The Decidability of the First-order Theory of Knuth-Bendix Order

Ting Zhang, Henny B. Sipma, Zohar Manna*

Computer Science Department
Stanford University
Stanford, CA 94305-9045
{tingz,sipma,zm}@theory.stanford.edu

Abstract. Two kinds of orderings are widely used in term rewriting and theorem proving, namely *recursive path ordering* (RPO) and *Knuth-Bendix ordering* (KBO). They provide powerful tools to prove the termination of rewriting systems. They are also applied in ordered resolution to prune the search space without compromising refutational completeness. Solving ordering constraints is therefore essential to the successful application of ordered rewriting and ordered resolution. Besides the needs for decision procedures for quantifier-free theories, situations arise in constrained deduction where the truth value of quantified formulas must be decided. Unfortunately, the full first-order theory of recursive path orderings is undecidable. This leaves an open question whether the first-order theory of KBO is decidable. In this paper, we give a positive answer to this question using quantifier elimination. In fact, we shall show the decidability of a theory that is more expressive than the theory of KBO.

1 Introduction

Two kinds of orderings are widely used in term rewriting and theorem proving. One is *recursive path ordering* (RPO) which is based on syntactic precedence [9]. The other is *Knuth-Bendix ordering* (KBO) which is of hybrid nature; it relies on numerical values assigned to symbols as well as syntactic precedence [13]. In ordered term rewriting, a strategy built on ordering constraints can dynamically orient an equation, at the time of instantiation, even if the equation is not uniformly orientable. This provides a powerful tool to prove the termination of rewriting systems [6]. In ordered resolution and paramodulation, ordering constraints are used to select maximal literals to perform resolution. It also serves as enabling conditions for inference rules and such conditions can be inherited from previous inferences at each deduction step. This helps to prune redundancy of the search space without compromising refutational completeness [25].

Solving ordering constraints is therefore essential to the successful application of ordered rewriting and ordered resolution. The decision procedures for

* This research was supported in part by NSF grants CCR-01-21403, CCR-02-20134, CCR-02-09237, CNS-0411363, and CCF-0430102, by ARO grant DAAD19-01-1-0723, and by NAVY/ONR contract N00014-03-1-0939.

quantifier-free constraints of both types of orderings have been well-studied [3, 12, 23, 22, 24, 14, 15]. However, situations arise where we need to decide the truth values of quantified formulas on those orderings, especially in the $\exists^*\forall^*$ fragment. Examples include checking the soundness of simplification rules in constrained deduction [7]. Unfortunately, the full first-order theory of recursive path orderings is undecidable [28, 7] except for the special case where the language only has unary functions and the precedence order is total [21]. Until now it has been an open question whether the first-order theory of Knuth-Bendix order is decidable (RTA open problem ♯99). Here we answer this question affirmatively by showing that an extended theory of term algebras with Knuth-Bendix order admits quantifier elimination.

The basic framework is the combination of term algebras with Presburger arithmetic. The extended language has two sorts; the integer sort $\mathbb{Z}$ and the term sort TA. Intuitively, the language is the set-theoretic union of the language of term algebras and the language of Presburger arithmetic. Formulas are formed from *term literals* and *integer literals* using logical connectives and quantifications. The combination is tightly coupled in the following sense. We have a *weight function* mapping terms to integers as well as various *boundary functions* mapping integers to terms. In addition, the Knuth-Bendix order is expanded in two directions. First, the order is decomposed into three disjoint suborders depending on which of three conditions is used in the definition. Secondly, all orders (including the suborders) are extended to gap orders, which assert the least number of distinct objects between two terms. Moreover, as Knuth-Bendix order is recursively defined on a lexicographic extension of itself, gap orders are extended to tuples of terms. Thus we actually establish the decidability of a richer theory.

Related Work and Comparison. Presburger arithmetic (PA) was first shown to be decidable in 1929 by the quantifier elimination method [10]. Efficient algorithms were later discovered by Cooper [8] and further improved in [26].

The decidability of the first-order theory of term algebras was first shown by Mal'cev using quantifier elimination [20]. This result was proved again later in different settings [19, 5, 11, 4, 2, 27, 17, 18, 29, 30].

Quantifier elimination has been used to obtain decidability results for various extensions of term algebras. [19] shows the decidability of the theory of infinite and rational trees. [4] presents an elimination procedure for term algebras with membership predicate in the regular tree language. [2] presents an elimination procedure for structures of feature trees with arity constraints. [27] shows the decidability of term algebras with queues. [18] shows the decidability of term powers, which are term algebras augmented with coordinatewise-defined predicates. [29] extends the quantifier elimination procedure in [11] for term algebras with constant weight function.

The decidability of the theory of RPO has been well-studied. [3] proves the decidability of the quantifier-free theory of total lexicographic path ordering (LPO, a variant of RPO). A similar result holds for RPO [12]. [23] (resp. [22]) establishes the NP-completeness for the quantifier-free theory of LPO (resp.

RPO). A more efficient algorithm for the quantifier-free theory of RPO is given in [24]. [28, 7] show the undecidability of the first-order theory of LPO and the undecidability of the first-order theory of RPO in case of partial precedence. The decidability of the first-order theory of RPO (LPO) in case of unary signature and total precedence is due to [21]. The decidability of the first-order theory of RPO in case of total precedence remains open.

Recently some partial decidability results for the theory of KBO have been obtained. [14] shows the decidability of the quantifier-free theory of term algebras with KBO. [15] improves the algorithm and shows that the quantifier-free theory of KBO is NP-complete. Analogous to [21], [16] shows the decidability of the first-order theory of KBO in the case where all functions are unary.

In this paper, we show the general decidability result for an extended theory of KBO with arbitrary function symbols and weight functions. The method combines the extraction of integer constraints from term constraints with a reduction of quantifiers on term variables to quantifiers on integer variables.

Paper Organization. Section 2 defines term algebras. Section 3 introduces the theory of term algebras with Knuth-Bendix ordering and presents the technical machinery for eliminating quantifiers. Section 4 presents the main contribution of this paper: it expands the elimination procedure in [29] for the extended theory of KBO and proves its correctness. Section 5 briefly explains how to adapt the elimination procedure to the special case where the language contains a unary function of weight 0. Section 6 concludes with some ideas for future work. Due to space limitation all proofs have been omitted from this paper. An extended version of this paper, which includes a detailed description of notation and terminology, and all proofs, is available from the first author's webpage.

2 Term Algebras

We present a general language and structure of term algebras. *In this paper we assume that the signature of our language is finite.* For notation convenience, we do not distinguish syntactic terms in the language from semantic terms in the corresponding structure. The meaning should be clear from the context.

Definition 1. *A term algebra* $\mathfrak{A}_{\mathsf{TA}} : \langle \mathsf{TA}; \mathcal{C}, \mathcal{A}, \mathcal{S}, \mathcal{T} \rangle$ *consists of*

1. TA*: The term domain, which exclusively consists of terms recursively built up from constants by applying constructors. The type of a term* t*, denoted by* $\mathsf{type}(t)$*, is the outmost constructor of* t*. We say that* t *is* α*-typed (or is an* α*-term) if* $\alpha = \mathsf{type}(t)$*.*
2. $\mathcal{C}$*: A finite set of constructors:* $\alpha, \beta, \gamma, \ldots$ *The arity of* α *is denoted by* $\mathsf{ar}(\alpha)$*.*
3. $\mathcal{A}$*: A finite set of constants:* $a, b, c, \ldots$ *We require* $\mathcal{A} \neq \emptyset$ *and* $\mathcal{A} \subseteq \mathcal{C}$*. For* $a \in \mathcal{A}$*,* $\mathsf{ar}(a) = 0$ *and* $\mathsf{type}(a) = a$*.*
4. $\mathcal{S}$*: A finite set of selectors. For a constructor* α *with arity* $k > 0$*, there are* k *selectors* $\mathsf{s}^{\alpha}_1, \ldots, \mathsf{s}^{\alpha}_k$ *in* $\mathcal{S}$*. We call* s^{α}_i $(1 \leq i \leq k)$ *the* i^{th} α*-selector. For a term* x*,* $\mathsf{s}^{\alpha}_i(x)$ *returns the* i^{th} *component of* x *if* x *is an* α*-term and* x *itself otherwise.*
5. $\mathcal{T}$*: A finite set of testers. For each constructor* α *there is a corresponding tester* Is_{α}*. For a term* x*,* $\mathsf{Is}_{\alpha}(x)$ *is true if and only if* x *is an* α*-term. Note that for a constant* a*,* $\mathsf{Is}_a(x)$ *is just* $x = a$*. In addition there is a special tester* Is_{A} *such that* $\mathsf{Is}_{\mathsf{A}}(x)$ *is true if and only if* x *is a constant.*

We use $\mathscr{L}_{\mathsf{TA}}$ to denote the language of term algebras.

