$ by ground derivations, where $R$ is our linear flat rewrite system. Equivalently, we we want a tree automaton that can recognize terms reachable via $R \cup R^-$ from $t$. Note that $R \cup R^-$ is also a linear flat rewrite system.

While we have assumed that $R$ has no variable left-hand sides of rules, we must take into account that $R \cup R^-$ may have variables as left-hand sides of rules.

We can use Lemma 25 from [3] to get this result. Comon shows there that linear shallow term rewrite systems preserve regularity. A term rewrite system $R'$ preserves regularity if for any recognizable subset $L$ of $\mathcal{T}(\mathcal{F}, X)$, the set $\{s \in \mathcal{T}(\mathcal{F}, X) \mid s \to_{R'}^* u \text{ for some } u \in L\}$ is recognizable. For our purposes, $L = \{t\}$, which is recognizable. Also, for any term $s$, $s \to_{R \cup R^-}^* t$ if and only if $t \to_{R \cup R^-}^* s$. Thus the set $\{s \in \mathcal{T}(\mathcal{F}, X) \mid t \to_{R'}^* s\}$ is recognizable, i.e. there is a tree automaton that recognizes the terms reachable via $R \cup R^-$ from $t$. Therefore, there is a tree automaton that recognizes the ground terms that are $R$-equivalent to $t$ by ground derivations.

To keep this paper self-contained, we include a direct construction of a version of the Comon result. Our construction is more limited than [3] because the source is one term $t$ rather than any recognizable set of terms. However, this construction actually does allow rewrite systems with rules having variables as left-hand sides, which are mentioned in but omitted from Comon's proof.

Note that tree automata operate on ground terms of a fixed signature.

Let $R$ be a linear shallow rewrite system. $R$ need not be flat and may have variables as left-hand sides of rules. Let $S$ be the set containing the subterms of $t$ and the ground subterms of the left- and right-hand sides of the rules of $R$. (Note that $S$ contains only ground terms since $t$ is ground.) We first define a nondeterministic automaton $B_0$ for which the set $Q$ of states is defined by $Q = \{q_s \mid s \in S\} \cup \{q_x\}$, and the set $Q_f$ of final states is defined by $Q_f = \{q_t\}$, and the set $\Delta_0$ of transition rules is defined by

$$\begin{aligned}\Delta_0 = \{&f(q_{s_1}, \ldots, q_{s_n}) \to q_{f(s_1,\ldots,s_n)} \mid s_1, \ldots, s_n, f(s_1, \ldots, s_n) \in S\} \\ &\cup \{f(q_1, \ldots, q_n) \to q_x \mid q_1, \ldots, q_n \in Q\}\,.\end{aligned}$$

At this point, $B_0$ accepts only $t$. Every term reaches $q_x$, so $B_0$ is complete.

We construct $B$ by keeping the states and final states of $B_0$, and letting the set $\Delta$ of transition rules be the smallest set of transition rules that contains $\Delta_0$ and is closed under the inference rules

(B1) $$\frac{f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m) \in R \qquad f(q_1, \ldots, q_n) \to q \in \Delta}{g(q'_1, \ldots, q'_m) \to q \in \Delta}$$

if the following requirements are met:

1. For each $i$, if $l_i$ is ground then $l_i \vdash^* q_i$.
2. $q'_j$ is defined by

$$q'_j = \begin{cases} q_{r_j} & \text{if } r_j \text{ is ground} \\ q_i & \text{if } r_j \text{ is a variable and } r_j = l_i \\ q_x & \text{if } r_j \text{ is a variable that does not appear in } f(l_1, \ldots, l_n) \end{cases}$$

(Note that if $r_j$ is a variable and there is an $i$ such that $r_j = l_i$, then that $i$ is unique because $f(l_1, \ldots, l_n)$ is linear.)

(B2) $$\frac{f(l_1, \ldots, l_n) \to x \in R \qquad f(q_1, \ldots, q_n) \to q \in \Delta}{g(q'_1, \ldots, q'_m) \to q \in \Delta}$$

if the following requirements are met:

1. For each $i$, if $l_i$ is ground then $l_i \vdash^* q_i$.
2. If $x = l_i$ then there is already a transition rule $g(q'_1, \ldots, q'_m) \to q_i \in \Delta$.

