Orienting rewrite rules with the Knuth-Bendix order

Konstantin Korovin and Andrei Voronkov

October 8, 2001

Abstract

We consider two decision problems related to the Knuth-Bendix order (KBO). The first problem is *orientability*: given a system of rewrite rules R, does there exist an instance of KBO which orients every ground instance of every rewrite rule in R. The second problem is whether a given instance of KBO orients every ground instance of a given rewrite rule. This problem can also be reformulated as the problem of solving a single ordering constraint for the KBO. We prove that both problems can be solved in polynomial time. The polynomial-time algorithm for orientability builds upon an algorithm for solving systems of homogeneous linear inequalities over integers. The polynomial-time algorithm for solving a single ordering constraint does not need to solve systems of linear inequalities and can be run in time $O(n^2)$. We show that both the orientability problem is P-complete. Also we show that if a system is orientable using a real-valued instance of KBO, then it is also orientable using an integer-valued instance of KBO. Therefore, all our results hold both for the integer-valued and the real-valued KBO.

1 Introduction

In this section we give an informal overview of the results proved in this paper. The formal definitions will be given in the next section.

Let $\succ$ be any order on ground terms and $l \to r$ be a rewrite rule. We say that $\succ$ *orients* $l \to r$, if for every ground instance $l' \to r'$ of $l \to r$ we have $l' \succ r'$. We write $l \succeq r$ if for every ground instance $l' \to r'$ of $l \to r$ we have $l' \succ r'$ or $l' = r'$. There are situations where we want to check if there *exists* a simplification order on ground terms that orients a given system of (possibly non-ground) rewrite rules. We call this problem *orientability*. Orientability can be useful when a theorem prover is run on a new problem for which no suitable simplification order is known, or when termination of a rewrite system is to be established automatically. For a recent survey, see [Dershowitz and Plaisted 2001]. We consider the orientability problem for the Knuth-Bendix order (in the sequel *KBO*) [Knuth and Bendix 1970] on ground terms. We give a polynomial-time algorithm for checking orientability by the KBO. A similar problem of orientability by the non-ground version of the real-valued KBO was studied in [Dick, Kalmus and Martin 1990] and an algorithm for orientability was given. We prove that any rewrite rule system orientable by a real-valued KBO is also orientable by an integer-valued KBO. This result also holds for the non-ground version of the KBO considered in [Dick et al. 1990]. In our proofs we use some techniques of [Dick et al. 1990]. We also show that some rewrite systems could not be oriented by non-ground version of the KBO, but can be oriented by our algorithm.

The second problem we consider is solving ordering constraints consisting of a single inequality, over a given instance of the Knuth-Bendix order. If $\succ$ is total on ground terms, then the problem of checking if $\succ$ orients $l \to r$ has relation to the problem of solving ordering constraints over $\succ$. Indeed, $\succ$ *does not* orient $l \to r$ if and only if there exists a ground instance $l' \to r'$ of $l \to r$

such that $r' \succeq l'$, i.e., if and only if the ordering constraint $r \succeq l$ has a solution. This means that any procedure for solving ordering constraints consisting of a single inequality can be used for checking whether a given system of rewrite rules is oriented by $\succ$, and vice versa. Using the same technique as for the orientability problem, we show that the problem of solving an ordering constraint consisting of a single inequality for the KBO can be solved in polynomial time.

Algorithms for, and complexity of, orientability problem for various versions of the recursive path orders were considered in [Lescanne 1984, Detlefs and Forgaard 1985, Krishnamoorthy and Narendran 1985]. The problems of solving ordering constraints for lexicographic, recursive path orders and for KBO are NP-complete [Comon 1990, Jouannaud and Okada 1991, Nieuwenhuis 1993, Narendran, Rusinowitch and Verma 1999, Korovin and Voronkov 2001], see also [Korovin and Voronkov 2000*b*]. However, to check if $\succ$ orients $l \to r$, it is sufficient to check solvability of a *single* ordering constraint $r \succeq l$. This problem is NP-complete for LPO [Comon and Treinen 1994], and therefore the problem of checking if an LPO orients a rewrite rule is coNP-complete.

2 Preliminaries

A *signature* is a finite set of function symbols with associated arities. In this paper Σ denotes an arbitrary signature. *Constants* are function symbols of the arity 0. We assume that Σ contains at least one constant. We denote variables by x, y, z, constants by a, b, c, d, e, function symbols by f, g, h, and terms by l, r, s, t. Systems of rewrite rules and rewrite rules are defined as usual, see e.g. [Baader and Nipkow 1998, Dershowitz and Plaisted 2001]. An expression E (e.g. a term or a rewrite rule) is called *ground* if no variable occurs in E. Denote the set of natural numbers by $\mathbb{N}$.

The Knuth-Bendix order is a family of orders parametrized by two parameters: a weight function and a precedence relation.

Definition 2.1 (weight function) We call a *weight function* on Σ any function $w : \Sigma \to \mathbb{N}$ such that (i) $w(a) > 0$ for every constant $a \in \Sigma$, (ii) there exist at most one unary function symbol $f \in \Sigma$ such that $w(f) = 0$. Given a weight function w, we call $w(g)$ the *weight* of g. The *weight* of any ground term t, denoted $|t|$, is defined as follows: for every constant c we have $|c| = w(c)$ and for every function symbol g of a positive arity $|g(t_1, \ldots, t_n)| = w(g) + |t_1| + \ldots + |t_n|$.

Definition 2.2 A *precedence relation* on Σ is any total order $\gg$ on Σ. A precedence relation $\gg$ is said to be *compatible* with a weight function w if for every unary function symbol f, if $w(f) = 0$, then f is the greatest element w.r.t. $\gg$.

Definition 2.3 (Knuth-Bendix order) Let w be a weight function on Σ and $\gg$ a precedence relation on Σ compatible with w. The instance of the *Knuth-Bendix order induced by* $(w, \gg)$ is the binary relation $\succ$ on the set of ground terms of Σ defined as follows. For all ground terms $t = g(t_1, \ldots, t_n)$ and $s = h(s_1, \ldots, s_k)$ we have $t \succ s$ if one of the following conditions holds:

1. $|t| > |s|$;
2. $|t| = |s|$ and $g \gg h$;
3. $|t| = |s|$, $g = h$ and for some $1 \leq i \leq n$ we have $t_1 = s_1, \ldots, t_{i-1} = s_{i-1}$ and $t_i \succ s_i$.

The compatibility condition ensures that every instance of the Knuth-Bendix order is a simplification order total on ground terms.

In the sequel we will often refer to the least and the greatest terms among the terms of the minimal weight for a given instance of KBO. It is easy to see that every term of the minimal weight is either a constant of the minimal weight, or a term $f^n(c)$, where c is a constant of the minimal weight, and $w(f) = 0$. Therefore, the least term of the minimal weight is always the constant of the minimal weight which is the least among all such constants w.r.t. $\gg$. This constant is also the least term w.r.t. $\succ$.

The greatest term of the minimal weight exists if and only if there is no unary function symbol of the weight 0. In this case, this term is the constant of the minimal weight which is the greatest among such constants w.r.t. $\gg$.

DEFINITION 2.4 (substitution) A *substitution* is a mapping from a set of variables to the set of terms. A substitution θ is *grounding* for an expression E (i.e., term, rewrite rule etc.) if for every variable x occurring in E the term $\theta(x)$ is ground. We denote by $E\theta$ the expression obtained from E by replacing in it every variable x by $\theta(x)$. A *ground instance* of an expression E is any expression $E\theta$ which is ground.

The following definition is central to this paper.

DEFINITION 2.5 (orientability) An instance $\succ$ of KBO *orients* a rewrite rule $l \to r$ if for every ground instance $l' \to r'$ of $l \to r$ we have $l' \succ r'$. An instance of KBO *orients a system* R of rewrite rules if it orients every rewrite rule in R.

The decidability of the orientability problem for the KBO does not follow immediately from the decidability of the KBO ordering constraints [Korovin and Voronkov 2001], as it is in the case of the recursive path ordering. For a given finite signature, there exists only a finite number of instances of the recursive path ordering. But there exists an infinite number of instances of the KBO, since the weight function is not restricted.

We define orientability in terms of ground instances of rewrite rules. One can also define orientability using the non-ground version of the KBO as originally defined in [Knuth and Bendix 1970]. But then we obtain a weaker notion (fewer systems can be oriented) as the following example from [Korovin and Voronkov 2000*a*] shows.

EXAMPLE 2.6 Consider the following rewrite rule:

$$g(x, a, b) \to g(b, b, a). \tag{1}$$

For any choice of the weight function w and order $\gg$, $g(x, a, b) \succ g(b, b, a)$ does not hold for the original Knuth-Bendix order with variables. However, rewrite rule (1) can be oriented by any instance of KBO such that $w(a) \geq w(b)$ and $a \gg b$.

In fact the order based on all ground instances is the greatest simplification order extending the instance of KBO from ground terms to non-ground terms.