Proposition 1 (Axiomatization of Term Algebras). *Let $\bar{z}_\alpha$ abbreviate $z_1, \ldots, z_{\mathsf{ar}(\alpha)}$. The following formula schemes, in which variables are implicitly universally quantified over TA, axiomatize $\mathrm{Th}(\mathfrak{A}_{\mathsf{TA}})$.*

A1. $t(x) \neq x$, *if t is built solely by constructors and t properly contains x.*
A2. $\alpha(x_1 \ldots, x_{\mathsf{ar}(\alpha)}) \neq \beta(y_1, \ldots, y_{\mathsf{ar}(\beta)})$, *if $\alpha, \beta \in C$ and $\alpha \not\equiv \beta$.*
A3. $\alpha(x_1, \ldots, x_{\mathsf{ar}(\alpha)}) = \alpha(y_1, \ldots, y_{\mathsf{ar}(\alpha)}) \rightarrow \bigwedge_{1 \leq i \leq \mathsf{ar}(\alpha)} x_i = y_i$.
A4. $\mathsf{Is}_\alpha(x) \leftrightarrow \exists\, \bar{z}_\alpha \alpha(\bar{z}_\alpha) = x$, *if $\alpha \in C \setminus \mathcal{A}$;* $\mathsf{Is}_a(x) \leftrightarrow x = a$, *if $a \in \mathcal{A}$.*
A5. $\mathsf{Is}_{\mathsf{A}}(x) \leftrightarrow \bigvee_{a \in \mathcal{A}} \mathsf{Is}_a(x)$.
A6. $\mathsf{s}_i^\alpha(x) = y \leftrightarrow \exists \bar{z}_\alpha \Big(\alpha(\bar{z}_\alpha) = x \wedge y = z_i\Big) \vee \Big(\forall \bar{z}_\alpha(\alpha(\bar{z}_\alpha) \neq x) \wedge x = y\Big)$.

This set of axioms is a variant of the axiomatization given in [11].

Selectors and testers can be defined by constructors and vice versa. One direction has been shown by (A4-A6), which are pure definitional axioms. The other direction follows from the equivalence of $\bigwedge_{i=1}^k \mathsf{s}_i^\alpha(x) = x_i \wedge \mathsf{Is}_\alpha(x)$ and $x = \alpha(x_1, \ldots, x_k)$. For simplicity, from now on we assume $\mathscr{L}_{\mathsf{TA}}$ only has selector functions, and we use $x = \alpha(x_1, \ldots, x_k)$ only in discussions at the semantic level.

We write $\alpha = (\mathsf{s}_1^\alpha, \ldots, \mathsf{s}_k^\alpha)$ $(k > 0)$ to mean that α is a constructor of arity k, and $\mathsf{s}_1^\alpha, \ldots, \mathsf{s}_k^\alpha$ are the corresponding selectors of α. We use L to denote selector sequences. If $L = \mathsf{s}_1, \ldots, \mathsf{s}_n$, Lx stands for $\mathsf{s}_1(\ldots(\mathsf{s}_n(x)\ldots))$, and we say that the *depth* of x in Lx is n. The depth of x in a formula φ is the maximum depth of x in the selector terms in φ, denoted by $\mathsf{depth}_\varphi(x)$.

3 Term Algebras with Knuth-Bendix Order

In this section we introduce the theory of term algebras with KBO and present the technical machinery needed in the quantifier elimination procedure.

Let Σ be a finite signature in the constructor language (i.e., $\Sigma = C$ in Def. 1) and $\mathsf{W} : \Sigma \rightarrow \mathbb{N}$ a weight function. We expand $\mathsf{dom}(\mathsf{W})$ to TA by recursively defining $\mathsf{W}(\alpha(t_1, \ldots, t_k)) = \mathsf{W}(\alpha) + \sum_{i=1}^k \mathsf{W}(t_i)$. Let $\prec^\Sigma$ be a linear precedence order on symbols in Σ. We enumerate all symbols in the decreasing $\prec^\Sigma$-order such that $\alpha_1 \succ^\Sigma \alpha_2 \succ^\Sigma \ldots \succ^\Sigma \alpha_{|\Sigma|}$.

Definition 2 (Knuth-Bendix Order [13]). *A Knuth-Bendix order (KBO) $\prec^{\mathsf{kb}}$ (parameterized with a weight function W and a precedence order $\prec^\Sigma$) is defined recursively such that for $u, v \in \mathsf{TA}$, $u \prec^{\mathsf{kb}} v$ if and only if one of the following conditions holds: (i) $\mathsf{W}(u) < \mathsf{W}(v)$, (ii) $\mathsf{W}(u) = \mathsf{W}(v)$ and $\mathsf{type}(u) \prec^\Sigma \mathsf{type}(v)$, (iii) $\mathsf{W}(u) = \mathsf{W}(v)$, $u \equiv \alpha(u_1, \ldots, u_k)$, $v \equiv \alpha(v_1, \ldots, v_k)$ and*

$$(\exists i) \Big[1 \leq i \leq k \wedge u_i \prec^{\mathsf{kb}} v_i \wedge \forall j (1 \leq j < i \rightarrow u_j = v_j) \Big]. \tag{1}$$

The KBO $\prec^{\mathsf{kb}}$ is a well-founded total order on TA [13, 1]. To guarantee well-foundedness, two *compatibility conditions* for W and $\prec^\Sigma$ are required: (i) $\mathsf{W}(a) > 0$ for any constant a, and (ii) a unary function of weight 0, if present, should be the maximum in $\prec^\Sigma$. Let us denote by $\perp$ the smallest term with respect to $\prec^{\mathsf{kb}}$.

It follows from (i) and (ii) that $\perp$ must be an atom and so it can be determined when W and $\prec^{\Sigma}$ are given. By (ii) if a unary function of weight 0 exists, it must be unique. For presentation simplicity, we assume that $\mathsf{W}(\alpha_1) > 0$. However, the existence of such function actually simplifies our decision procedure. We defer the discussion to Sec. 5.

Definition 3. *The structure of term algebras with KBO is* $\mathfrak{A}_{\mathsf{kb}} = \langle \mathfrak{A}_{\mathsf{TA}}; \prec^{\mathsf{kb}} \rangle$. *Let* $\mathscr{L}_{\mathsf{kb}}$ *denote the language of* $\mathfrak{A}_{\mathsf{kb}}$.

3.1 Proof Plan

We shall show the decidability of $\mathrm{Th}(\mathfrak{A}_{\mathsf{kb}})$ by quantifier elimination. The procedure relies on the following two ideas: *solved form* and *depth reduction*.

1. *Solved Form.* A quantifier-free formula $\varphi(x, \bar{y})$ is *solved* in x if it is in the form
$$\bigwedge_{i \leq m} u_i \prec^{\mathsf{kb}} x \ \wedge \ \bigwedge_{j \leq n} x \prec^{\mathsf{kb}} v_j \ \wedge \ \varphi'(\bar{y}), \tag{2}$$
where x does not appear in u_i, v_i and φ'. It is not hard to argue that $(\exists x)\, \varphi(x, \bar{y})$ simplifies to
$$\bigwedge_{i \leq m, j \leq n} u_i \prec_2^{\mathsf{kb}} v_j \ \wedge \ \varphi'(\bar{y}) \tag{3}$$
where $\prec_n^{\mathsf{kb}}$, called *gap order*, is an extension of $\prec^{\mathsf{kb}}$ such that $x \prec_n^{\mathsf{kb}} y$ states there is an increasing chain from x to y with at least $n-1$ elements in between [10, page 196]. It is clear that the elimination of $\exists x$, the transformation from (2) to (3), becomes straightforward once the matrix $\varphi(x, \bar{y})$ is solved in x, or equivalently, $\mathsf{depth}_{\varphi}(x) = 0$. That leads us to the notion of *depth reduction*.
2. *Depth Reduction.* Let us first consider the simple case where x is α-typed for a proper constructor α and all occurrences of x have depth greater than 0. By introducing new variables $x_1, \ldots, x_{\mathsf{ar}(\alpha)}$ (called the descendants of x) to represent x, we can rewrite $\exists x \varphi(x, \bar{y})$ to
$$\exists x_1, \ldots, \exists x_{\mathsf{ar}(\alpha)} \varphi'(x_1, \ldots, x_{\mathsf{ar}(\alpha)}, \bar{y}), \tag{4}$$
where $\varphi'(x_1, \ldots, x_{\mathsf{ar}(\alpha)}, \bar{y})$ is obtained from $\varphi(x, \bar{y})$ by substituting x_i for $\mathsf{s}_i^{\alpha} x$ $(1 \leq i \leq \mathsf{ar}(\alpha))$. It is clear that $\mathsf{depth}_{\varphi'}(x_i) < \mathsf{depth}_{\varphi}(x)$. If all occurrences of x have the same depth, then by repeating the process we can generate a formula solved in $\bar{x}^*$ where $\bar{x}^*$ are descendants of x. A difficulty arises when not all occurrences of x have equal depth. So eventually we meet the situation where some occurrences of x have depth 0 and some do not. Here we have to represent all occurrences of x of depth 0 in terms of $\mathsf{s}_1^{\alpha}(x), \ldots, \mathsf{s}_{\mathsf{ar}(\alpha)}^{\alpha}(x)$. This amounts to reducing literals of the form $x \prec_n^{\mathsf{kb}} t$ and literals of the form $t \prec_n^{\mathsf{kb}} x$ to quantifier-free formulas using $\mathsf{s}_1^{\alpha}(x), \ldots, \mathsf{s}_{\mathsf{ar}(\alpha)}^{\alpha}(x)$. After that we can introduce new variables and do quantifier manipulation just as in the simple case to bring $\exists x \varphi(x, \bar{y})$ into the form of (4). Therefore depth reduction essentially depends on the reduction of $x \prec_n^{\mathsf{kb}} t$ and reduction of $t \prec_n^{\mathsf{kb}} x$. In order to carry out the reduction we need to extend the language as follows.