(If $x$ does not appear in $f(l_1, \ldots, l_n)$ then the function symbol $g$ and states $q'_1, \ldots, q'_m$ are completely free.)

(B3) $\dfrac{x \to g(r_1, \ldots, r_m) \in R \qquad q \in Q}{g(q'_1, \ldots, q'_m) \to q \in \Delta}$
if $q'_j$ is defined by

$$q'_j = \begin{cases} q_{r_j} & \text{if } r_j \text{ is ground} \\ q & \text{if } r_j \text{ is a variable and } r_j = x \\ q_x & \text{if } r_j \text{ is a variable and } r_j \neq x\,. \end{cases}$$

We add only rules and not states, so there are only a finite number of rules that can be added to $\Delta_0$. Thus, we can view the construction of $B$ as a terminating process of adding one transition rule after another for a finite number of iterations. If there are $\phi$ number of function symbols in the signature, and the largest arity of a function symbol is $\alpha$, then the number of rules in $B$ is less than or equal to $\phi|Q|^{\alpha+1}$. $|Q|$ is a polynomial in size of $R$. Also, we assume that $\alpha$ is bounded, so $\alpha$ is independent of the size of $R$. Therefore, the size of $B$ is a polynomial in size of $R$.

We will show that the language accepted by automaton $B$ is the set of terms that are reachable from $t$ by $R$. This is a simple consequence of the following lemmas which state that for any term $u$ and any term $s \in S$, $s \to_R^* u$ if and only if $u \vdash_B^* q_s$.

**Lemma 28.** *For any term $u$ and any term $s \in S$, if $s \to_R^* u$ then $u \vdash_B^* q_s$.*

*Proof.* We induct on the length $k$ of $s \to_R^* u$. For $k = 0$ we need to show that for all $u$, $u \vdash_B^* q_u$. This requires only a simple structural induction on $u$. Now we assume for a particular $k$ that for any term $u$ and any term $s \in S$, if $s \to_R^k u$ then $u \vdash_B^* q_s$, and we want to show that for any term $u$ and any term $s \in S$, if $s \to_R^{k+1} u$ then $u \vdash_B^* q_s$. So, we pick a term $u$ and a term $s \in S$ such that $s \to_R^{k+1} u$. We have $s \to_R^k v \to_R u$ for some term $v$. Thus $v \vdash_B^* q_s$ by the induction hypothesis. There are three cases.

*Case 1*: There is a rewrite rule of the form $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m)$ in $R$ and $v = C[f(l_1, \ldots, l_n)\sigma]$ and $u = C[g(r_1, \ldots, r_m)\sigma]$ for some context $C[\,]$ and substitution $\sigma$. Then, since $v \vdash_B^* q_s$, there must be states $q_1, \ldots, q_n$ and a state $q$ such that

$$v = C[f(l_1, \ldots, l_n)\sigma] \vdash_B^* C[f(q_1, \ldots, q_n)] \vdash_B C[q] \vdash_B^* q_s\,.$$

This implies that $l_i\sigma \vdash_B^* q_i$ for each $i$, and that there must be a transition rule $f(q_1, \ldots, q_n) \to q$ of $B$. For each $l_i$ that is ground, $l_i \vdash_B^* q_i$ since $l_i\sigma = l_i$. We also know that

$$u = C[g(r_1, \ldots, r_n)\sigma] \vdash_B^* C[g(q'_1, \ldots, q'_n)]$$

when the states $q'_1, \ldots, q'_m$ are defined by

$$q'_j = \begin{cases} q_{r_j} & \text{if } r_j \text{ is ground} \\ q_i & \text{if } r_j \text{ is a variable and } r_j = l_i \text{ (Remember } l_i\sigma \vdash_B^* q_i) \\ q_x & \text{if } r_j \text{ is a variable that does not appear in } f(l_1, \ldots, l_n)\,. \end{cases}$$

By (B1), we must have added the transition rule $g(q'_1, \ldots, q'_m) \to q$ to $B$. Thus we have the result $u = C[g(r_1, \ldots, r_n)\sigma] \vdash_B^* C[g(q'_1, \ldots, q'_n)] \vdash_B C[q] \vdash_B^* q_s$. Thus $u \vdash_B^* q_s$.