3 Systems of homogeneous linear inequalities

In our proofs and in the algorithm we will use several properties of homogeneous linear inequalities. The definitions related to systems of linear inequalities can be found in standard textbooks (e.g., [Schrijver 1998]). We will denote column vectors of variables by X, integer or real vectors by V, W, integer or real matrices by A, B. Column vectors consisting of 0's will be denoted by $\mathbf{0}$. The set of real numbers is denoted by $\mathbb{R}$, and the set of non-negative real numbers by $\mathbb{R}^+$.

Definition 3.1 (homogeneous linear inequalities) A *homogeneous linear inequality* has the form either $VX \geq 0$ or $VX > 0$. A *system of homogeneous linear inequalities* is a finite set of homogeneous linear inequalities.

Solutions (real or integer) to systems of homogeneous linear inequalities are defined as usual.

We will use the following fundamental property of system of homogeneous linear inequalities:

Lemma 3.2 *Let $AX \geq \mathbf{0}$ be a system of homogeneous linear inequalities, where A is an integer matrix. Then there exists a finite number of integer vectors $V_1, \ldots, V_n$ such that the set of solutions to $AX \geq \mathbf{0}$ is*

$$\{r_1 V_1 + \ldots + r_n V_n \mid r_1, \ldots, r_n \in \mathbb{R}^+\}. \tag{2}$$

The proof can be found in e.g., [Schrijver 1998].

The following lemma was proved in [Martin 1987] for the systems of linear homogeneous inequalities over the real numbers. We will give a simpler proof of it here.

Lemma 3.3 *Let $AX \geq \mathbf{0}$ be a system of homogeneous linear inequalities where A is an integer matrix and* Sol *be the set of all real solutions to the system. Then the system can be split into two disjoint subsystems $BX \geq \mathbf{0}$ and $CX \geq \mathbf{0}$ such that*

1. *$BV = \mathbf{0}$ for every $V \in \mathrm{Sol}$.*
2. *If C is non-empty then there exists a solution $V \in \mathrm{Sol}$ such that $CV > \mathbf{0}$.*

Proof. By Lemma 3.2 we can find integer vectors $V_1, \ldots, V_n$ such that the set Sol is (2). We define $BX \geq \mathbf{0}$ to be the system consisting of all inequalities $WX \geq 0$ in the system such that $WV_i = 0$ for all $i = 1, \ldots, n$; then property 1 is obvious.

Note that the system $CX \geq \mathbf{0}$ consists of the inequalities $WX \geq 0$ such that for some V_i we have $WV_i > 0$. Take V to be $V_1 + \ldots + V_n$, then it is not hard to argue that $CV > \mathbf{0}$. □

Let $\mathbb{D}$ be a system of homogeneous linear inequalities with a real matrix. We will call the subsystem $BX \geq \mathbf{0}$ of $\mathbb{D}$ the *degenerate subsystem* if the following holds. Denote by C the matrix of the complement to $BX \geq \mathbf{0}$ in $\mathbb{D}$ and by Sol the set of all real solutions to $\mathbb{D}$. Then

1. $BV = \mathbf{0}$ for every $V \in \mathrm{Sol}$.
2. If C is non-empty then there exists a solution $V \in \mathrm{Sol}$ such that $CV > \mathbf{0}$.

For every system $\mathbb{D}$ of homogeneous linear inequalities the degenerate subsystem of $\mathbb{D}$ will be denoted by $\mathbb{D}^=$. Note that the degenerate subsystem is defined for arbitrary systems, not only those of the form $AX \geq 0$.

Let us now prove another key property of integer systems of homogeneous linear inequalities: the existence of a real solution implies the existence of an integer solution.

Lemma 3.4 *Let $\mathbb{D}$ be a system of homogeneous linear inequalities with an integer matrix. Let V be a real solution to this system and for some subsystem of $\mathbb{D}$ with the matrix B we have $BV > \mathbf{0}$. Then there exists an integer solution V' to $\mathbb{D}$ for which we also have $BV' > \mathbf{0}$.*

Proof. Let $\mathbb{D}'$ be obtained from $\mathbb{D}$ by replacement of all strict equalities $WX > 0$ by their non-strict versions $WX \geq 0$. Take vectors $V_1, \ldots, V_n$ so that the set of solutions to $\mathbb{D}'$ is (2). Evidently, for every inequality $WX \geq 0$ in $BV > \mathbf{0}$ there exists some V_i such that $WV_i > 0$. Define V' as $V_1 + \ldots + V_n$, then it is not hard to argue that $BV' > 0$. We claim that V' is a solution to $\mathbb{D}$. Assume the converse, then there exists an inequality $WX > 0$ in $\mathbb{D}$ such that $WV' = 0$. But $WV' = 0$ implies that $WV_i = 0$ for all i, so $\mathbb{D}$ has no real solution, contradiction. □

The following lemma follows from Lemmas 3.3 and 3.4.

Lemma 3.5 *Let $\mathbb{D}$ be a system of homogeneous linear inequalities with an integer matrix and its degenerate subsystem is different from $\mathbb{D}$. Let B be the matrix of the complement of the degenerate subsystem. Then there exists an integer solution V to $\mathbb{D}$ such that $BV > \mathbf{0}$.* □

The following result is well-known, see e.g., [Schrijver 1998].

Lemma 3.6 *The existence of a real solution to a system of linear inequalities can be decided in polynomial time.* □

This lemma and Lemma 3.4 imply the following key result.

Lemma 3.7 *(i) The existence of an integer solution to an integer system of homogeneous linear inequalities can be decided in polynomial time. (ii) If an integer system $\mathbb{D}$ of homogeneous linear inequalities has a solution, then its degenerate subsystem $\mathbb{D}^=$ can be found in polynomial time.* □

4 States

In Section 6 we will present an algorithm for orientability by the Knuth-Bendix order. This algorithm will work on *states* which generalize systems of rewrite rules in several ways. A state will use a generalization of rewrite rules to tuples of terms and some information about possible solutions.

Let $\succ$ be any order on ground terms. We extend it lexicographically to an order on tuples of ground terms as follows: we write $\langle l_1, \ldots, l_n \rangle \succ \langle r_1, \ldots, r_n \rangle$ if for some $i \in \{1, \ldots, n\}$ we have $l_1 = r_1, \ldots, l_{i-1} = r_{i-1}$ and $l_i \succ r_i$. We call a *tuple inequality* any expression $\langle l_1, \ldots, l_n \rangle > \langle r_1, \ldots, r_n \rangle$. The *length* of this tuple inequality is n.

In the sequel we assume that Σ is a fixed signature and e is a constant not belonging to Σ. The constant e will play the role of a temporary substitute for a constant of the minimal weight. We will present the algorithm for orienting a system of rewrite rules as a sequence of state changes. We call a *state* a tuple $(\mathbb{R}, \mathbb{M}, \mathbb{D}, \mathbb{U}, \mathbb{G}, \mathbb{L}, \ggg)$, where

1. $\mathbb{R}$ is a set of tuple inequalities $\langle l_1, \ldots, l_n \rangle > \langle r_1, \ldots, r_n \rangle$, such that every two different tuple inequalities in this set have disjoint variables.

2. $\mathbb{M}$ is a set of variables. This set denotes the variables ranging over the terms of the minimal weight.

3. $\mathbb{D}$ is a system of homogeneous linear inequalities over the variables $\{w_g \mid g \in \Sigma \cup \{e\}\}$. This system denotes constraints on the weight function collected so far, and w_e denotes the minimal weight of terms.

4. $\mathbb{U}$ is one of the following values *one* or *any*. The value *one* signals that there exists exactly one term of the minimal weight, while *any* means that no constraints on the number of elements of the minimal weight have been imposed.

5. $\mathbb{G}$ and $\mathbb{L}$ are sets of constants, each of them contains at most one element. If $d \in \mathbb{G}$ (respectively $d \in \mathbb{L}$), this signals that d is the greatest (respectively least) term among the terms of the minimal weight.

6. $\ggg$ is a binary relation on Σ. This relation denotes the subset of the precedence relation computed so far.

Let w be a weight function on Σ, $\gg$ a precedence relation on Σ compatible with w, and $\succ$ the instance of the Knuth-Bendix order induced by $(w, \gg)$. A substitution σ grounding for a set of variables X is said to be *minimal for* X if for every variable $x \in X$ the term $\sigma(x)$ is of the minimal weight. We extend w to e by defining $w(e)$ to be the minimal weight of a constant of Σ.

We say that the pair $(w, \gg)$ is a *solution* to a state $(\mathbb{R}, \mathbb{M}, \mathbb{D}, \mathbb{U}, \mathbb{G}, \mathbb{L}, \ggg)$ if

1. For every tuple inequality $\langle l_1, \ldots, l_n \rangle > \langle r_1, \ldots, r_n \rangle$ in $\mathbb{R}$ and every substitution σ grounding for this tuple inequality and minimal for $\mathbb{M}$ we have $\langle l_1\sigma, \ldots, l_n\sigma \rangle \succ \langle r_1\sigma, \ldots, r_n\sigma \rangle$.

2. The weight function w solves every inequality in $\mathbb{D}$ in the following sense: replacement of each w_g by $w(g)$ gives a tautology. In addition, $w(e)$ coincides with the minimal weight $w(c)$ of constants $c \in \Sigma$.