(a) We decompose $\prec^{\mathsf{kb}}$ into three disjoint suborders $\prec^{\mathsf{w}}$, $\prec^{\mathsf{p}}$ and $\prec^{\mathsf{l}}$, each of which is also extended to gap orders.
(b) We introduce Presburger arithmetic explicitly in order to define *counting constraints* to count how many distinct terms there are at certain weight, and define *boundary functions* to *delineate* gap orders.
(c) The reduction of literals like $x \prec_n^{\mathsf{kb}} t$ or $t \prec_n^{\mathsf{kb}} x$ eventually comes down to resolving relations between two terms of the same weight and of the same type. So we need to extend all aforementioned notions to tuples of terms of the same total weight.

In the rest of this section we define these extensions.

3.2 Decomposition of Knuth-Bendix Order

Definition 4. *A Knuth-Bendix order $\prec^{\mathsf{kb}}$ can be decomposed into three* disjoint *orders, a* weight order $\prec^{\mathsf{w}}$*, a* precedence order $\prec^{\mathsf{p}}$*, and a* lexicographical order $\prec^{\mathsf{l}}$*, as follows:*

$$
\begin{aligned}
u \prec^{\mathsf{w}} v &\Leftrightarrow \mathsf{W}(u) < \mathsf{W}(v),\\
u \prec^{\mathsf{p}} v &\Leftrightarrow \mathsf{W}(u) = \mathsf{W}(s) \ \& \ \mathsf{type}(u) \prec^{\Sigma} \mathsf{type}(v),\\
u \prec^{\mathsf{l}} v &\Leftrightarrow \mathsf{W}(u) = \mathsf{W}(v) \ \& \ \mathsf{type}(u) = \mathsf{type}(v) \ \& \ u \prec^{\mathsf{kb}} v,
\end{aligned}
$$

such that $u \prec^{\mathsf{kb}} v$ is equivalent to $u \prec^{\mathsf{w}} v \vee u \prec^{\mathsf{p}} v \vee u \prec^{\mathsf{l}} v$. We write $u \prec^{\mathsf{pl}} v$ as an abbreviation for $u \prec^{\mathsf{p}} v \vee u \prec^{\mathsf{l}} v$.

3.3 Gap Orders

To express formulas of the form $\exists x(u \prec^{\sharp} x \prec^{\sharp} v)$ in a quantifier-free language we need to extend all aforementioned orders to "gap" orders.

Definition 5 (Gap Orders). *Define $\prec_n^{\mathsf{kb}}$ $(n \geq 0)$ such that*

$$u \prec_n^{\mathsf{kb}} v \leftrightarrow (\exists u_1, \ldots, \exists u_n)\Big[u \prec^{\mathsf{kb}} u_1 \prec^{\mathsf{kb}} \ldots \prec^{\mathsf{kb}} u_n \preceq^{\mathsf{kb}} v\Big].$$

For $\sharp \in \{\mathsf{w}, \mathsf{p}, \mathsf{l}, \mathsf{pl}\}$, define $\prec_n^{\sharp}$ such that $u \prec_n^{\sharp} v \leftrightarrow u \prec_n^{\mathsf{kb}} v \wedge u \prec^{\sharp} v$, and $u \leqq_n^{\sharp} v$ such that $(u \prec_n^{\sharp} v) \wedge \neg(u \prec_{n+1}^{\sharp} v)$.

A gap order $u \prec_n^{\sharp} v$ $(n \geq 1)$ states that "u is less than v w.r.t. $\prec^{\sharp}$, and there are *at least* $n-1$ elements in between." Similarly, $u \leqq_n^{\sharp} v$ $(n \geq 1)$ states that "u is less than v w.r.t. $\prec^{\sharp}$, and there are *exactly* $n-1$ elements in between". Note that $\prec_1^{\sharp}$ is just $\prec^{\sharp}$, $\prec_0^{\sharp}$ is $\preceq^{\sharp}$, $\leqq_0^{\sharp}$ is $=$, and we have $u \prec_n^{\sharp} v \leftrightarrow u \prec_{n+1}^{\sharp} v \vee u \leqq_n^{\sharp} v$.

Example 1. The formula $\exists x(u \prec^{\mathsf{l}} x \prec^{\mathsf{l}} v)$ reduces to $u \prec_2^{\mathsf{l}} v$ if u, v do not contain x.

3.4 Boundary Functions

Consider the formula $u \leqq_1^{\mathsf{w}} v$. Intuitively it states "$\mathsf{W}(u) < \mathsf{W}(v)$ and there are no terms z such that $u \prec^{\mathsf{kb}} z \prec^{\mathsf{kb}} v$, that is, u is the largest term of weight $\mathsf{W}(u)$ and v is the smallest term of weight $\mathsf{W}(v)$". To express this we introduce *boundary functions*.

Definition 6 (Boundary Functions). *Let $n, p > 0$. The following functions are called boundary functions:*

1. $\mathsf{0}^{\mathsf{w}} : \mathbb{N} \to \mathsf{TA}$ *such that $\mathsf{0}^{\mathsf{w}}(n)$ is the smallest term (w.r.t. $\prec^{\mathsf{kb}}$) of weight n,*
2. $\mathsf{0}^{\mathsf{p}} : \mathbb{N}^2 \to \mathsf{TA}$ *such that $\mathsf{0}^{\mathsf{p}}(n, p)$ is the smallest term (w.r.t. $\prec^{\mathsf{kb}}$) of weight n and type α_p,*

where, for all of the above, $f(n) = \perp$ and $f(n, p) = \perp$, if no such term exists.

Similarly we define $\mathsf{1}^{\mathsf{w}} : \mathbb{N} \to \mathsf{TA}$ and $\mathsf{1}^{\mathsf{p}} : \mathbb{N}^2 \to \mathsf{TA}$ as the largest terms with the corresponding properties. We write $\mathsf{0}^{\sharp}_{(\ldots)}$ for $\mathsf{0}^{\sharp}(\ldots)$ and $\mathsf{1}^{\sharp}_{(\ldots)}$ for $\mathsf{1}^{\sharp}(\ldots)$. Terms having one of these functions as root symbol are called *boundary terms*. A literal of the form $u \star v$, where $\star$ is either equality or a gap order, is *open* if both u and v are ordinary terms in TA, *closed* if both u and v are boundary terms, and *half-open* otherwise.

3.5 Integer Extension of Term Algebras

To be able to express the boundary terms in the formal language, we extend term algebras with Presburger arithmetic (PA).

Definition 7. *The structure of term algebras with integers is $\mathfrak{A}^{\mathbb{Z}}_{\mathsf{TA}} = \langle \mathfrak{A}_{\mathsf{TA}}; \mathfrak{A}_{\mathbb{Z}}; (.)^{\mathsf{w}} \rangle$, where $\mathfrak{A}_{\mathbb{Z}}$ is Presburger arithmetic and $(.)^{\mathsf{w}}$ denotes the weight function.*

We call terms of sort TA (resp. $\mathbb{Z}$) TA-*terms* (resp. *integer terms*), similarly for variables and quantifiers. We also use "term" for "TA" when there is no confusion. A TA-term can occur inside the weight function. Such occurrence is called *integer occurrence* to be distinguished from the normal *term occurrence*. From now on, we freely use integer terms t^{w} to form Presburger formulas, and we use $\mathsf{depth}_{\varphi}(x)$ to denote the maximum depth of term occurrences of x in φ.

Example 2. The formula $(\exists x : \mathsf{TA})\Big[\mathsf{0}^{\mathsf{w}}_{(x^{\mathsf{w}})} \prec^{\mathsf{pl}} x \prec^{\mathsf{pl}} \mathsf{1}^{\mathsf{w}}_{(x^{\mathsf{w}})}\Big]$ states that there exists a term $t \in \mathsf{TA}$ such that there are at least three elements with the same weight as t (including t itself). Note that the first and the third occurrences of x are integral while the second one is an ordinary term.