*Case 2*: There is a rewrite rule of the form $f(l_1, \ldots, l_n) \to x$ in $R$ and $v = C[f(l_1, \ldots, l_n)\sigma]$ and $u = C[x\sigma]$ for some context $C[\,]$ and substitution $\sigma$. Note that $x\sigma$ will be a term of the form $g(w_1, \ldots, w_m)$. Then, since $v \vdash_B^* q_s$, there must be states $q_1, \ldots, q_n$ and state $q$ such that

$$v = C[f(l_1, \ldots, l_n)\sigma] \vdash_B^* C[f(q_1, \ldots, q_n)] \vdash_B C[q] \vdash_B^* q_s\,.$$

This implies that $l_i\sigma \vdash_B^* q_i$ for each $i$, and that there must be a transition rule $f(q_1, \ldots, q_n) \to q$ of $B$. For each $l_i$ that is ground, $l_i \vdash_B^* q_i$ since $l_i\sigma = l_i$. Now we have to treat two subcases. First, if $x = l_i$ for some (unique) $i$, then $x\sigma \vdash_B^* q_i$, so $x\sigma \vdash_B^* g(q'_1, \ldots, q'_m) \vdash_B q_i$ for some states $q'_1, \ldots, q'_m$. By (B2), we must have added the transition rule $g(q'_1, \ldots, q'_m) \to q$ to $B$. Thus we have the result $u = C[x\sigma] \vdash_B^* C[g(q'_1, \ldots, q'_n)] \vdash_B C[q] \vdash_B^* q_s$. Second, if $x$ does not appear in $f(l_1, \ldots, l_n)$, then $x\sigma = g(w_1, \ldots, w_m) \vdash_B^* g(q'_1, \ldots, q'_m)$ for some states $q'_1$, $\ldots$, $q'_m$. By (B2), we must have added the transition rule $g(q'_1, \ldots, q'_m) \to q$ to $B$. Thus we have the result $u = C[x\sigma] \vdash_B^* C[g(q'_1, \ldots, q'_n)] \vdash_B C[q] \vdash_B^* q_s$. Thus $u \vdash_B^* q_s$.

*Case 3*: There is a rewrite rule of the form $x \to g(r_1, \ldots, r_m)$ in $R$ and $v = C[x\sigma]$ and $u = C[g(r_1, \ldots, r_m)\sigma]$ for some context $C[\,]$ and substitution $\sigma$. Then, since $v \vdash_B^* q_s$, there must be a state $q$ such that

$$v = C[x\sigma] \vdash_B^* C[q] \vdash_B^* q_s\,.$$

We know that

$$u = C[g(r_1, \ldots, r_n)\sigma] \vdash_B^* C[g(q'_1, \ldots, q'_n)]$$

when the states $q'_1$, $\ldots$, $q'_m$ are defined by

$$q'_j = \begin{cases} q_{r_j} & \text{if } r_j \text{ is ground} \\ q & \text{if } r_j \text{ is a variable and } r_j = x \text{ (Remember } x\sigma \vdash_B^* q) \\ q_x & \text{if } r_j \text{ is a variable that does not appear in } g(r_1, \ldots, r_m)\,. \end{cases}$$

By (B3), we must have added the transition rule $g(q'_1, \ldots, q'_m) \to q$ to $B$. Thus we have the result $u = C[g(r_1, \ldots, r_n)\sigma] \vdash_B^* C[g(q'_1, \ldots, q'_n)] \vdash_B C[q] \vdash_B^* q_s$. Thus $u \vdash_B^* q_s$. ■

Now we prove the converse:

**Lemma 29.** *For any term $u$ and any term $s \in S$, if $u \vdash_B^* q_s$ then $s \to_R^* u$.*

*Proof.* We induct on the number of inference rule applications in $u \vdash_B^* q_s$. Let $B_N$ be the automaton generated after $N$ inference rule applications. For the base case we need to show that if $u \vdash_{B_0}^* q_s$ then $s \to_R^* u$. An easy structural induction on $u$ shows that, in fact, we can prove that $s = u$. Now we assume for a particular $N$ that for any term $u$ and any term $s \in S$, if $u \vdash_{B_N}^* q_s$ then $s \to_R^* u$, and we want to show that for any term $u$ and any term $s \in S$, if $u \vdash_{B_{N+1}}^* q_s$ then $s \to_R^* u$. Let $\rho$ be the transition rule $g(q'_1, \ldots, q'_m) \to q$ that we added to $B_N$ to get $B_{N+1}$. We induct on the number $M$ of $\rho$-applications in $u \vdash_{B_{N+1}}^* q_s$. For $M = 0$, if we pick a term $u$ and a term $s \in S$ such that $u \vdash_{B_{N+1}}^* q_s$ with $M$ applications of transition rule $\rho$, then $u \vdash_{B_N}^* q_s$, so $s \to_R^* u$ by the induction hypothesis on $N$. Now we assume for a particular $M$ that for any term $u$ and any term $s \in S$, if $u \vdash_{B_{N+1}}^* q_s$ with $M$ applications of transition rule $\rho$, then $s \to_R^* u$. We want to show that for any term $u$ and any term $s \in S$, if $u \vdash_{B_{N+1}}^* q_s$ with $M + 1$ applications of transition rule $\rho$ then $s \to_R^* u$. So, we pick a term $u$ and a term $s \in S$ such that $u \vdash_{B_{N+1}}^* q_s$ with $M + 1$ applications of transition rule $\rho$. Then we have $u \vdash_{B_N}^* v_1 \vdash_\rho v_2 \vdash_{B_{N+1}}^* q_s$ for some $v_1$ and $v_2$ where there are $M$ applications of $\rho$ in $v_2 \vdash_{B_{N+1}}^* q_s$. Because of the application of transition rule $\rho$, we must have $v_1 = C[g(q'_1, \ldots, q'_m)]$ and $v_2 = C[q]$ for some context $C[\,]$. Consequently, $u = C'[g(u_1, \ldots, u_m)]$ for some $u_1$, $\ldots$, $u_m$ where $C'[\,] \vdash_{B_N}^* C[\,]$ and $u_j \vdash_{B_N}^* q'_j$ for every $j$. To recap, the situation is

$$u = C'[g(u_1, \ldots, u_m)] \vdash_{B_N}^* C[g(q'_1, \ldots, q'_m)] \vdash_\rho C[q] \vdash_{B_{N+1}}^* q_s\,.$$

Now there are three cases to consider.

*Case 1*: Transition rule $\rho$ was added by (B1). Then there is a rewrite rule $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m) \in R$, and there are states $q_1$, $\ldots$, $q_n$ such that if $l_i$ is ground then $l_i \vdash_{B_N}^* q_i$ for each $i$, and $f(q_1, \ldots, q_n) \to q$ is a transition rule of $B_N$, and we know about each $q'_j$ that

$$q'_j = \begin{cases} q_{r_j} & \text{if } r_j \text{ is ground} \\ q_i & \text{if } r_j \text{ is a variable and } r_j = l_i \\ q_x & \text{if } r_j \text{ is a variable that does not appear in } f(l_1, \ldots, l_n)\,. \end{cases}$$