3. If $\mathbb{U} = one$, then there exists exactly one term of the minimal weight.

4. If $d \in \mathbb{G}$ (respectively $d \in \mathbb{L}$) for some constant d, then d is the greatest (respectively least) term among the terms of the minimal weight. Note that if d is the greatest term of the minimal weight, then the signature contains no unary function symbol of the weight 0.

5. $\gg$ extends $\ggg$.

We will now show how to reduce the orientability problem for the systems of rewrite rules to the solvability problem for states.

Let R be a system of rewrite rules such that every two different rules in R have disjoint variables. Denote by $\mathbb{S}_R$ the state $(\mathbb{R}, \mathbb{M}, \mathbb{D}, \mathbb{U}, \mathbb{G}, \mathbb{L}, \ggg)$ defined as follows.

1. $\mathbb{R}$ consists of all tuple inequalities $\langle l \rangle > \langle r \rangle$ such that $l \to r$ belongs to R.

2. $\mathbb{M} = \emptyset$.

3. $\mathbb{D}$ consists of (a) all inequalities $w_g \geq 0$, where $g \in \Sigma$ is a non-constant; (b) the inequality $w_e > 0$ and all inequalities $w_d - w_e \geq 0$, where d is a constant of Σ.

4. $\mathbb{U} = any$.

5. $\mathbb{G} = \mathbb{L} = \emptyset$.

6. $\ggg$ is the empty binary relation on Σ.

LEMMA 4.1 *Let w be a weight function, $\gg$ a precedence relation on Σ compatible with w, and $\succ$ the instance of KBO induced by $(w, \gg)$. Then $\succ$ orients R if and only if $(w, \gg)$ is a solution to $\mathbb{S}_R$.* □

The proof is straightforward.

5 Trivial signatures

For technical reasons, we will distinguish two kinds of signatures. Essentially, our algorithm depends on whether the weights of terms are restricted or not. For the so-called *non-trivial* signatures, the weights are not restricted. When we present the orientability algorithm for the non-trivial signatures, we will use the fact that terms of sufficiently large weights always exist. For the trivial signatures we will present a simpler orientability algorithm in Section 7.

A signature is called *trivial* if it contains no function symbols of arity ≥ 2, and at most one unary function symbol. Note that a signature is non-trivial if and only if it contains either a function symbol of arity ≥ 2 or at least two function symbols of arity 1.

LEMMA 5.1 *Let Σ be a non-trivial signature and w be a weight function for Σ. Then for every integer m there exists a ground term of the signature Σ such that $|t| > m$.*

PROOF. It is enough to show how for every term t build a term of the weight greater than $|t|$. Note that the weight of any term is positive. If Σ contains a function symbol g of arity $n \geq 2$, then $|g(t, \ldots, t)| = w(g) + n \cdot |t| > |t|$. If Σ contains two unary function symbols, then for at least one of them g we have $w(g) > 0$. Then $|g(t)| = w(g) + |t| > |t|$. □

6 An algorithm for orientability in the case of non-trivial signatures

In this section we only consider non-trivial signatures. An algorithm for trivial signatures is given in Section 7. The algorithm given in this section will be illustrated below in Section 6.5 on the rewrite rule of Example 2.6.

Our algorithm works as follows. Given a system R of rewrite rules, we build the initial state $\mathbb{S}_R = (\mathbb{R}, \mathbb{M}, \mathbb{D}, \mathbb{U}, \mathbb{G}, \mathbb{L}, \ggg)$. Then we repeatedly transform $(\mathbb{R}, \mathbb{M}, \mathbb{D}, \mathbb{U}, \mathbb{G}, \mathbb{L}, \ggg)$ as described below. We call the *size* of the state the total number of occurrences of function symbols and variables in $\mathbb{R}$. Every transformation step will terminate with either success or failure, or else decrease the size of $\mathbb{R}$.

At each step we assume that $\mathbb{R}$ consists of k tuple inequalities

$$\begin{array}{c} \langle l_1, L_1 \rangle > \langle r_1, R_1 \rangle, \\ \cdots \\ \langle l_k, L_k \rangle > \langle r_k, R_k \rangle, \end{array} \tag{3}$$

such that all of the L_i, R_i are tuples of terms.

We will label parts of the algorithm, these labels will be used in the proof of its soundness. The algorithm can make a non-deterministic choice, but at most once, and the number of non-deterministic branches is bounded by the number of constants in Σ.

When the set $\mathbb{D}$ of linear inequalities changes, we assume that we check the new set for satisfiability, and terminate with failure if it is unsatisfiable. Likewise, when we change $\ggg$, we check if it can be extended to an order and terminate with failure if it cannot.

6.1 The algorithm

The algorithm works as follows. Every step consists of a number of state transformations, beginning with PREPROCESS defined below. During the algorithm, we will perform two kinds of *consistency* *checks*:

- The *consistency check on* $\mathbb{D}$ is the check if $\mathbb{D}$ has a solution. If it does not, we terminate with failure.
- The *consistency check on* $\ggg$ is the check if $\ggg$ can be extended to an order, i.e., the transitive closure $\gg$ of $\ggg$ is irreflexive, i.e., for no $g \in \Sigma$ we have $g \gg g$. If $\ggg$ cannot be extended to an order, we terminate with failure.

It is not hard to argue that both kinds of consistency checks can be performed in polynomial time. The consistency check on $\mathbb{D}$ is polynomial by Lemma 3.7. The consistency check on $\ggg$ is polynomial since the transitive closure of a binary relation can be computed in polynomial time, see e.g. [Cormen, Leiserson and Rivest 1991].

PREPROCESS. Do the following transformations while possible. If any tuple inequality in $\mathbb{R}$ has length 0, remove it from $\mathbb{R}$. If $\mathbb{R}$ contains a tuple inequality $\langle l_1, \ldots, l_n \rangle > \langle l_1, \ldots, l_n \rangle$, terminate with failure. Otherwise, if $\mathbb{R}$ contains a tuple inequality $\langle l, l_1, \ldots, l_n \rangle > \langle l, r_1, \ldots, r_n \rangle$, replace it by $\langle l_1, \ldots, l_n \rangle > \langle r_1, \ldots, r_n \rangle$.

If $\mathbb{R}$ becomes empty, proceed to TERMINATE, otherwise continue with MAIN.

MAIN. Now we can assume that in (3) each l_i is a term different from the corresponding term r_i. For every variable x and term t denote by $n(x,t)$ the number of occurrences of x in t. For example, $n(x, g(x, h(y, x))) = 2$. Likewise, for every function symbol $g \in \Sigma$ and term t denote by $n(g,t)$ the number of occurrences of g in t. For example, $n(h, g(x, h(y, x))) = 1$.

(M1) For all x and i such that $n(x, l_i) > n(x, r_i)$, add x to $\mathbb{M}$.

(M2) If for some i there exists a variable $x \notin \mathbb{M}$ such that $n(x, l_i) < n(x, r_i)$, then terminate with failure.

For every pair of terms l, r, denote by $W(l,r)$ the linear inequality obtained as follows. Let v_l and v_r be the numbers of occurrences of variables in l and r respectively. Then

$$W(l,r) = \sum_{g \in \Sigma} (n(g,l) - n(g,r)) w_g + (v_l - v_r) w_e \geq 0. \tag{4}$$

For example, if $l = h(x, f(y))$ and $r = f(g(x, g(x, y)))$, then

$$W(l,r) = w_h - 2 \cdot w_g - w_e \geq 0.$$

(M3) Add to $\mathbb{D}$ all the linear inequalities $W(l_i, r_i)$ for all i and perform the consistency check on $\mathbb{D}$.

Now compute $\mathbb{D}^=$. If $\mathbb{D}^=$ contains none of the inequalities $W(l_i, r_i)$, proceed to TERMINATE. Otherwise, for all i such that $W(l_i, r_i) \in \mathbb{D}^=$ apply the applicable case below, depending on the form of l_i and r_i.

(M4) If (l_i, r_i) has the form $(g(s_1, \ldots, s_n), h(t_1, \ldots, t_p))$, where g is different from h, then extend $\ggg$ by adding $g \ggg h$ and remove the tuple inequality $\langle l_i, L_i \rangle > \langle r_i, R_i \rangle$ from $\mathbb{R}$. Perform the consistency check on $\ggg$.

(M5) If (l_i, r_i) has the form $(g(s_1,\dots,s_n), g(t_1,\dots,t_n))$, then replace $\langle l_i, L_i\rangle > \langle r_i, R_i\rangle$ by $\langle s_1,\dots,s_n,L_i\rangle > \langle t_1,\dots,t_n,R_i\rangle$.

(M6) If (l_i, r_i) has the form (x, y), where x and y are different variables, do the following. (Note that at this point $x, y \in \mathbb{M}$.) If L_i is empty, then terminate with failure. Otherwise, set $\mathbb{U}$ to *one* and replace $\langle l_i, L_i\rangle > \langle r_i, R_i\rangle$ by $\langle L_i\rangle > \langle R_i\rangle$.

(M7) If (l_i, r_i) has the form (x, t), where t is not a variable, do the following. If t is not a constant, or L_i is empty, then terminate with failure. So assume that t is a constant c. If $\mathbb{L} = \{d\}$ for some d different from c, then terminate with failure. Otherwise, set $\mathbb{L}$ to $\{c\}$. Replace in L_i and R_i the variable x by c, obtaining L_i' and R_i' respectively, and then replace $\langle l_i, L_i\rangle > \langle r_i, R_i\rangle$ by $\langle L_i'\rangle > \langle R_i'\rangle$.