The truth value of the formula in Ex. 2 relies on the number of distinct TA-terms of a certain weight. This is the essential use of Presburger arithmetic.

Definition 8 (Counting Constraint). *A* counting constraint *is a predicate $\mathsf{CNT}^{\alpha}_{n}(z)$ that states there are at least $n+1$ different α-terms of weight z. $\mathsf{CNT}_n(z)$ is similarly defined with α-terms replaced by TA-terms. We write Tree^{α} (resp. Tree) for $\mathsf{CNT}^{\alpha}_{0}$ (resp. CNT_0).*

Counting constraints play a central role in our elimination procedure; it helps reduce term quantifiers to integer quantifiers.

Example 3. The formula from Ex. 2 is reduced to $(\exists z : \mathbb{Z})\, \mathsf{CNT}_2(z)$.

It was proved in [14, 29] that counting constraints can be expressed in PA.

Example 4. Consider $\mathfrak{A}^{\mathbb{Z}}_{\mathsf{list}} = (\mathfrak{A}_{\mathsf{list}}; \mathfrak{A}_{\mathbb{Z}}; (.)^{\mathsf{w}})$ where $\mathfrak{A}_{\mathsf{list}} = \langle \mathsf{list}, \mathsf{cons}, \mathsf{car}, \mathsf{cdr}, a\rangle$ is the LISP list structure with the only atom a, and $(.)^{\mathsf{w}}$ is a constant weight function equal to 1. It has been shown in [30] that $\mathsf{CNT}^{\mathsf{cons}}_n(x)$ is $x \geq 2m-1 \wedge 2 \nmid m$ where m is the least number such that the m-th *Catalan number* $C_m = \frac{1}{m}\binom{2m-2}{m-1}$ is greater than n. This is not surprising as C_m gives the number of binary trees with m leaves (that tree has $2m-1$ nodes).

3.6 Extension of Knuth-Bendix Order

Definition 9. *The structure of term algebras with KBO, extended with gap orders, boundary functions and Presburger arithmetic, is*

$$\mathfrak{A}^{\mathbb{Z}}_{\mathsf{kb}^+} = \langle \mathfrak{A}_{\mathsf{kb}};\ \mathfrak{A}_{\mathbb{Z}};\ \prec^{\sharp}_n, \leqq^{\sharp}_n,\ \sharp \in \{\mathsf{kb}, \mathsf{w}, \mathsf{p}, \mathsf{l}, \mathsf{pl}\}, n \geq 0;\ \mathsf{0}^{*}_{(\ldots)}, \mathsf{1}^{*}_{(\ldots)},\ * \in \{\mathsf{w}, \mathsf{p}\}\rangle.$$

We denote by $\mathscr{L}_{\mathsf{kb}^+}$ the language extending $\mathscr{L}_{\mathsf{kb}}$ with gap orders and boundary terms and by $\mathscr{L}_{\mathbb{Z}}$ the language of Presburger arithmetic (including weight functions on terms). The complete language is denoted by $\mathscr{L}^{\mathbb{Z}}_{\mathsf{kb}^+}$.

3.7 Tuples of Terms

The extensions for tuples of terms are defined as follows:

Definition 10 (Orders on Tuples). *Let $\bar{u} = \langle u_1, \ldots, u_k\rangle, \bar{v} = \langle v_1, \ldots, v_k\rangle$ such that $\Sigma^k_{i=1}\mathsf{W}(u_i) = \Sigma^k_{i=1}\mathsf{W}(v_i)$. The lexicographical extension $\prec^{k;\mathsf{kb}}$ $(k \geq 1)$ of $\prec^{\mathsf{kb}}$ on k-tuples of the same weight is defined such that $\bar{u} \prec^{k;\mathsf{kb}} \bar{v}$ if and only if* (1) *holds.*

Definition 11 (Suborders on Tuples). *Let $\bar{u} = \langle u_1, \ldots, u_k\rangle, \bar{v} = \langle v_1, \ldots, v_k\rangle \in \mathsf{TA}^k$, $\sharp \in \{\mathsf{w}, \mathsf{p}, \mathsf{l}, \mathsf{pl}\}$. We define those composite orders on tuples as follows.*

$$\bar{u} \prec^{k;\sharp} \bar{v} \leftrightarrow u_1 \prec^{\sharp} v_1 \ \vee\ (u_1 = v_1 \ \wedge\ \langle u_2, \ldots, u_k\rangle \prec^{k-1;\mathsf{kb}} \langle v_2, \ldots, v_k\rangle)$$

We say that $\bar{u} \prec^{k;\sharp} \bar{v}$ is proper *if $u_1 \prec^{\sharp} v_1$ and we have $\bar{u} \prec^{k;\mathsf{kb}} \bar{v} \leftrightarrow \bar{u} \prec^{k;\mathsf{w}} \bar{v} \vee \bar{u} \prec^{k;\mathsf{p}} \bar{v} \vee \bar{u} \prec^{k;\mathsf{l}} \bar{v}$.*

Definition 12 (Gap Orders between Tuples). *We define $\prec^{k;\mathsf{kb}}_n$ $(k \geq 1; n \geq 0)$ such that*

$$\bar{u} \prec^{k;\mathsf{kb}}_n \bar{v} \leftrightarrow (\exists \bar{u}_1, \ldots, \exists \bar{u}_n : \mathsf{TA}^k)\left[\bar{u} \prec^{k;\mathsf{kb}} \bar{u}_1 \prec^{k;\mathsf{kb}} \ldots \prec^{k;\mathsf{kb}} \bar{u}_n \preceq^{k;\mathsf{kb}} \bar{v}\right].$$

For $\sharp \in \{\mathsf{w}, \mathsf{p}, \mathsf{l}, \mathsf{pl}\}$, define $\prec^{k;\sharp}_n$ such that $\bar{u} \prec^{k;\sharp}_n \bar{v} \leftrightarrow \bar{u} \prec^{k;\mathsf{kb}}_n \bar{v} \wedge \bar{u} \prec^{k;\sharp} \bar{v}$, and $\bar{u} \leqq^{k;\sharp}_n \bar{v}$ such that $(\bar{u} \prec^{k;\sharp}_n \bar{v}) \wedge \neg(\bar{u} \prec^{k;\sharp}_{n+1} \bar{v})$. Again note that $\prec^{k;\sharp}_1$ is just $\prec^{k;\sharp}$, $\prec^{k;\sharp}_0$ is $\preceq^{k;\sharp}$, $\leqq^{k;\sharp}_0$ is $=$, and $\bar{u} \prec^{k;\sharp}_n \bar{v} \leftrightarrow \bar{u} \prec^{k;\sharp}_{n+1} \bar{v} \vee \bar{u} \leqq^{k;\sharp}_n \bar{v}$.

Definition 13 (Tuple Boundary Functions). *Let $k, n, m, p > 0$. Define partial functions:*

1. *$\bar{\mathbf{0}}^{k;\mathsf{kb}} : \mathbb{N} \to \mathsf{TA}^k$ $(k \geq 1)$ such that $\bar{\mathbf{0}}^{k;\mathsf{kb}}(n)$ is the smallest k-tuple (w.r.t. $\prec^{k;\mathsf{kb}}$) of weight n.*
2. *$\bar{\mathbf{0}}^{k;\mathsf{w}} : \mathbb{N}^2 \to \mathsf{TA}^k$ $(k \geq 1)$ such that $\bar{\mathbf{0}}^{k;\mathsf{w}}(n, m)$ is the smallest k-tuple (w.r.t. $\prec^{k;\mathsf{kb}}$) of weight n and the first component has weight m.*
3. *$\bar{\mathbf{0}}^{k;\mathsf{p}} : \mathbb{N}^3 \to \mathsf{TA}^k$ $(k \geq 1)$ such that $\bar{\mathbf{0}}^{k;\mathsf{p}}(n, m, p)$ is the smallest k-tuple (w.r.t. $\prec^{k;\mathsf{kb}}$) of weight n and the first component has weight m and type α_p.*

Similarly we define $\bar{\mathbf{1}}^{k;\mathsf{kb}} : \mathbb{N} \to \mathsf{TA}^k$, $\bar{\mathbf{1}}^{k;\mathsf{w}} : \mathbb{N}^2 \to \mathsf{TA}^k$ and $\bar{\mathbf{1}}^{k;\mathsf{p}} : \mathbb{N}^3 \to \mathsf{TA}^k$ to be the largest k-tuples with the corresponding properties. As before these functions are made total by assigning $\langle \bot, \ldots \bot \rangle$ to undefined values. We write $\bar{\mathbf{0}}^{k;\sharp}_{(\ldots)}$ for $\bar{\mathbf{0}}^{k;\sharp}(\ldots)$ and $\bar{\mathbf{1}}^{k;\sharp}_{(\ldots)}$ for $\bar{\mathbf{1}}^{k;\sharp}(\ldots)$. Terms having one of these functions as root symbol are called *boundary tuples*. As before we call a literal $\bar{\boldsymbol{u}} \star \bar{\boldsymbol{v}}$ *open* if both $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{v}}$ are ordinary tuples, *closed* if both $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{v}}$ are boundary tuples, and *half-open* otherwise.