We define a new term $v$ by $v = C'[f(l_1, \ldots, l_n)\sigma]$, where the substitution $\sigma$ is defined by

$$y\sigma = \begin{cases} u_j & \text{if } y = r_j \\ w_i & \text{if } y = l_i \text{ and } l_i \text{ does not appear in } g(r_1, \ldots, r_m)\,, \end{cases}$$

where $w_i$ is any term that reaches state $q_i$. (We can always find such a $w_i$. If $q_i = q_{s'}$ where $s' \in S$, then we can let $w_i = s'$. If $q_i = q_x$, then any term will do.) For this $v$, we have the result

$$v = C'[f(l_1, \ldots, l_n)\sigma] \vdash^*_{B_N} C[f(q_1, \ldots, q_n)] \vdash_{B_N} C[q] \vdash^*_{B_{N+1}} q_s$$

where the entire derivation contains $M$ applications of $\rho$. The justification for this is as follows. First, if $l_i$ is ground then $l_i \vdash^*_{B_N} q_i$ and $l_i\sigma = l_i$, so $l_i\sigma \vdash^*_{B_N} q_i$. Second, if $l_i$ is a variable and $l_i = r_j$ then $l_i\sigma = u_j$ and $q'_j = q_i$. Also remember that $u_j \vdash^*_{B_N} q'_j$, which gives us that $l_i\sigma \vdash^*_{B_N} q_i$. Third, if $l_i$ is a variable that does not occur in $g(r_1, \ldots, r_m)$, then $l_i\sigma = w_i$. Since $w_i$ reaches state $q_i$, we again have $l_i\sigma \vdash^*_{B_N} q_i$. Finally, the presence of the transition rule $f(q_1, \ldots, q_n) \to q$ gives us the result.

Since $v \vdash^*_{B_{N+1}} q_s$ with $M$ applications of $\rho$, by the induction hypothesis on $M$ we know that $s \to^*_R v$. It remains to show that $v \to^*_R u$. We know that

$$v = C'[f(l_1, \ldots, l_n)\sigma] \to_R C'[g(r_1, \ldots, r_m)\sigma]\,,$$

so we just need to show that

$$C'[g(r_1, \ldots, r_m)\sigma] \to^*_R C'[g(u_1, \ldots, u_m)] = u\,.$$

If $r_j$ is a variable, then $r_j\sigma = u_j$, so $r_j\sigma \to^*_R u_j$. If $r_j$ is ground, then $r_j\sigma = r_j$ and $q'_j = q_{r_j}$. Remember that $u_j \vdash^*_{B_N} q'_j$, so $u_j \vdash^*_{B_N} q_{r_j}$. By the induction hypothesis on $N$, we have $r_j \to^*_R u_j$, and thus $r_j\sigma \to^*_R u_j$. Therefore $s \to^*_R u$.

*Case 2*: Transition rule $\rho$ was added by (B2). Then there is a rewrite rule $f(l_1, \ldots, l_n) \to x \in R$, and there are states $q_1, \ldots, q_n$ such that if $l_i$ is ground then $l_i \vdash^*_{B_N} q_i$ for each $i$, and $f(q_1, \ldots, q_n) \to q$ is a transition rule of $B_N$, and if $x = l_i$ then there is also a transition rule $g(q'_1, \ldots, q'_m) \to q_i$ in $B_N$.

We define a new term $v$ by $v = C'[f(l_1, \ldots, l_n)\sigma]$, where the substitution $\sigma$ is defined by

$$y\sigma = \begin{cases} g(u_1, \ldots, u_m) & \text{if } y = x \\ w_i & \text{if } y = l_i \text{ and } l_i \neq x\,, \end{cases}$$

where $w_i$ is any term that reaches state $q_i$. (We can always find such a $w_i$. If $q_i = q_{s'}$ where $s' \in S$, then we can let $w_i = s'$. If $q_i = q_x$, then any term will do.) For this $v$, we have the result

$$v = C'[f(l_1, \ldots, l_n)\sigma] \vdash^*_{B_N} C[f(q_1, \ldots, q_n)] \vdash_{B_N} C[q] \vdash^*_{B_{N+1}} q_s$$

where the entire derivation contains $M$ applications of $\rho$. The justification for this is as follows. First, if $l_i$ is ground then $l_i \vdash^*_{B_N} q_i$ and $l_i\sigma = l_i$, so $l_i\sigma \vdash^*_{B_N} q_i$. Second, if $l_i$ is a variable and $l_i \neq x$, then $l_i\sigma = w_i$. Since $w_i$ reaches state $q_i$, we again have $l_i\sigma \vdash^*_{B_N} q_i$. Third, if $l_i$ is a variable and $l_i = x$, then $l_i\sigma = g(u_1, \ldots, u_m) \vdash^*_{B_N} g(q'_1, \ldots, q'_m) \vdash_{B_N} q_i$ since by (B2) there must be a transition rule $g(q'_1, \ldots, q'_m) \to q_i$ of $B_N$.