(M8) If (l_i, r_i) has the form (t, x), where t is not a variable, do the following. If t contains x, remove $\langle l_i, L_i\rangle > \langle r_i, R_i\rangle$ from $\mathbb{R}$. Otherwise, if t is a non-constant or L_i is empty, terminate with failure. (Note that at this point $x \in \mathbb{M}$ and $W(t, x) \in \mathbb{D}^=$.) Let now t be a constant c. If $\mathbb{G} = \{d\}$ for some d different from c, then terminate with failure. Otherwise, set $\mathbb{G}$ to $\{c\}$. Replace in L_i and R_i the variable x by c, obtaining L_i' and R_i' respectively, and then replace $\langle l_i, L_i\rangle > \langle r_i, R_i\rangle$ by $\langle L_i'\rangle > \langle R_i'\rangle$.

After this step repeat PREPROCESS.

TERMINATE. Let $(\mathbb{R}, \mathbb{M}, \mathbb{D}, \mathbb{U}, \mathbb{G}, \mathbb{L}, \ggg)$ be the current state. Do the following.

(T1) If $d \in \mathbb{G}$, then for all constants c different from d such that $w_c - w_e \geq 0$ belongs to $\mathbb{D}^=$ extend $\ggg$ by adding $d \ggg c$. Likewise, if $c \in \mathbb{L}$, then for all constants d different from c such that $w_d - w_e \geq 0 \in \mathbb{D}^=$ extend $\ggg$ by adding $d \ggg c$. Perform the consistency check on $\ggg$.

(T2) For all f in Σ do the following. If f is a unary function symbol and $w_f \geq 0$ belongs to $\mathbb{D}^=$, then extend $\ggg$ by adding $f \ggg h$ for all $h \in \Sigma - \{f\}$. Perform the consistency check on $\ggg$. If $\mathbb{U} = one$ or $\mathbb{G} \neq \emptyset$, then terminate with failure.

(T3) If there exists no constant c such that $w_c - w_e \geq 0$ is in $\mathbb{D}^=$, then non-deterministically choose a constant $c \in \Sigma$, add $w_e - w_c \geq 0$ to $\mathbb{D}$, perform the consistency check on $\mathbb{D}$ and repeat PREPROCESS.

(T4) If $\mathbb{U} = one$, then terminate with failure if there exists more than one constant c such that $w_c - w_e \geq 0$ belongs to $\mathbb{D}^=$.

(T5) Terminate with success.

We will show how to build a solution at step (T5) below in Lemma 6.19.

6.2 Correctness

In this section we prove correctness of the algorithm. In Section 6.3 we show how to find a solution when the algorithm terminates with success. The correctness will follow from a series of lemmas asserting that the transformation steps performed by the algorithm preserve the set of solutions. We will use notation and terminology of the algorithm. We say that a step of the algorithm is *correct* if the set of solutions to the state before this step coincides with the set of solutions after the step.

When we prove correctness of a particular step, we will always denote by $\mathbb{S} = (\mathbb{R}, \mathbb{M}, \mathbb{D}, \mathbb{U}, \mathbb{G}, \mathbb{L}, \ggg)$ the state before this step, and by $\mathbb{S}'$ the state after this step. When we use substitutions in the proof, we always assume that the substitutions are grounding for the relevant terms.

The following two lemmas can be proved by a straightforward application of the definition of solution to a state.

LEMMA 6.1 (consistency check) *If consistency check on $\mathbb{D}$ or on $\ggg$ terminates with failure, then $\mathbb{S}$ has no solution.* □

LEMMA 6.2 *Step* PREPROCESS *is correct.* □

Let us now analyze MAIN. For every weight function w and precedence relation $\gg$ compatible with w we call a *counterexample* to $\langle l_i, L_i\rangle > \langle r_i, R_i\rangle$ w.r.t. $(w, \gg)$ any substitution σ minimal for $\mathbb{M}$ such that $\langle r_i\sigma, R_i\sigma\rangle \succeq \langle l_i\sigma, L_i\sigma\rangle$ for the order $\succ$ induced by $(w, \gg)$.

Denote by $\mathbb{S}^{-i}$ the state obtained from $\mathbb{S}$ by removal of the ith tuple inequality $\langle l_i, L_i\rangle > \langle r_i, R_i\rangle$ from $\mathbb{R}$. The following lemma follows immediately from the definition of solution.

LEMMA 6.3 (counterexample) *If for every solution $(w, \gg)$ to $\mathbb{S}^{-i}$ there exists a counterexample to $\langle l_i, L_i\rangle > \langle r_i, R_i\rangle$ w.r.t. $(w, \gg)$, then $\mathbb{S}$ has no solution. If for every solution $(w, \gg)$ to $\mathbb{S}^{-i}$ there exists no counterexample to the tuple inequality $\langle l_i, L_i\rangle > \langle r_i, R_i\rangle$, then removing this tuple inequality from $\mathbb{R}$ does not change the set of solutions to $\mathbb{S}$.* □

This lemma means that we can change $\langle l_i, L_i\rangle > \langle r_i, R_i\rangle$ into a different tuple inequality or change $\mathbb{M}$, if we can prove that this change does not influence the existence of a counterexample.

Let σ be a substitution, x a variable and t a term. Denote by σ_x^t the substitution defined by

$$\sigma_x^t(y) = \begin{cases} \sigma(y), & \text{if } y \neq x, \\ t, & \text{if } y = x. \end{cases}$$

LEMMA 6.4 *Let w be a weight function on Σ and $\gg$ a precedence relation on Σ compatible with w. Suppose also that for some x and i we have $n(x, l_i) > n(x, r_i)$ and there exists a counterexample σ to $\langle l_i, L_i\rangle > \langle r_i, R_i\rangle$ w.r.t. $(w, \gg)$. Then there exists a counterexample σ' to $\langle l_i, L_i\rangle > \langle r_i, R_i\rangle$ w.r.t. $(w, \gg)$ minimal for $\{x\}$.*

PROOF. Suppose that σ is not minimal for $\{x\}$. Denote by c a minimal constant w.r.t. w and by t the term $x\sigma$. Since σ is not minimal for $\{x\}$, we have $|t| > |c|$. Consider the substitution σ_x^c. Since σ is a counterexample, we have $|r_i\sigma| \geq |l_i\sigma|$. We have

$$\begin{array}{rcl} |l_i\sigma_x^c| & = & |l_i\sigma| - n(x, l_i)\cdot(|t| - |c|); \\ |r_i\sigma_x^c| & = & |r_i\sigma| - n(x, r_i)\cdot(|t| - |c|). \end{array}$$

Then

$$\begin{array}{rcccl} |r_i\sigma_x^c| & = & |r_i\sigma| - n(x, r_i)\cdot(|t| - |c|) & \geq & |l_i\sigma| - n(x, r_i)\cdot(|t| - |c|) \\ & > & |l_i\sigma| - n(x, l_i)\cdot(|t| - |c|) & = & |l_i\sigma_x^c|. \end{array}$$

Therefore, $|r_i\sigma_x^c| > |l_i\sigma_x^c|$, and so σ_x^c is a counterexample too. □

One can immediately see that this lemma implies correctness of step (M1).

LEMMA 6.5 *Step (M1) is correct.*

Proof. Evidently, every solution to $\mathbb{S}$ is also a solution to $\mathbb{S}'$. But by Lemma 6.4, every counterexample to $\mathbb{S}$ can be turned into a counterexample to $\mathbb{S}'$, so every solution to $\mathbb{S}'$ is also a solution to $\mathbb{S}$. □

Let us now turn to step (M2).

Lemma 6.6 (M2) *If for some i and $x \notin \mathbb{M}$ we have $n(x, l_i) < n(x, r_i)$, then $\mathbb{S}$ has no solution. Therefore, step (M2) is correct.*

Proof. We show that for every $(w, \gg)$ there exists a counterexample to $\langle l_i, L_i \rangle > \langle r_i, R_i \rangle$ w.r.t. $(w, \gg)$. Let σ be any substitution grounding for this tuple inequality. Take any term t and consider the substitution σ_x^t. We have

$$|r_i \sigma_x^t| - |l_i \sigma_x^t| = |r_i \sigma| - |l_i \sigma| + (n(x, r_i) - n(x, l_i)) \cdot (|t| - |x\sigma|).$$

By Lemma 5.1 there exist terms of an arbitrarily large weight, so for a term t of a large enough weight we have $|r_i \sigma_x^t| > |l_i \sigma_x^t|$, and so σ_x^t is a counterexample to $\langle l_i, L_i \rangle > \langle r_i, R_i \rangle$.

Correctness of (M2) is straightforward. □

Note that after step (M2) for all i and $x \notin \mathbb{M}$ we have $n(x, l_i) = n(x, r_i)$.

Denote by Θ_c the substitution such that $\Theta_c(x) = c$ for every variable x.