To avoid unnecessary complications, we choose to treat tuples (including boundary tuples) as "syntactic sugar"; they are only used in the intermediate steps of the reduction. Lemma 5 shows that literals containing tuples can be reduced to formulas in $\mathscr{L}^{\mathbb{Z}}_{\mathsf{kb}^+}$.

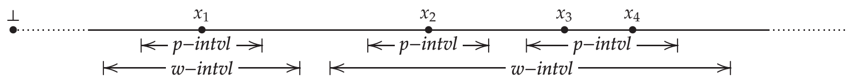
3.8 Delineated Gap Order Completion

Revisiting the transformation from (2) to (3), we see that the number of gap orders in (3) is quadratic in the number of gap orders in (2). This complicates the termination proof for the elimination procedure. Nevertheless, we can avoid this difficulty by postulating the relative positions of parameters. This leads to the notion of *order completion*.

Definition 14 (Gap Order Completion). *A* gap order completion *(GOC) φ' of a conjunction of literals $\varphi(t_1, \ldots, t_n)$ is chain $t_{f(1)} \trianglerighteq \ldots \trianglerighteq t_{f(n)}$, where f is a permutation function on $\{1, \ldots, n\}$ and $\trianglerighteq$ stands for $=$, $\leqq^{\sharp}_n$ or $\prec^{\sharp}_n$ ($\sharp \in \{\mathsf{w}, \mathsf{p}, \mathsf{l}, \mathsf{pl}\}$, $n \geq 1$).*

Example 5. A possible GOC of $\varphi(x, y, z) : x \prec^{\mathsf{w}}_9 y \wedge x \prec^{\mathsf{pl}} z \wedge z \prec^{\mathsf{w}} y$ is $x \prec^{\mathsf{pl}}_5 z \prec^{\mathsf{w}}_4 y$.

However, gap order completions are not sufficient. It is quite clear to see $(\exists x : \mathsf{TA})[u \prec^{\mathsf{w}} x \prec^{\mathsf{p}} v]$ implies $u \prec^{\mathsf{w}}_2 v$. But for the converse to hold, $v \neq \mathsf{0}^{\mathsf{w}}_{(v^{\mathsf{w}})}$ is required. As another example, $(\exists x : \mathsf{TA})[u \prec^{\mathsf{p}} x \prec^{\mathsf{p}} v]$ implies $u \prec^{\mathsf{p}}_2 v$, but not vice versa. In order to preserve equivalence, intuitively, we need to "delineate" a GOC to make sure ordinary terms in different *intervals* (a notion to be define precisely soon) are not related in any gap orders. For example, consider the linear order $x_1 \prec^{\mathsf{w}}_{n_1} x_2 \prec^{\mathsf{p}}_{n_2} x_3 \prec^{\mathsf{l}}_{n_3} x_4$ The order imposed may be viewed as follows



The weight of x_1 is strictly lower than that of x_2, x_3, and x_4. The weight of x_2, x_3, and x_4 is the same, but the precedence of x_2 is lower than that of x_3 and x_4. Finally, x_3 is smaller than x_4 in the lexicographic order. We call a maximal list of elements with the same weight a w-interval, and similarly a maximal list of elements with the same weight and precedence order a p-interval. Thus, the second w-interval above has two inner p-intervals.

We want to avoid relating ordinary elements at different levels in different intervals. Therefore we augment the gap order completion with boundary terms, called a delineated gap order completion.

Definition 15 (Delineated Gap Order Completion). *A delineated gap order completion (DGOC) is a GOC in which if there occurs the following pattern $v_1 \trianglerighteq^{\sharp}_{n_1} u \trianglerighteq^{\natural}_{n_2} v_2$, where $n_1, n_2 > 0$, $\trianglerighteq$ stands for either $\prec$ or $\leqq$, $\sharp, \natural \in \{\mathsf{w}, \mathsf{p}, \mathsf{l}, \mathsf{pl}\}$, and u is an ordinary term in $\mathscr{L}_{\mathsf{kb}}$, then either $\sharp \equiv \natural \equiv \mathsf{pl}$ or $\sharp \equiv \natural \equiv \mathsf{l}$. I.e., ordinary terms do not delineate two intervals unless they are asserted equal to boundary terms.*

Example 6. Revisit Ex. 5. A possible DGOC of $\varphi(x, y, z)$ is

$$\varphi'(x, y, z) : \underbrace{\mathsf{0}^{\mathsf{w}}_{(x^{\mathsf{w}})} \prec^{\mathsf{pl}}_1 x \prec^{\mathsf{pl}}_5 z \prec^{\mathsf{pl}}_2 \mathsf{1}^{\mathsf{w}}_{(x^{\mathsf{w}})}}_{\mathsf{w}\text{-interval}} \prec^{\mathsf{w}}_1 \underbrace{\mathsf{0}^{\mathsf{w}}_{(y^{\mathsf{w}})} \prec^{\mathsf{pl}}_1 y \prec^{\mathsf{pl}}_1 \mathsf{1}^{\mathsf{w}}_{(y^{\mathsf{w}})}}_{\mathsf{w}\text{-interval}}$$

Now we have $(\exists z : \mathsf{TA})\varphi'(x, y, z) \leftrightarrow \mathsf{0}^{\mathsf{w}}_{(x^{\mathsf{w}})} \prec^{\mathsf{pl}}_1 x \prec^{\mathsf{pl}}_7 \mathsf{1}^{\mathsf{w}}_{(x^{\mathsf{w}})} \prec^{\mathsf{w}}_1 \mathsf{0}^{\mathsf{w}}_{(y^{\mathsf{w}})} \prec^{\mathsf{pl}}_1 y \prec^{\mathsf{pl}}_1 \mathsf{1}^{\mathsf{w}}_{(y^{\mathsf{w}})}$.

Lemma 1 (Delineated Gap Order Completion). *Any conjunction of positive literals in $\mathscr{L}_{\mathsf{kb}^+}$ is equivalent to a finite disjunction of delineated gap order completions.*

Now we state a sequence of lemmas which will justify the elimination procedure given in the next section. These lemmas share the following common features: (i) they state the soundness of symbolic transformations for formulas in *primitive form*, a special prenex form where the prefix only consists of existential quantifiers and the matrix is a conjunction of literals; (ii) a formula φ is transformed to a finite disjunction $\bigvee_i \varphi_i$ where for any i, φ_i is in primitive form and the matrix of φ_i contains no more open gap order literals than that of φ does. To save space, we omit these conditions in the description of each lemma.

In principle, boundary terms can appear in the weight function or in selectors, selector terms can occur in the weight function, and the weight function can be used to construct boundary terms. Repeating this process we can build more and more complex terms. The following lemma eliminates this superficial complication. From now on, we assume that boundary terms are not properly embedded in other terms.

Lemma 2 (Depth Reduction of Boundary Terms). *Any formula in $\mathscr{L}^{\mathsf{Z}}_{\mathsf{kb}^+}$ can be effectively reduced to an equivalent formula in which no boundary terms appear inside selectors or the weight function.*

The following lemma states that we can always assume that all *term occurrences* of a TA-variable have the same depth, and hence we are able to reduce them all together to depth 0.

Lemma 3 (Depth Reduction). *Let $\star \in \{\prec^{\mathsf{kb}}_n, \prec^{\mathsf{w}}_n, \prec^{\mathsf{p}}_n, \prec^{\mathsf{l}}_n, \prec^{\mathsf{pl}}_n, \leqq^{\mathsf{kb}}_n, \leqq^{\mathsf{w}}_n, \leqq^{\mathsf{p}}_n, \leqq^{\mathsf{l}}_n, \leqq^{\mathsf{pl}}_n\}$. If x is of type α_p with $\alpha_p = (\mathsf{s}^{\alpha_p}_1, \ldots, \mathsf{s}^{\alpha_p}_k)$ and t is an arbitrary term, then $x \star t$ ($t \star x$) can be effectively reduced to an equivalent quantifier-free formula $\varphi(\mathsf{s}^{\alpha_p}_1 x, \ldots, \mathsf{s}^{\alpha_p}_k x)$ (in $\mathscr{L}^{\mathsf{Z}}_{\mathsf{kb}^+}$) in which x does not appear and $\mathsf{s}^{\alpha_p}_i x$ ($1 \leq i \leq k$) is not inside selectors.*

As we mentioned before, this is the main battlefield of quantifier elimination. To streamline the proof, we introduce the following two lemmas.