Since $v \vdash^*_{B_{N+1}} q_s$ with $M$ applications of $\rho$, by the induction hypothesis on $M$ we know that $s \to^*_R v$. It remains to show that $v \to^*_R u$, which holds because

$$v = C'[f(l_1, \ldots, l_n)\sigma] \to_R C'[x\sigma] = C'[g(u_1, \ldots, u_m)] = u\,.$$

*Case 3*: Transition rule $\rho$ was added by (B3). Then there is a rewrite rule $x \to g(r_1, \ldots, r_m) \in R$, a state $q \in Q$, and we know about each $q'_j$ that

$$q'_j = \begin{cases} q_{r_j} & \text{if } r_j \text{ is ground} \\ q & \text{if } r_j \text{ is a variable and } r_j = x \\ q_x & \text{if } r_j \text{ is a variable and } r_j \neq x\,. \end{cases}$$

We define a new term $v$ by $v = C'[x\sigma]$, where the substitution $\sigma$ is defined by

$$y\sigma = \begin{cases} u_j & \text{if } y = r_j \\ w & \text{if } y = x \text{ and } x \text{ does not appear in } g(r_1, \ldots, r_m) \,, \end{cases}$$

where $w$ is any term that reaches state $q$. (We can always find such a $w$. If $q = q_{s'}$ where $s' \in S$, then we can let $w = s'$. If $q = q_x$, then any term will do.) For this $v$, we have the result

$$v = C'[x\sigma] \vdash^*_{B_N} C[q] \vdash^*_{B_{N+1}} q_s$$

where the entire derivation contains $M$ applications of $\rho$. The justification for this is as follows. First, if $x = r_j$ then $x\sigma = r_j\sigma = u_j$ and $q'_j = q$. Also remember that $u_j \vdash^*_{B_N} q'_j$, which gives us that $x\sigma \vdash^*_{B_N} q$. Second, if $x$ does not occur in $g(r_1, \ldots, r_m)$, then $x\sigma = w$. Since $w$ reaches state $q$, we again have $x\sigma \vdash^*_{B_N} q$.

Since $v \vdash^*_{B_{N+1}} q_s$ with $M$ applications of $\rho$, by the induction hypothesis on $M$ we know that $s \to^*_R v$. It remains to show that $v \to^*_R u$. We know that

$$v = C'[x\sigma] \to_R C'[g(r_1, \ldots, r_m)\sigma] \,,$$

so we just need to show that

$$C'[g(r_1, \ldots, r_m)\sigma] \to^*_R C'[g(u_1, \ldots, u_m)] = u \,.$$

If $r_j$ is a variable, then $r_j\sigma = u_j$, so $r_j\sigma \to^*_R u_j$. If $r_j$ is ground, then $r_j\sigma = r_j$ and $q'_j = q_{r_j}$. Remember that $u_j \vdash^*_{B_N} q'_j$, so $u_j \vdash^*_{B_N} q_{r_j}$. By the induction hypothesis on $N$, we have $r_j \to^*_R u_j$, and thus $r_j\sigma \to^*_R u_j$. Therefore $s \to^*_R u$. ■

**Theorem 30.** *Automaton $B$ accepts the terms reachable from $t$.*

*Proof.* We need to show that for any term $u$, $u \vdash^*_B q_t$ if and only if $t \to^*_R u$. This is a direct consequence of the previous two lemmas. ■

**Corollary 31.** *For a linear, shallow TRS $R$ and two ground terms $s$ and $t$, we can decide in polynomial time if there is a ground derivation of $s \leftrightarrow^*_R t$.*

*Proof.* For term $t$ we create the automaton $B$. By [4], membership of $s$ in $B$ can be decided in time proportional to the size of $B$ plus the size of $s$. ■

# 7 Conclusion and Further Work

This paper has provided a polynomial time algorithm that solves the $\text{UN}^=$ problem for linear shallow term rewrite systems.