Lemma 6.7 (M3) *Let for all i and $x \notin \mathbb{M}$ we have $n(x, l_i) = n(x, r_i)$. Every solution $(w, \gg)$ to $\mathbb{S}$ is also a solution to $W(l_i, r_i)$. Therefore, step (M3) is correct.*

Proof. Let c be a constant of the minimal weight. Consider the substitution Θ_c. Note that this substitution is minimal for $\mathbb{M}$. It follows from the definition of W that $(w, \gg)$ is a solution to $W(l_i, r_i)$ if and only if $|l_i \Theta_c| \geq |r_i \Theta_c|$. But $|l_i \Theta_c| \geq |r_i \Theta_c|$ is a straightforward consequence of the definition of solutions to tuple inequalities.

Correctness of (M3) is straightforward. □

Lemma 6.8 *Let for all $x \notin \mathbb{M}$ we have $n(x, l_i) = n(x, r_i)$. Let also $W(l_i, r_i) \in \mathbb{D}^=$. Then for every solution to $\mathbb{S}^{-i}$ and every substitution σ minimal for $\mathbb{M}$ we have $|l_i \sigma| = |r_i \sigma|$.*

Proof. Using the fact that $n(x, l_i) = n(x, r_i)$ for all $x \notin \mathbb{M}$, it is not hard to argue that $|l_i \sigma| - |r_i \sigma|$ does not depend on σ, whenever σ is minimal for $\mathbb{M}$.

Let c be a constant of the minimal weight. It follows from the definition of W that if $W(l_i, r_i) \in \mathbb{D}^=$, then for every solution to $\mathbb{D}$ (and so for every solution to $\mathbb{S}^{-i}$) we have $|l_i \Theta_c| = |r_i \Theta_c|$. Therefore, $|l_i \sigma| = |r_i \sigma|$ for all substitutions σ minimal for $\mathbb{M}$. □

The proof of correctness of steps (M4)–(M8) will use this lemma in the following way. A pair $(w, \gg)$ is a solution to $\mathbb{S}$ if and only if it is a solution to $\mathbb{S}^{-i}$ and a solution to $\langle l_i, L_i \rangle > \langle r_i, R_i \rangle$. Equivalently, $(w, \gg)$ is a solution to $\mathbb{S}$ if and only if it is a solution to $\mathbb{S}^{-i}$ and for every substitution σ minimal for $\mathbb{M}$ we have $\langle l_i \sigma, L_i \sigma \rangle \succ \langle r_i \sigma, R_i \sigma \rangle$. But by Lemma 6.8 we have $|l_i \sigma| = |r_i \sigma|$, so $\langle l_i \sigma, L_i \sigma \rangle \succ \langle r_i \sigma, R_i \sigma \rangle$ must be satisfied by either condition 2 or condition 3 of the definition of the KBO order.

This consideration can be summarized as follows.

LEMMA 6.9 *Let for all* $x \notin \mathbb{M}$ *we have* $n(x, l_i) = n(x, r_i)$. *Let also* $W(l_i, r_i) \in \mathbb{D}^=$. *Then a pair* $(w, \gg)$ *is a solution to* $\mathbb{S}$ *if and only if it is a solution to* $\mathbb{S}^{-i}$ *and for every substitution* σ *minimal for* $\mathbb{M}$ *the following holds. Let* $l_i\sigma = g(t_1, \ldots, t_n)$ *and* $r_i\sigma = h(s_1, \ldots, s_p)$. *Then at least one of the following conditions holds*

1. $l_i\sigma = r_i\sigma$ *and* $L_i\sigma \succ R_i\sigma$; *or*
2. $g \gg h$; *or*
3. $g = h$ *and for some* $1 \leq i \leq n$ *we have* $t_1\sigma = s_1\sigma, \ldots, t_{i-1}\sigma = s_{i-1}\sigma$ *and* $t_i\sigma \succ s_i\sigma$. □

LEMMA 6.10 *Step (M4) is correct.*

PROOF. We know that $l_i = g(s_1, \ldots, s_n)$ and $r_i = h(t_1, \ldots, t_p)$ for $g \neq h$. Take any substitution σ minimal for $\mathbb{M}$. Obviously, $l_i\sigma = r_i\sigma$ is impossible, so $\langle l_i, L_i\rangle\sigma \succ \langle r_i, R_i\rangle\sigma$ if and only if $l_i\sigma \succ r_i\sigma$. By Lemma 6.9 this holds if and only if $g \gg h$, so step (M4) is correct. □

LEMMA 6.11 *Step (M5) is correct.*

PROOF. We know that $l_i = g(s_1, \ldots, s_n)$ and $r_i = g(t_1, \ldots, t_n)$. Note that due to PREPROCESS, $l_i \neq r_i$, so $n \geq 1$. It follows from Lemma 6.9 that $\langle l_i, L_i\rangle\sigma \succ \langle r_i, R_i\rangle\sigma$ if and only if $\langle s_1, \ldots, s_n, L_i\rangle\sigma \succ \langle t_1, \ldots, t_n, R_i\rangle\sigma$, so step (M5) is correct. □

LEMMA 6.12 *Step (M6) is correct.*

PROOF. We know that $l_i = x$ and $r_i = y$, where x, y are different variables. Note that if L_i is empty, then the substitution Θ_c, where c is of the minimal weight, is a counterexample to $\langle x, L_i\rangle > \langle y, R_i\rangle$. So assume that L_i is non-empty and consider two cases.

1. If there exist at least two terms s, t of the minimal weight, then there exists a counterexample to $\langle x, L_i\rangle > \langle y, R_i\rangle$. Indeed, if $s \succ t$, then $y\sigma \succ x\sigma$ for every σ such that $\sigma(x) = t$ and $\sigma(y) = s$.
2. If there exists exactly one term t of the minimal weight, then $x\sigma = y\sigma$ for every σ minimal for $\mathbb{M}$. Therefore, $\langle x, L_i\rangle > \langle y, R_i\rangle$ is equivalent to $\langle L_i\rangle > \langle R_i\rangle$.

In either case it is not hard to argue that step (M6) is correct. □

LEMMA 6.13 *Step (M7) is correct.*

PROOF. We know that $l_i = x$ and $r_i = t$. Let c be the least constant in the signature. If $t \neq c$, then Θ_c is obviously a counterexample to $\langle x, L_i\rangle > \langle t, R_i\rangle$. Otherwise $t = c$, then for every counterexample σ we have $\sigma(x) = c$. In either case it is not hard to argue that step (M7) is correct. □

LEMMA 6.14 *Step (M8) is correct.*

PROOF. We know that $l_i = t$ and $r_i = x$. Note that $t \neq x$ due to the PREPROCESS step, so if x occurs in t we have $t\sigma \succ x\sigma$ for all σ. Assume now that x does not occur in t. Then $x \in \mathbb{M}$. Consider two cases.

1. t is a non-constant. For every substitution σ minimal for $\mathbb{M}$ we have $|t\sigma| = |x\sigma|$, hence $t\sigma$ is a non-constant term of the minimal weight. This implies that the signature contains a unary function symbol f of the weight 0. Take any substitution σ. It is not hard to argue that $\sigma_x^{f(t)\sigma}$ is a counterexample to $\langle t, L_i \rangle > \langle x, R_i \rangle$.

2. t is a constant c. Let d be the greatest constant in the signature among the constants of the minimal weight. If $d \neq c$, then Θ_d is obviously a counterexample to $\langle c, L_i \rangle > \langle x, R_i \rangle$. Otherwise $d = c$, then for every counterexample σ we have $\sigma(x) = c$.

In either case it is not hard to argue that step (M8) is correct. $\square$

Let us now analyze steps TERMINATE. Note that for every constant c the inequality $w_c - w_e \geq 0$ belongs to $\mathbb{D}$ and for every function symbol g the inequality $w_g \geq 0$ belongs to $\mathbb{D}$ too.

LEMMA 6.15 *Step (T1) is correct.*

PROOF. Suppose $d \in \mathbb{G}$, $c \neq d$, and $w_c - w_e \geq 0$ belongs to $\mathbb{D}^=$. Then for every solution to $\mathbb{S}$ we have $w(c) = w(e)$, and therefore c is a constant of the minimal weight. But since for every solution d is the greatest constant among those having the minimal weight, we must have $d \gg c$.

The case $c \in \mathbb{L}$ is similar. $\square$

LEMMA 6.16 *Step (T2) is correct.*

PROOF. If f is a unary function symbol and $w_f \geq 0$ belongs to $\mathbb{D}^=$, then for every solution $w(f) = 0$. By the definition of the KBO we must have $f \gg g$ for all $g \in \Sigma - \{f\}$. But then (i) there exists an infinite number of terms of the minimal weight and (ii) a constant $d \in \mathbb{G}$ cannot be the greatest term of the minimal weight (since for example $f(d) \succ d$ and $|f(d)| = |d|$). $\square$

Step (T3) makes a non-deterministic choice, which can result in several states $\mathbb{S}_1, \ldots, \mathbb{S}_n$. We say that such a step is correct if the set of solutions to $\mathbb{S}$ is the union of the sets of solutions to $\mathbb{S}_1, \ldots, \mathbb{S}_n$.