Lemma 4 (Term Reduction). *Let $\star \in \{\prec^{\mathsf{kb}}_n, \prec^{\mathsf{w}}_n, \prec^{\mathsf{p}}_n, \prec^{\mathsf{l}}_n, \prec^{\mathsf{pl}}_n, \leqq^{\mathsf{kb}}_n, \leqq^{\mathsf{w}}_n, \leqq^{\mathsf{p}}_n, \leqq^{\mathsf{l}}_n, \leqq^{\mathsf{pl}}_n\}$.*

1. *If x is an ordinary term of type α_p with $\alpha_p = (\mathsf{s}_1^{\alpha_p}, \dots, \mathsf{s}_k^{\alpha_p})$ and t is either a boundary term or an ordinary term not containing x, then $x \star t$ ($t \star x$) can be effectively reduced to an equivalent quantifier-free formula $\varphi(\mathsf{s}_1^{\alpha_p} x, \dots, \mathsf{s}_k^{\alpha_p} x)$ in which x does not occur and $\mathsf{s}_i^{\alpha_p} x$ ($1 \le i \le k$) is not inside selectors.*
2. *If $x \star t$ ($t \star x$) is closed, i.e., both t and x are boundary terms, then it can be effectively reduced to an equivalent Presburger formula.*

Lemma 4 states that literals containing non-atom terms can be expressed only using the components of those terms. The reduction eventually comes down to the success of decomposing relations between tuples of the same weight, as is stated by the following lemma.

Lemma 5 (Tuple Reduction). *Let $\star \in \{\prec_n^{k;\mathsf{kb}}, \prec_n^{k;\mathsf{w}}, \prec_n^{k;\mathsf{p}}, \prec_n^{k;\mathsf{l}}, \prec_n^{k;\mathsf{pl}}, \preceq_n^{k;\mathsf{kb}}, \preceq_n^{k;\mathsf{w}}, \preceq_n^{k;\mathsf{p}}, \preceq_n^{k;\mathsf{l}}, \preceq_n^{k;\mathsf{pl}}\}$, and U, V be k-tuples of the same weight.*

1. *If $U = \langle u_1, \dots, u_k \rangle$ is an ordinary tuple, then $U \star V$ ($V \star U$) can be effectively reduced to an equivalent quantifier-free formula $\varphi(u_1, \dots, u_k)$ (in $\mathscr{L}_{\mathsf{kb}^+}^{\mathbb{Z}}$) in which u_i ($1 \le i \le k$) does not occur inside selectors.*
2. *If $U \star V$ ($V \star U$) is a closed tuple, i.e., both U and V are boundary tuples, then it can be effectively reduced to an equivalent Presburger formula.*

Lemma 6 (Elimination of Term Variables). *Let x be a term variable, $\varphi_{\mathsf{kb}^+}(x)$ a conjunction of literals in $\mathscr{L}_{\mathsf{kb}^+}$ with $\mathsf{depth}_{\varphi_{\mathsf{kb}^+}}(x) = 0$, and $\varphi_{\mathbb{Z}}(x)$ a Presburger formula in which x occurs inside the weight function. Then $(\exists x : \mathsf{TA})[\varphi_{\mathsf{kb}^+}(x) \wedge \varphi_{\mathbb{Z}}(x)]$ can be effectively reduced to $\varphi'_{\mathsf{kb}^+} \wedge \varphi'_{\mathbb{Z}}$ in which x does not occur and φ'_{kb^+} is quantifier-free.*

Lemma 6 states that we can remove term quantifiers by reducing them to integer quantifiers. The next lemma guarantees the elimination of integer quantifiers.

Lemma 7 (Elimination of Integer Variables). *Let z be an integer variable, $\varphi_{\mathsf{kb}^+}(z)$ a conjunction of literals in $\mathscr{L}_{\mathsf{kb}^+}$ where z occurs inside boundary terms, and $\varphi_{\mathbb{Z}}(z)$ a Presburger formula. Then $(\exists z : \mathbb{Z})[\varphi_{\mathsf{kb}^+}(z) \wedge \varphi_{\mathbb{Z}}(z)]$ can be effectively reduced to $\varphi'_{\mathsf{kb}^+} \wedge \varphi'_{\mathbb{Z}}$ where no z occurs and φ'_{kb^+} is quantifier-free.*

4 Quantifier Elimination for $\mathrm{Th}(\mathfrak{A}_{\mathsf{kb}^+}^{\mathbb{Z}})$

In this section we extend the quantifier elimination procedure for $\mathrm{Th}(\mathfrak{A}_{\mathsf{TA}}^{\mathbb{Z}})$ [29] to an elimination procedure for $\mathrm{Th}(\mathfrak{A}_{\mathsf{kb}^+}^{\mathbb{Z}})$. First we introduce some notations to simplify the algorithm description.

4.1 Primitive Form

It is well-known that eliminating arbitrary quantifiers reduces to eliminating existential quantifiers from *primitive formulas* of the form

$$(\exists x)\, \varphi(x, \bar{y}) \equiv (\exists x) \Big[\, A_i(x, \bar{y}) \ \wedge\ \cdots\ \wedge\ A_n(x, \bar{y}) \,\Big], \tag{5}$$

where $A_i(x, \bar{y})$ are ($1 \le i \le n$) literals [11]. We also assume that $A_i(x, \bar{y})$ are not of the form $x = t$ in case t does not contain x, as $(\exists x)[x = t \wedge \varphi'(x, \bar{y})]$ simplifies to $\varphi'(t, \bar{y})$. In addition we can assume A_i are positive literals. The details of elimination of negation are given in the extended version of this paper.

4.2 Nondeterminism

All transformations are carried out on formulas of the form (5). Each step of the transformations manipulates (5) to produce a version of the same form (or multiple versions of the same form in case disjunctions are introduced), and thus in each step $(\exists x)\varphi(x, \bar{y})$ refers to the updated version rather than to the original input formula. Whenever we say "guess ψ", we mean to add a finite disjunction $\bigvee_i \varphi_i$, which is valid in the context and contains ψ as a disjunct, to $\varphi(\bar{x}, \bar{y})$. It should be understood that an implicit disjunctive splitting is carried out and we work on each resultant "simultaneously".

4.3 Type Completion

We say a selector term $\mathsf{s}_i^\alpha(t)$ is *proper* if $\mathsf{Is}_\alpha(t)$ holds. We can make selector terms proper with type information.

Definition 16 (Type Completion). *φ' is a type completion of φ if φ' is obtained from φ by conjoining tester predicates such that for any term t in φ, exactly one type of tester predicate $\mathsf{Is}_\alpha(t)$ ($\alpha \in C$) is in φ'.*

Example 7. Let $\alpha, \beta \in C$, $\alpha \not\equiv \beta$ and $\alpha = (\mathsf{s}_1^\alpha)$. A possible type completion for $y = \mathsf{s}_1^\alpha(x)$ is $y = \mathsf{s}_1^\alpha(x) \wedge \mathsf{Is}_\beta(x) \wedge \mathsf{Is}_\beta(\mathsf{s}_1^\alpha(x)) \wedge \mathsf{Is}_\beta(y)$, which simplifies to $y = x \wedge \mathsf{Is}_\beta(x) \wedge \mathsf{Is}_\beta(y)$ by Axioms (A4) and (A6). Another type completion is $y = \mathsf{s}_1^\alpha(x) \wedge \mathsf{Is}_\alpha(x) \wedge \mathsf{Is}_\beta(\mathsf{s}_1^\alpha(x)) \wedge \mathsf{Is}_\beta(y)$ in which the selector term is proper. As the third example, a type completion could be $y = \mathsf{s}_1^\alpha(x) \wedge \mathsf{Is}_\alpha(x) \wedge \mathsf{Is}_\alpha(\mathsf{s}_1^\alpha(x)) \wedge \mathsf{Is}_\beta(y)$ which simplifies to false.

We assume that all formulas are type-complete. In particular, all selector terms are (simplified to) proper ones. The reason behind this assumption is that a symbolic transformation can always be carried to replace a non-type-complete formula φ by an equivalent finite disjunction of type completions of φ. In terms of efficiency, however, one would prefer doing the on-the-fly disjunctive splitting when the type information of a specific term is needed. We also assume that every type completion is sound with respect to types. Certain type completion of φ may be contradictory due to type conflicts. For example, $\mathsf{Is}_{\mathsf{A}}(x) \wedge \mathsf{Is}_\alpha(\mathsf{s}(x))$ ($\alpha \in C \setminus \mathcal{A}$) is unsatisfiable. Nevertheless, unsatisfiable disjuncts will not affect soundness of the transformation and they can be easily detected and removed. At last, note that we omit listing tester literals unless they are needed for correctness proof.