The exact boundary between decidability and undecidability for this property is an interesting direction for future research. Recently, it was proved in [11] that the $\text{UN}^=$ problem is undecidable for linear rewrite systems in which the height of both sides of every rule is restricted to at most two. Moreover, it has been shown in [12] that the $\text{UN}^=$ problem is undecidable for right-ground systems. Since right-ground systems can be flattened as describe here so that the right-hand sides are flat terms, the problem is undecidable also for right-flat systems. Thus, the undecidability frontier is not far when the height of sides and linearity restrictions are of interest.

# References


[1] Franz Baader and Tobias Nipkow. *Term Rewriting and All That.* Cambridge University Press, Cambridge, 1998.

[2] Inge Bethke, Jan Willem Klop, and Roel de Vrijer. Descendants and origins in term rewriting. *Information and Computation*, 159(1–2):59–124, May 2000.

[3] Hubert Comon. Sequentiality, monadic second-order logic and tree automata. *Information and Computation*, 157(1–2):25–51, 2000.

[4] Hubert Comon, Max Dauchet, Remi Gilleron, Florent Jacquemard, Denis Lugiez, Sophie Tison, and Marc Tommasi. Tree automata techniques and applications. Available online at `http://www.grappa.univ-lille3.fr/tata/`, September 2005.

[5] Guillem Godoy, Robert Nieuwenhuis, and Ashish Tiwari. Classes of term rewrite systems with polynomial confluence problems. *ACM Transactions on Computational Logic*, 5(2):321–331, April 2004.

[6] Guillem Godoy and Ashish Tiwari. Termination of rewrite systems with shallow right-linear, collapsing, and right-ground rules. In Robert Nieuwenhuis, editor, *Proceedings of the International Conference on Automated Deduction 2005*, volume 3632 of *Lecture Notes in Computer Science*, pages 164–176. Springer, 2005.

[7] Guillem Godoy, Ashish Tiwari, and Rakesh Verma. Deciding confluence of certain term rewriting systems in polynomial time. *Annals of Pure and Applied Logic*, 130(1–3):33–59, 2004.

[8] Donald Knuth and Peter Bendix. Simple word problems in universal algebras. In John Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, Oxford, 1970.

[9] Robert Nieuwenhuis. Decidability and complexity analysis by basic paramodulation. *Information and Computation*, 147:1–21, 1998.

[10] Terese. *Term Rewriting Systems*. Cambridge University Press, Cambridge, 2003.

[11] Rakesh Verma. New undecidability results for problems of term rewriting systems. *Proc. of RULE Workshop*, 2008, to appear.

[12] Rakesh Verma. Complexity of normal form properties and reductions for term rewriting problems. *Fundamenta Informaticae*, 2008. Accepted for publication; to appear.

[13] Rakesh Verma, Michael Rusinowitch, and Denis Lugiez. Algorithms and reductions for rewriting problems. *Fundamenta Informaticae*, 46(3):257–276, 2001. Also in Proceedings of the International Conference on Rewriting Techniques and Applications 1998.