LEMMA 6.17 *Step (T3) is correct.*

PROOF. Note that w is a solution to $w_e - w_c \geq 0$ if and only if $w(c)$ is the minimal weight, so addition of $w_e - w_c \geq 0$ to $\mathbb{D}$ amounts to stating that c has the minimal weight. Evidently, for every solution, there must be a constant c of the minimal weight, so the step is correct. $\square$

LEMMA 6.18 *Step (T4) is correct.*

PROOF. Suppose $\mathbb{U} = one$, then for every solution there exists a unique term of the minimal weight. If, c is a constant such that $w_c - w_e \geq 0$ belongs to $\mathbb{D}^=$, then c must be a term of the minimal weight. Therefore, there cannot be more than one such a constant c. $\square$

6.3 Extracting a solution

In this section we will show how to find a solution when the algorithm terminates with success.

LEMMA 6.19 *Step (T5) is correct.*

PROOF. To prove correctness of (T5) we have to show the existence of solution. In fact, we will show how to build a particular solution.

Note that when we terminate at step (T5), the system $\mathbb{D}$ is solvable, since it was solvable initially and we performed consistency checks on every change of $\mathbb{D}$.

By Lemma 3.5 there exists an integer solution w to $\mathbb{D}$ which is also a solution to the strict versions of every inequality in $\mathbb{D} - \mathbb{D}^{=}$. Likewise, there exists a linear order $\gg$ extending $\ggg$, since we performed consistency checks on every change of $\ggg$. We claim that $(w, \gg)$ is a solution to $(\mathbb{R}, \mathbb{M}, \mathbb{D}, \mathbb{U}, \mathbb{G}, \mathbb{L}, \ggg)$. To this end we have to show that w is weight function, $\gg$ is compatible with w and all items 1–5 of the definition of solution are satisfied.

Let us first show that w is a weight function. Note that $\mathbb{D}$ contains all inequalities $w_g \geq 0$, where $g \in \Sigma$ is a non-constant, the inequality $w_e > 0$ and the inequalities $w_d - w_e \geq 0$ for every constant $d \in \Sigma$. So to show that w is a weight function it remains to show that at most one unary function symbol f has weight 0. Indeed, if there were two such function symbols f_1 and f_2, then at step (T2) we would add both $f_1 \ggg f_2$ and $f_2 \ggg f_1$, but the following consistency check on $\ggg$ would fail.

The proof that $\gg$ is compatible with w is similar.

Denote by $\succ$ the instance of KBO order induced by $(w, \gg)$.

1. *For every tuple inequality $\langle l_i, L_i \rangle > \langle r_i, R_i \rangle$ in $\mathbb{R}$ and every substitution σ minimal for $\mathbb{M}$ we have $\langle l_i\sigma, L_i\sigma \rangle \succ \langle r_i\sigma, R_i\sigma \rangle$.* In the proof we will use the fact that $w(e)$ is the minimal weight.

   By step (M3), the inequality $W(l_i, r_i)$ does not belong to $\mathbb{D}^{=}$ (otherwise $\langle l_i, L_i \rangle > \langle r_i, R_i \rangle$ would be removed at one of steps (M4)–(M8)). It follows from the definition of W that if $W(l_i, r_i) \in \mathbb{D} - \mathbb{D}^{=}$, then $|l_i\Theta_c| > |r_i\Theta_c|$, where c is any constant of the minimal weight. In Lemma 6.8 we proved that $|l_i\sigma| - |r_i\sigma|$ does not depend on σ, whenever σ is minimal for $\mathbb{M}$. Therefore, $|l_i\sigma| > |r_i\sigma|$ for all substitutions σ minimal for $\mathbb{M}$.

2. *The weight function w solves every inequality in $\mathbb{D}$ and $w(e)$ coincides with the minimal weight.* This follows immediately from our construction, if we show that $w(e)$ is the minimal weight. Let us show that w_e is the minimal weight. Indeed, since $\mathbb{D}$ initially contains the inequalities $w_c - w_e \geq 0$ for all constants c, we have that $w(e)$ is less than or equal to the minimal weight. By step (T3), there exists a constant c such that $w_c - w_e \geq 0$ is in $\mathbb{D}^{=}$, hence $w(c) = w(e)$, and so $w(e)$ is greater than or equal to the minimal weight.

3. *If $\mathbb{U} = one$, then there exists exactly one term of the minimal weight.* Assume $\mathbb{U} = one$. We have to show that (i) there exists no unary function symbol f of weight 0 and (ii) there exists exactly one constant of the minimal weight. Let f be a unary function symbol. By our construction, $w_f \geq 0$ belongs to $\mathbb{D}$. By step (T2) $w_f \geq 0$ does not belong to $\mathbb{D}^{=}$, so by the definition of w we have $w(f) > 0$. By our construction, $w_c - w_e \geq 0$ belongs to $\mathbb{D}$ for every constant c. By step (T4), at most one of such inequalities belongs to $\mathbb{D}^{=}$. But if $w_c - w_e \geq 0$ does not belong to $\mathbb{D}^{=}$, then $w(c) - w(e) > 0$ by the construction of w. Therefore, there exists at most one constant of the minimal weight.

4. *If $d \in \mathbb{G}$ (respectively $d \in \mathbb{L}$) for some constant d, then d is the greatest (respectively least) term among the terms of the minimal weight.* We consider the case $d \in \mathbb{G}$, the case $d \in \mathbb{L}$ is similar. But by step (T2) there is no unary function symbol f such that $w_f \geq 0$ belongs to $\mathbb{D}^=$, therefore $w(f) > 0$ for all unary function symbols f. This implies that only constants may have the minimal weight. But by step (T1) and the definition of w, for all constants c of the minimal weight we have $d \ggg c$, and hence also $d \gg c$.

5. *$\gg$ extends $\ggg$.* This follows immediately from our construction.

□

6.4 Time complexity

Provided that we use a polynomial-time algorithm for solving homogeneous linear inequalities, and a polynomial-time algorithm for transitive closure, we can prove the following lemma.

Lemma 6.20 *The algorithm runs in time polynomial of the size of the system of rewrite rules.*

Proof. Note that the algorithm makes polynomial number of steps. Indeed, initially the size of $\mathbb{R}$ is $O(n \log n)$ of the size of the system of rewrite rules (and can even be made linear, if we avoid renaming variables). Each of the steps (M4)–(M8) decreases the size of $\mathbb{R}$. The algorithm can make a non-deterministic choice, but at most once, and the number of non-deterministic branches is bounded by the number of constants, so it is linear in the size of the original system.

We proved that the number of steps is polynomial in the size of the input. It remains to prove that every step can be made in polynomial time of the size of a state and that the size of every state is polynomial in the size of the input.

Solvability of $\mathbb{D}$ can be checked in polynomial time by Lemma 3.7. The system $\mathbb{D}^=$ can be built in polynomial time by the same lemma. The relation $\ggg$ can be extended to an order if and only if the transitive closure $\ggg'$ of $\ggg$ is irreflexive, i.e., there is no g such that $g \ggg' g$. The transitive closure can be built in polynomial time. The check for irreflexivity can be obviously done in polynomial time too. Therefore, every step can be performed in polynomial time of the size of the state.

It remains to show that the size of $\mathbb{S}$ is bound by a polynomial. The only part of $\mathbb{S}$ that is not immediately seen to be polynomial is $\mathbb{D}$. However, it is not hard to argue that the number of equations in $\mathbb{S}$ of the form $W(l,r)$ is bound by the size of the input, and every equation obviously has a polynomial size. It is also easy to see that the size of the remaining equations is polynomial too. □

6.5 A simple example

Let us consider how the algorithm works on the rewrite rule $g(x,a,b) \to g(b,b,a)$ of Example 2.6. Initially, $\mathbb{R}$ consists of one tuple inequality

$$\langle g(x,a,b) \rangle > \langle g(b,b,a) \rangle \tag{5}$$

and $\mathbb{D}$ consists of the following linear inequalities:

$$w_g \geq 0, \quad w_e > 0, \quad w_a - w_e \geq 0, \quad w_b - w_e \geq 0.$$

At step (M1) we note that $n(x, g(x, a, b)) = 1 > 0 = n(x, g(b, b, a))$. Therefore, we add x to $\mathbb{M}$.

At step (M3) we add the linear inequality $w_e - w_b \geq 0$ to $\mathbb{D}$ obtaining

$$w_g \geq 0, \quad w_e > 0, \quad w_a - w_e \geq 0, \quad w_b - w_e \geq 0, \quad w_e - w_b \geq 0.$$

Now we compute $\mathbb{D}^=$. It consists of two equations $w_b - w_e \geq 0$ and $w_e - w_b \geq 0$, so we have to apply one of the steps (M4)–(M8), in this case the applicable step is (M5). We replace (5) by

$$\langle x, a, b \rangle > \langle b, b, a \rangle. \tag{6}$$

At the next iteration of step (M3) we should add to $\mathbb{D}$ the linear inequality $w_e - w_b \geq 0$, but this linear inequality is already a member of $\mathbb{D}$, and moreover a member of $\mathbb{D}^=$. So we proceed to step (M7). At this step we set $\mathbb{L} = \{b\}$ and replace (6) by

$$\langle a, b \rangle > \langle b, a \rangle. \tag{7}$$

Then at step (M2) we add $w_a - w_b \geq 0$ to $\mathbb{D}$ obtaining

$$w_g \geq 0, \quad w_e > 0, \quad w_a - w_e \geq 0, \quad w_b - w_e \geq 0, \quad w_e - w_b \geq 0, \quad w_a - w_b \geq 0.$$

Now $w_a - w_b \geq 0$ does not belong to the degenerate subsystem of $\mathbb{D}$, so we proceed to TERMINATE. Steps (T1)–(T4) change neither $\mathbb{D}$ nor $\ggg$, so we terminate with success.