4.4 Elimination Procedure

The elimination procedure consists of the following two algorithms:

Algorithm 1 (Elimination of Integer Variables).
We assume that formulas with quantifiers on integer variables are in the form

$$(\exists \bar{z} : \mathbb{Z}) \left[\varphi_{\mathbb{Z}}(\bar{x}, \bar{y}, \bar{z}) \ \wedge\ \varphi_{\mathsf{kb}^+}(\bar{x}, \bar{y}, \bar{z})\right], \tag{6}$$

where $\bar{y}$, $\bar{z}$ are integer variables, $\bar{x}$ are term variables. Note that $\bar{x}$ may occur inside the weight function in $\varphi_{\mathbb{Z}}(\bar{x}, \bar{y}, \bar{z})$ and $\bar{y}, \bar{z}$ may appear inside boundary terms in $\varphi_{\mathsf{kb}^+}(\bar{x}, \bar{y}, \bar{z})$.

Repeatedly apply the following subprocedures (A') and (B') to (6) *until $\bar{z} = \emptyset$.*

1. *If none of $\bar{z}$ appears inside any boundary terms, then $\varphi_{\mathsf{kb}^+}(\bar{x},\bar{y},\bar{z})$ is just $\varphi_{\mathsf{kb}^+}(\bar{x},\bar{y})$, which can be moved out of $\exists\bar{z}$. We then obtain*

$$(\exists \bar{z} : \mathbb{Z}) \left[\varphi_{\mathbb{Z}}(\bar{x},\bar{y},\bar{z})\right] \ \wedge\ \varphi_{\mathsf{kb}^+}(\bar{x},\bar{y}).$$

   Since $(\exists\bar{z}:\mathbb{Z})[\varphi_{\mathbb{Z}}(\bar{x},\bar{y},\bar{z})]$ is in $\mathscr{L}_{\mathbb{Z}}$, we can proceed to remove the block of existential quantifiers using Cooper's method ([8, 26]). In fact, we can defer the elimination until all term quantifiers are gone.
2. *If for some $z \in \bar{z}$, z occurs inside some boundary terms, we eliminate z by Lemma 7.*

Algorithm 2 (Elimination of Term Variables).
We assume that formulas with quantifiers on term variables are in the form

$$(\exists \bar{x} : \mathsf{TA}) \left[\varphi_{\mathsf{kb}^+}(\bar{x},\bar{y},\bar{z}) \ \wedge\ \varphi_{\mathbb{Z}}(\bar{x},\bar{y},\bar{z})\right], \tag{7}$$

where $\bar{x}$, $\bar{y}$ are term variables, $\bar{z}$ are integer variables. Note that $\bar{z}$ may occur inside boundary terms in $\varphi_{\mathsf{kb}^+}(\bar{x},\bar{y},\bar{z})$, and $\bar{x},\bar{y}$ may occur inside weight function in $\varphi_{\mathbb{Z}}(\bar{x},\bar{y},\bar{z})$.

Repeatedly apply the following subprocedures (A) and (B) to (7) *until $\bar{x} = \emptyset$.*

(A) *Depth Reduction.* ***Repeat*** *(a),(b),(c) in the order* ***while*** $(\forall \mathbf{x} \in \bar{x})\, \mathsf{depth}_{\varphi_{\mathsf{kb}^+}}(\mathbf{x}) > \mathbf{0}$.

(a) Selection. *Select a α-typed variable $x \in \bar{x}$ for some $\alpha = (\mathsf{s}_1^\alpha, \ldots, \mathsf{s}_{\mathsf{ar}(\alpha)}^\alpha)$. This selection is always possible as $\mathsf{depth}_{\varphi_{\mathsf{kb}^+}}(x) > 0$. We require that in the next run of (a), we choose one of the variables generated by* this *run of (b). I.e., the variable selection is done in depth-first manner. This is crucial to guarantee that a run eventually leaves (A). Let $\bar{x}' \equiv \bar{x} \setminus x$.*

(b) Decomposition. *We rewrite* (7) *to:*

$$\left(\exists \bar{x}', x_1, \ldots, x_{\mathsf{ar}(\alpha)}, x : \mathsf{TA}\right) \Big[\mathsf{Is}_\alpha(x) \ \wedge \bigwedge_{1\le i\le \mathsf{ar}(\alpha)} \mathsf{s}_i^\alpha(x) = x_i$$
$$\wedge\ \varphi_{\mathsf{kb}^+}(\bar{x},\bar{y},\bar{z}) \ \wedge\ \varphi_{\mathbb{Z}}(\bar{x},\bar{y},\bar{z})\Big]. \tag{8}$$

(c) Simplification. *Exhaustively apply the following simplification rules to φ_{kb^+} and $\varphi_{\mathbb{Z}}$ in* (8)*:*

(1) replace $\mathsf{s}_i^\alpha(x)$ by x_i $(1 \le i \le \mathsf{ar}(\alpha))$;

(2) replace x^{w} by $\Sigma_{i=1}^{\mathsf{ar}(\alpha)} x_i^{\mathsf{w}} + \mathsf{W}(\alpha)$;

(3) replace $x \prec_n^\sharp t$ by depth-reduction *($x \prec_n^\sharp t$);*

(4) similar for $t \prec_n^\sharp x$, $x \leqq_n^\sharp t$ and $t \leqq_n^\sharp x$.

The existence of depth-reduction *follows from Lemma 3. Let the resulting formula be*

$$\left(\exists \bar{x}', x_1, \ldots, x_{\mathsf{ar}(\alpha)}, x : \mathsf{TA}\right) \Big[\mathsf{Is}_\alpha(x) \ \wedge \bigwedge_{1\le i\le \mathsf{ar}(\alpha)} \mathsf{s}_i^\alpha(x) = x_i$$
$$\varphi'_{\mathsf{kb}^+}(\bar{x}', \mathsf{s}_1^\alpha(x), \ldots, \mathsf{s}_{\mathsf{ar}(\alpha)}^\alpha(x), \bar{y}, \bar{z}) \ \wedge\ \varphi'_{\mathbb{Z}}(\bar{x}', \mathsf{s}_1^\alpha(x), \ldots, \mathsf{s}_{\mathsf{ar}(\alpha)}^\alpha(x), \bar{y}, \bar{z})\ \Big]. \tag{9}$$

It is now clear that if x occurs in φ'_{kb^+} and $\varphi'_{\mathbb{Z}}$ it occurs inside some of $\mathsf{s}_1^\alpha(x), \ldots, \mathsf{s}_{\mathsf{ar}(\alpha)}^\alpha(x)$. Since

$$\left(\forall x_1, \ldots, x_{\mathsf{ar}(\alpha)} : \mathsf{TA}\right)\left(\exists x : \mathsf{TA}\right)\Big[\ \mathsf{Is}_\alpha(x) \ \wedge \bigwedge_{1\le i\le \mathsf{ar}(\alpha)} \mathsf{s}_i^\alpha(x) = x_i\ \Big]$$

is valid in $\mathfrak{A}_{\mathsf{TA}}$*, we can replace in* (9)*,* $\mathsf{s}_1^{\alpha}(x), \ldots, \mathsf{s}_{\mathsf{ar}(\alpha)}^{\alpha}(x)$*, respectively, by* $x_1, \ldots, x_{\mathsf{ar}(\alpha)}$*, and hence remove* $\bigwedge_{1\le i\le \mathsf{ar}(\alpha)} \mathsf{s}_i^{\alpha}(x) = x_i$*,* $\mathsf{Is}_{\alpha}(x)$ *together with* $\exists x$*, obtaining*

$$\Big(\exists\, \bar{x}', x_1, \ldots, x_{\mathsf{ar}(\alpha)} : \mathsf{TA}\Big) \Big[\ \varphi'_{\mathsf{kb}^+}(\bar{x}', x_1, \ldots, x_{\mathsf{ar}(\alpha)}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{z}}) \\ \wedge\ \varphi'_{\mathbb{Z}}(\bar{x}', x_1, \ldots, x_{\mathsf{ar}(\alpha)}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{z}})\ \Big]. \quad (10)$$

(B) *Elimination.* ***Repeat*** *(B)* ***while*** $(\exists \mathbf{x} \in \bar{x})\, \mathsf{depth}_{\varphi_{\mathsf{kb}^+}}(\mathbf{x}) = \mathbf{0}$.
Take the x as in the guard condition, guess a DGOC for all terms related with x in gap order literals (by Lemma 1) and then eliminate x by Lemma 6.