# A Nondeterministic Tree Automata for Reducible Terms

We shall define a nondeterministic tree automaton that accepts $R$-reducible terms. Let $L$ be the set of left-hand sides of rules and let $S(L)$ be the set of non-variable subterms of terms in $L$. We define the set $Q$ of states by $Q = \{q_r, q_x\} \cup \{q_t \mid t \in S(L) - L\}$, and set $Q_f$ of final states by $Q_f = \{q_r\}$. The transition rules of our automaton will be such that reducible terms will reach $q_r$, all terms will reach $q_x$, and instances of a term $t \in S(L) - L$ will reach $q_t$. For each $t = f(t_1, \ldots, t_n)$ in $S(L) - L$ we have a transition rule $f(q_{t_1}, \ldots, q_{t_n}) \to q_t$, where $q_{t_i}$ is understood to be $q_x$ if $t_i$ is a variable. For each left-hand side $f(l_1, \ldots, l_n)$ of a rewrite rule, we have a transition rule $f(q_{l_1}, \ldots, q_{l_n}) \to q_r$, where $q_{l_i}$ is understood to be $q_x$ if $l_i$ is a variable. Finally, we have the transition rules $f(q_1, \ldots, q_n) \to q_r$ if there is an $i$ such that $q_i = q_r$, and $f(q_x, \ldots, q_x) \to q_x$. A simple structural induction shows that every term reaches $q_x$.

**Proposition 32.** *For a term $t \in S(L) - L$ and a term $u$, $u$ reaches $q_t$ if and only if $u$ is an instance of $t$.*

*Proof.* ⇒: By structural induction. Let $u = f(u_1, \ldots, u_n)$ where for each $i$ and any term $t \in S(L) - L$, if $u_i$ reaches $q_t$ then $u_i$ is an instance of $t$. Now we pick a term $t \in S(L) - L$ such that $u$ reaches state $q_t$ and show that $u$ is an instance of $t$. The only way that $u$ can reach $q_t$ is if $t = f(t_1, \ldots, t_n)$ and $u \vdash^* f(q_{t_1}, \ldots, q_{t_n})$. Then we know that $u_i$ is an instance of $t_i$ for each $i$. Because of the linearity of $t$, we get that $u$ is an instance of $t$.

⇐: By structural induction. Let $u = f(u_1, \ldots, u_n)$ where for each $i$ and any term $t \in S(L) - L$, if $u_i$ is an instance of $t$ then $u_i$ reaches $q_t$. Now we pick a term $t \in S(L) - L$ such that $u$ is an instance of $t$ and show that $u$ reaches state $q_t$. Each $u_i$ is an instance of $t_i$, so each $u_i$ reaches state $q_{t_i}$. Thus $u \vdash^* f(q_{t_1}, \ldots, q_{t_n})$. There is a transition rule $f(q_{t_1}, \ldots, q_{t_n}) \to q_t$, so $u$ reaches state $q_t$. ∎

**Lemma 33 ([3]).** *The set of terms reducible by a left-linear TRS $R$ is recognizable by a non-deterministic bottom-up tree automaton.*

*Proof.* We show that a term $u$ reaches state $q_r$ if and only if $u$ is reducible by $R$. Let $u = f(u_1, \ldots, u_n)$ where for each $i$, $u_i$ reaches state $q_r$ if and only if $u_i$ is reducible by $R$.

⇒: Assume that $u$ reaches state $q_r$ and show that $u$ is reducible. If $u \vdash^* f(q_x, \ldots, q_x, q_r, q_x, \ldots, q_x) \vdash q_r$, then there is an $i$ such that $u_i$ reaches $q_r$. Then by the induction hypothesis, $u_i$ is reducible, and thus $u$ is reducible. Otherwise $u \vdash^* f(q_{l_1}, \ldots, q_{l_n}) \vdash q_r$ for some left-hand side $f(l_1, \ldots, l_n)$ of a rule of $R$. By the claim above, $u_i$ is an instance of $l_i$ for each $i$. Because $f(l_1, \ldots, l_n)$ is linear, $u$ is an instance of $f(l_1, \ldots, l_n)$. Thus $u$ is reducible.

⇐: Assume $u$ is reducible and show that $u$ reaches $q_r$. If $u_i$ is reducible for some $i$, then $u_i$ reaches $q_r$, and thus $u$ reaches $q_r$. So assume for each $i$ that $u_i$ is not reducible. Then $u$ must be an instance of a left-hand side $f(l_1, \ldots, l_n)$ of a rule of $R$. By Proposition 32, $u \vdash^* f(q_{l_1}, \ldots, q_{l_n})$, and thus $u \vdash^* q_r$. ∎