Solutions extracted according to Lemma 6.19 will be any pairs $(w, \gg)$ such that $w(a) > w(b)$. Note that these are *not all* solutions. There are also solutions such that $w(a) = w(b)$ and $a \gg b$. However, if we try to find a description of all solutions we cannot any more guarantee that the algorithm runs in polynomial time.

7 Orientability for trivial signatures

Consider a trivial signature which consists of a unary function symbol g and some constants. Let R be a system of rewrite rules in this signature. If some rule in R has the form $t \to g^n(x)$ such that x does not occur in t, then the system is evidently not orientable. If R contains no such rule, then R can be replaced by an equally orientable ground system, as the following lemma shows.

LEMMA 7.1 *Let R be a system of rewrite rules in a trivial signature Σ such that no rule in R contains a variable occurring in its right-hand side but not the left-hand side. Define the ground system R' obtained from R by the following transformations:*

1. *Replace every rule $g^m(x) \to g^n(d)$ in R by all rules $g^m(c) \to g^n(d)$ such that c is a constant in Σ.*
2. *For every rule $g^m(x) \to g^n(x)$ in R, if $m > n$ then remove this rule, otherwise terminate with failure.*

Then an instance of KBO $\succ$ orients R if and only if it orients R'. □

We leave the proof of this lemma to the reader. Note that the size of R' in the lemma is polynomial in the sum of the sizes of R and Σ. Therefore, we can restrict ourselves to ground systems.

Moreover, we can assume that for every rule in R' the function symbol g never occurs in both left-hand side and right-hand side of R. Indeed, this can be achieved by replacing every rewrite rule $g(s) \to g(t)$ in R' by $s \to t$ until g occurs in at most one side of the rule. Evidently, we can assume that R' contains no trivial rules $c \to c$. So we obtain a system consisting of rules $g^n(c) \to d$, $c \to g^n(d)$, where $n > 0$, or $c \to d$ such that c, d are different constants. In other words, for every rule $l \to r$ in R' the outermost symbol of l is different from the outermost symbol of r.

In order to check orientability of R', consider the system of homogeneous linear inequalities $\mathbb{D}$ which consists of

1. the inequalities $w_c > 0$ for all constants $c \in \Sigma$ and the inequality $w_g \geq 0$;

2. for every rule $l \to r$ in R' the inequalities $W(l,r) = \sum_{h \in \Sigma}(n(h,l) - n(h,r))w_h \geq 0$.

Evidently, $\mathbb{D}$ can be built in time polynomial in the size of R'. Evidently, if $\mathbb{D}$ is unsatisfiable, then R' is not orientable. If $\mathbb{D}$ is satisfiable, let $\mathbb{D}^=$ be the degenerate subsystem of $\mathbb{D}$. Let us build a binary relation $\ggg$ on Σ as follows:

1. for every rule $l \to r$ in R', if $W(l,r) \in \mathbb{D}^=$, then we take the outermost symbols h_1 and h_2 of l and r respectively and add $h_1 \ggg h_2$ to $\ggg$;

2. if $w_g \geq 0$ belongs to $\mathbb{D}^=$, then add $g \ggg c$ to $\ggg$ for all constants $c \in \Sigma$.

We leave it to the reader to check that R' is orientable if and only if $\ggg$ can be extended to a linear order. We can prove in the same way as before, that the check for orientability of R' can be done in polynomial time.

8 The problem of orientability by the KBO is P-complete

In Section 6.4 we have shown that the orientability problem can be solved in polynomial time. In this section we show that this problem is P-complete, and moreover it is P-hard even for ground rewrite systems. To this end, we reduce the *circuit value problem* which is known to be P-complete (see e.g., [Papadimitriou 1994]), to the orientability problem. Our reduction consists of two steps:

1. we reduce the problem of solving systems of linear inequalities $AX \geq \mathbf{0}$, $X > \mathbf{0}$, where A is an integer matrix, to the orientability problem;

2. we reduce the circuit value problem to solvability of such systems.

In the systems of linear inequalities, we assume all coefficients to be written in the unary notation. Both reductions will be LOGSPACE.

Let $AX \geq \mathbf{0}$ be a system of linear inequalities and we are looking for strictly positive solutions to it. For every variable x_i in the system we introduce a unary function symbol f_i. We consider the signature Σ consisting of all such symbols f_i, two unary symbols g, h, and a constant c. We will construct a ground rewrite rule system R whose orientability will be equivalent to the existence of a solution to $AX \geq \mathbf{0}, X > \mathbf{0}$ as follows. First of all, R contains the rewrite rule

$$ghc \to hggc.$$

An instance of KBO with parameters $(w, \gg)$ orients this rule if and only if $w(g) = 0$ (and hence also $g \gg h$). For each linear inequality I in the system, we add to R a rewrite rule $r(I)$, which will be demonstrated by an example (in order to avoid double indices). Suppose, for example, that the inequality can be rewritten in the form

$$a_1x_1 + \ldots + a_kx_k \geq a_{k+1}x_{k+1} + \ldots + a_nx_n. \tag{8}$$

where $x_1, \ldots, x_n$ are different variables and $a_1, \ldots, a_n, b_1, \ldots, b_n$ are non-negative coefficients. Then $r(I)$ has the form

$$ghf_1^{a_1} \cdots f_k^{a_k}c \to hgf_{k+1}^{a_{k+1}} \cdots f_n^{a_n}c \tag{9}$$

Note that for every solution we must have $w(f_i) > 0$ since there may be at most one function symbol of the weight 0. For every weight function w consider the substitution s of integers to variables such that $w(f_i) = s(x_i)$ and let $\gg$ be an arbitrary precedence relation such that g is maximal w.r.t. $\gg$. We leave it to the reader to check that $(w, \gg)$ is a solution to R if and only if s is a solution to $AX \geq \mathbf{0}, X > \mathbf{0}$.

It is not hard to argue that the reduction of A to R is LOGSPACE, provided that the coefficients of the linear inequations are written in the unary notation.

Let us now describe a reduction of the circuit value problem to the problem of whether a given system of linear integer inequalities has a positive solution. Consider a circuit with gates $g_1, \ldots, g_n$. For each gate g_i we introduce a new numerical variable x_i. We will also use an auxiliary numerical variable y. We construct a system of linear integer inequalities $\mathbb{D}$ in such a way that the circuit has the value *TRUE* if and only if $\mathbb{D}$ has a positive solution. For each gate g_i we introduce a system of numerical constraints $\mathbb{D}_i$ in the following way. If g_i is a *FALSE* gate then $\mathbb{D}_i$ is $\{x_i = y\}$, likewise if g_i is a *TRUE* gate then $\mathbb{D}_i$ is $\{x_i = 2y\}$. If g_i is a *NOT* gate with an input g_j then $\mathbb{D}_i$ is $\{x_i = 3y - x_j\}$. If g_i is an *AND* gate with inputs g_j and g_k then $\mathbb{D}_i$ is $\{y \leq x_i \leq 2y,\ x_i \leq x_j,\ x_i \leq x_k,\ x_j + x_k - 2y \leq x_i\}$. Let $\mathbb{D}'$ be the union of all $\mathbb{D}_i$ for $1 \leq i \leq n$. It is straightforward to check that for every positive solution to the system $\mathbb{D}'$ each variable x_i has the value of the variable y or twice that value, moreover it has the value of y if and only if the gate g_i has the value *FALSE*. To complete the construction we obtain $\mathbb{D}$ by adding to $\mathbb{D}'$ an equation $x_n = 2y$. Note that the coefficients of $\mathbb{D}$ are small, so they can be considered as written in the unary notation.

We have shown how to reduce the circuit value problem to the orientability problem. It is clear that all reductions can be done by a logarithmic-space algorithm.

9 Solving constraints consisting of a single inequality

In [Korovin and Voronkov 2000*b*] it is shown that the problem of solving the Knuth-Bendix ordering constraints is NP-complete. Let us show that the problem of solving the Knuth-Bendix ordering constraints consisting of a single inequality can be solved in polynomial time. Let us fix an instance of KBO on ground terms, i.e., a precedence relation on the signature Σ and a weight function w. Our problem is to decide for a given pair of terms s and t whether there exists a grounding substitution σ such that $s\sigma \succ t\sigma$. Since every instance of the Knuth-Bendix order is total on ground terms our problem is equivalent to the following problem: for a given pair of terms t and s decide whether for every grounding substitutions σ, $t\sigma \succeq s\sigma$ holds. The algorithm we present is similar to the algorithm for the orientability. The main difference is that there is no need to solve systems of linear inequalities for this problem. Since the order is given, we can use a simpler version of the notion of state $\mathbb{S} = (\mathbb{R}, \mathbb{M})$, where $\mathbb{R}$ is a single tuple inequality and $\mathbb{M}$ is a set of variables. Instead of tuple inequalities $\langle L \rangle > \langle R \rangle$ we will consider a new kind of tuple inequalities $\langle L \rangle \geq \langle R \rangle$ with a natural interpretation. Initially $\mathbb{R}$ consists of the tuple inequality $\langle t \rangle \geq \langle s \rangle$ and $\mathbb{M} = \emptyset$. Let e denote the constant that is the minimal term w.r.t. $\succ$. Instead of using the inequality $W(l, r)$, we will use the inequality $W'(l,r) = \sum_{g\in\Sigma}(n(g,l) - n(g,r))w(g) + (v_l - v_r)w(e) \geq 0$, where v_l and v_r are the numbers of occurrences of variables in l and r respectively. Let us present the algorithm.

PREPROCESS. Do the following transformations while possible. If $\mathbb{R}$ has the form $\langle\rangle \geq \langle\rangle$, then terminate with success. If $\mathbb{R}$ consists of a tuple inequality $\langle l, l_1, \ldots, l_n\rangle \geq \langle l, r_1, \ldots, r_n\rangle$, replace it by $\langle l_1, \ldots, l_n\rangle \geq \langle r_1, \ldots, r_n\rangle$.

MAIN. Now we can assume that $\mathbb{R}$ consists of a tuple $\langle l, L\rangle \geq \langle r, R\rangle$ and the term l is different from the term r.

(M1) For all x such that $n(x,l) > n(x,r)$, add x to $\mathbb{M}$.

(M2) If there exists a variable $x \notin \mathbb{M}$ such that $n(x,l) < n(x,r)$, then terminate with failure.

(M3) If $W'(l,r) > 0$ then terminate with success. If $W'(l,r) < 0$ then terminate with failure.

Note that at this point we have $W'(l,r) = 0$.

(M4) If (l,r) has the form $(g(s_1,\ldots,s_n), h(t_1,\ldots,t_p))$ where g and h are distinct, then do the following. If $g \gg h$ terminate with success, otherwise terminate with failure.

(M5) If (l,r) has the form $(g(s_1,\ldots,s_n), g(t_1,\ldots,t_n))$, then replace $\langle l, L\rangle \geq \langle r, R\rangle$ by $\langle s_1,\ldots,s_n, L\rangle \geq \langle t_1,\ldots,t_n, R\rangle$.

(M6) If (l,r) has the form (x,y), where x and y are different variables, do the following. (Note that at this point $x, y \in \mathbb{M}$.) If there exists only one term of the minimal weight, then replace $\langle l, L\rangle \geq \langle r, R\rangle$ by $\langle L\rangle \geq \langle R\rangle$. Otherwise terminate with failure.

(M7) If (l,r) has the form (x,t), where t is not a variable, do the following. If t is different from e, then terminate with failure. Otherwise, replace all occurrences of x in L and R by e obtaining L' and R'. Replace $\langle l, L\rangle \geq \langle r, R\rangle$ by $\langle L'\rangle \geq \langle R'\rangle$.

(M8) If (l,r) has the form (t,x), where t is not a variable, do the following. If t contains x then terminate with success. Otherwise, if t is not the greatest term among the terms of the minimal weight, then terminate with failure. Otherwise, replace all occurrences of x in L and R by t obtaining L' and R', and replace $\langle l, L\rangle \geq \langle r, R\rangle$ by $\langle L'\rangle \geq \langle R'\rangle$. Note that this step does not increase the size of the tuple inequality since t must be a constant, when we substitute it for x.

After this step repeat PREPROCESS.

The proof of correctness of each step is almost the same as the proof of correctness for the corresponding steps in the orientability algorithm, so we leave it to the reader. It is obvious that the algorithm terminates in polynomial time, since every step of the algorithm can be done in polynomial time and after every step the size of $\mathbb{R}$ decreases.

10 Main results

Lemmas 6.1–6.19 guarantee that the orientability algorithm is correct. Lemma 6.20 implies that it runs in polynomial time. Hence we obtain the following theorem.

THEOREM 10.1 *The problem of the existence of an instance of KBO which orients a given rewrite rule systems can be solved in polynomial time.* □

From the reductions of Section 8 we also obtain the following.

THEOREM 10.2 *The orientability problem for the KBO is P-complete. Moreover, it is P-hard even for ground rewrite systems.* □

Similarly, in Section 9 we proved the following theorem.

THEOREM 10.3 *The problem of solving a given Knuth-Bendix ordering constraint consisting of a single inequality can be solved in polynomial time.* □

The *real-valued Knuth-Bendix order* is in the same way as above, except that the range of the weight function is the set of non-negative real numbers. The real-valued KBO was introduced in [Martin 1987]. Note that in view of the results of Section 3 on systems of homogeneous linear inequalities (Lemmas 3.4 and 3.5) the algorithm is also sound and complete for the real-valued orders. Therefore, we have

THEOREM 10.4 *If a rewrite rule system is orientable using the real-valued KBO, then it is also orientable using the integer-valued KBO.* □

It follows from this theorem that all our results formulated for the integer-valued KBO also hold for the real-valued KBO.

References

BAADER F. AND NIPKOW T. [1998], *Term Rewriting and All That*, Cambridge University press, Cambridge.

COMON H. [1990], 'Solving symbolic ordering constraints', *International Journal of Foundations of Computer Science* **1**(4), 387–411.

COMON H. AND TREINEN R. [1994], Ordering constraints on trees, *in* S. Tison, ed., 'Trees in Algebra and Programming: CAAP'94', Vol. 787 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 1–14.

CORMEN T., LEISERSON C. AND RIVEST R. [1991], *Introduction to Algorithms*, The MIT Press.

DERSHOWITZ N. AND PLAISTED D. [2001], Rewriting, *in* A. Robinson and A. Voronkov, eds, 'Handbook of Automated Reasoning', Vol. I, Elsevier Science, chapter 9, pp. 533–608.

DETLEFS D. AND FORGAARD R. [1985], A procedure for automatically proving the termination of a set of rewrite rules, *in* J.-P. Jouannaud, ed., 'Rewriting Techniques and Applications, First International Conference, RTA-85', Vol. 202 of *Lecture Notes in Computer Science*, Springer Verlag, Dijon, France, pp. 255–270.

DICK J., KALMUS J. AND MARTIN U. [1990], 'Automating the Knuth-Bendix ordering', *Acta Informatica* **28**(2), 95–119.

JOUANNAUD J.-P. AND OKADA M. [1991], Satisfiability of systems of ordinal notations with the subterm property is decidable, *in* J. Albert, B. Monien and M. Rodríguez-Artalejo, eds, 'Automata, Languages and Programming, 18th International Colloquium, ICALP'91', Vol. 510 of *Lecture Notes in Computer Science*, Springer Verlag, Madrid, Spain, pp. 455–468.

KNUTH D. AND BENDIX P. [1970], Simple word problems in universal algebras, *in* J. Leech, ed., 'Computational Problems in Abstract Algebra', Pergamon Press, Oxford, pp. 263–297.

KOROVIN K. AND VORONKOV A. [2000*a*], A decision procedure for the existential theory of term algebras with the Knuth-Bendix ordering, *in* 'Proc. 15th Annual IEEE Symp. on Logic in Computer Science', Santa Barbara, California, pp. 291–302.

KOROVIN K. AND VORONKOV A. [2000*b*], Knuth-Bendix constraint solving is NP-complete, Preprint CSPP-8, Department of Computer Science, University of Manchester.
URL: *http://www.cs.man.ac.uk/preprints/index.html*

KOROVIN K. AND VORONKOV A. [2001], Knuth-bendix ordering constraint solving is NP-complete, *in* F. Orejas, P. Spirakis and J. van Leeuwen, eds, 'Automata, Languages and Programming, 28th International Colloquium, ICALP 2001', Vol. 2076 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 979–992.

KRISHNAMOORTHY M. AND NARENDRAN P. [1985], 'On recursive path ordering', *Theoretical Computer Science* **40**, 323–328.

LESCANNE P. [1984], Term rewriting systems and algebra, *in* R. Shostak, ed., '7th International Conference on Automated Deduction, CADE-7', Vol. 170 of *Lecture Notes in Computer Science*, pp. 166–174.

MARTIN U. [1987], How to choose weights in the Knuth-Bendix ordering, *in* 'Rewriting Techniques and Applications', Vol. 256 of *Lecture Notes in Computer Science*, pp. 42–53.

NARENDRAN P., RUSINOWITCH M. AND VERMA R. [1999], RPO constraint solving is in NP, *in* G. Gottlob, E. Grandjean and K. Seyr, eds, 'Computer Science Logic, 12th International Workshop, CSL'98', Vol. 1584 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 385–398.

NIEUWENHUIS R. [1993], 'Simple LPO constraint solving methods', *Information Processing Letters* **47**, 65–69.

PAPADIMITRIOU C. [1994], *Computational Complexity*, Addison-Wesley.

SCHRIJVER A. [1998], *Theory of Linear and Integer Programming*, John Wiley and Sons.

Index