Theorem 1. $\mathrm{Th}(\mathfrak{A}^{\mathbb{Z}}_{\mathsf{kb}^+})$ *is decidable, and hence so is* $\mathrm{Th}(\mathfrak{A}_{\mathsf{kb}})$*.*

Example 8. Let us go through an example with emphasis on the depth reduction. Due to space limitation, we only show *one* simple trace of the reduction. Consider in the LISP list structure the following formula

$$(\exists x)\ \Big[\ \mathsf{car}(x) \prec^{\mathsf{l}}_2 \mathsf{cdr}(\mathsf{cdr}(x))\ \wedge\ \mathsf{cdr}(\mathsf{cdr}(\mathsf{car}(x))) \prec^{\mathsf{l}}_3 y\ \Big], \quad (11)$$

where $\mathsf{depth}_{(11)}(x) = 3$. At the first run of (A), we introduce fresh variables x_1 and x_2 to replace $\mathsf{car}(x)$ and $\mathsf{cdr}(x)$, respectively. By a standard quantifier manipulation we obtain

$$(\exists x_1 \exists x_2)\ \Big[x_1 \prec^{\mathsf{l}}_2 \mathsf{cdr}(x_2)\ \wedge\ \mathsf{cdr}(\mathsf{cdr}(x_1)) \prec^{\mathsf{l}}_3 y\Big], \quad (12)$$

where $\mathsf{depth}_{(12)}(x_1) = 2$ and $\mathsf{depth}_{(12)}(x_2) = 1$, both less than $\mathsf{depth}_{(11)}(x)$. In the second run of (A), we pick x_1 and replace $x_1 \prec^{\mathsf{l}}_2 \mathsf{cdr}(x_2)$ by $\mathsf{car}(x_1) = \mathsf{car}(\mathsf{cdr}(x_2)) \wedge \mathsf{cdr}(x_1) \prec^{\mathsf{l}}_2 \mathsf{cdr}(\mathsf{cdr}(x_2))$ (which is one of several choices). We obtain

$$(\exists x_2 \exists x_{11} \exists x_{12})\ \Big[x_{11} = \mathsf{car}(\mathsf{cdr}(x_2))\ \wedge\ x_{12} \prec^{\mathsf{l}}_2 \mathsf{cdr}(\mathsf{cdr}(x_2))\ \wedge\ \mathsf{cdr}(x_{12}) \prec^{\mathsf{l}}_3 y\Big]. \quad (13)$$

At this point we have $\mathsf{depth}_{(13)}(x_{11}) = 0$ and the run enters (B). In this case we can immediately remove $\exists x_{11}$, obtaining

$$(\exists x_2 \exists x_{12})\ \Big[x_{12} \prec^{\mathsf{l}}_2 \mathsf{cdr}(\mathsf{cdr}(x_2))\ \wedge\ \mathsf{cdr}(x_{12}) \prec^{\mathsf{l}}_3 y\Big], \quad (14)$$

where $\mathsf{depth}_{(14)}(x_{12}) = 1$ and $\mathsf{depth}_{(14)}(x_2) = 2$. At the third run of (A), we select x_{12}. The run could give us

$$(\exists x_2 \exists x_{121} \exists x_{122})\ \Big[x_{121} = \mathsf{car}(\mathsf{cdr}(\mathsf{cdr}(x_2))) \wedge x_{122} \prec^{\mathsf{l}}_2 \mathsf{cdr}(\mathsf{cdr}(x_2)) \wedge x_{122} \prec^{\mathsf{l}}_3 y\Big], \quad (15)$$

which as before by (B) simplifies to

$$(\exists x_2 \exists x_{122})\ \Big[x_{122} \prec^{\mathsf{l}}_2 \mathsf{cdr}(\mathsf{cdr}(x_2))\ \wedge\ x_{122} \prec^{\mathsf{l}}_3 y\Big]. \quad (16)$$

Still we have $\mathsf{depth}_{(16)}(x_{122}) = 0$ which justifies another run of (B). Let us take a gap order completion $x_{122} \prec^{\mathsf{l}}_2 \mathsf{cdr}(\mathsf{cdr}(x_2)) \prec^{\mathsf{l}}_1 y$ (which again is just one of many choices) and rewrite (16) to

$$(\exists x_2 \exists x_{122})\ \Big[x_{122} \prec^{\mathsf{l}}_2 \mathsf{cdr}(\mathsf{cdr}(x_2)) \prec^{\mathsf{l}}_1 y\Big]. \quad (17)$$

With the help of boundary functions, (17) reduces to

$$(\exists x_2)\ \Big[0^{\mathsf{w}}_{((\mathsf{cdr}(\mathsf{cdr}(x_2)))^{\mathsf{w}})} \prec^{\mathsf{l}}_2 \mathsf{cdr}(\mathsf{cdr}(x_2)) \prec^{\mathsf{l}}_1 y\Big]. \quad (18)$$

The fourth and the fifth runs of (A) (with the same trick of quantifier manipulation) give us

$$(\exists x_{222})\left[\mathsf{0}^{\mathsf{w}}_{(x^{\mathsf{w}}_{222})} \prec^{\mathsf{l}}_{2} x_{222} \prec^{\mathsf{l}}_{1} y\right]. \tag{19}$$

After that the run comes back again to (B) as $\mathsf{depth}_{(19)}(x_{222}) = 0$. Here we have to reduce term quantifiers to integer quantifiers in that x_{222} also appears in boundary terms. By Lemma 6, (19) is equivalent to

$$(\exists z)\left[\mathsf{0}^{\mathsf{w}}_{(z)} \prec^{\mathsf{l}}_{3} y \;\wedge\; \mathsf{Tree}^{\mathsf{cons}}(z)\right], \tag{20}$$

which simplifies to $\mathsf{0}^{\mathsf{w}}_{(y^{\mathsf{w}})} \prec^{\mathsf{l}}_{3} y \wedge \mathsf{Tree}^{\mathsf{cons}}(y^{\mathsf{w}})$, and in turn to

$$\mathsf{0}^{\mathsf{w}}_{(y^{\mathsf{w}})} \prec^{\mathsf{l}}_{3} y, \tag{21}$$

as $\mathsf{0}^{\mathsf{w}}_{(y^{\mathsf{w}})} \prec^{\mathsf{l}}_{3} y$ implies $\mathsf{Tree}^{\mathsf{cons}}(y^{\mathsf{w}})$. It is not hard to verify that (21) implies (11) as desired. (We do not have equivalence because this is just one trace of reduction.)

We note that the depth reduction of a variable is at the expense of increasing the depth of a term on the other side of a relation. This happens when φ contains $x \star t$ (or $t \star x$) and $\mathsf{depth}_{\varphi}(x) > 0$. For example, from (12) to (13), the depth of x_2 increases by 1. Moreover, the depth reduction in general introduces more existential quantifiers and more equalities in the matrix (e.g., also in the reduction from (12) to (13)). However, in each transformation, the number of open gap order literals in each resulting primitive formula is no more than that in the original (primitive) formula. Moreover, the final elimination procedure removes at least one open gap order literal if the eliminated variable occurs in such literals (e.g., from (17) to (18) and from (19) to (20)). When all open gap order literals are gone, the depths of terms will be strictly decreasing. This forces the run to eventually leave (A) and from then on to stay in (B) until all existential quantifiers are removed.

5 Presence of a 0-weight unary function

As mentioned earlier, the presence of a unary function α_0 of weight 0 in Σ simplifies the elimination procedure. Intuitively, the existence of α_0 makes $\prec^{\mathsf{w}}$ and $\prec^{\mathsf{p}}$ dense almost everywhere except around atoms. This follows from the fact that $\mathsf{1}^{\mathsf{w}}_{(m)}$ and $\mathsf{1}^{\mathsf{p}}_{(m,p)}$ are undefined (i.e., no maximum) except when α_p is an atom and $m = \mathsf{W}(\alpha_p)$. Accordingly, if u is not an atom, then for any $n \geq 1$, $u \prec^{\mathsf{w}}_{n} v$ (resp. $u \prec^{\mathsf{p}}_{n} v$) is equivalent to $u \prec^{\mathsf{w}} v$ (resp. $u \prec^{\mathsf{p}} v$). Also, it suffices for $\mathscr{L}^{\mathbb{Z}}_{\mathsf{kb}^{+}}$ to only have lower boundary functions in order to decompose gap orders. More details are given in the extended version of this paper.

6 Conclusion

We showed the decidability of the first-order theory of term algebras with Knuth-Bendix order by quantifier elimination. Our method combines the extraction of integer constraints from term constraints with the reduction of quantifiers on term variables to quantifiers on integer variables. In fact, we established the decidability of a much more expressive theory.

Two problems related to practical complexity need further investigation. First, as a rule of thumb, more expressive power means higher complexity. Even if the theoretical complexity bound is the same, in practice the efficiency will be compromised. It is worthwhile to search for the smallest extension of KBO which admits quantifier elimination. Second, the elimination is intrinsically limited to processing quantified variables one at a time. We plan to extend the method in [30] to eliminate a block of quantifiers of the same kind in one step. We believe this will be a significant improvement in pragmatic terms, since in most applications the quantifier alternation depth is small.

We also plan to investigate the decidability issue of the first-order theory of KBO in the term domain with variables [13, 1].

7 Acknowledgments

We thank Aaron Bradley for his comments on an earlier version of this paper. We thank the anonymous referees for their careful reading and suggestions.

References

1. Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, Cambridge, UK, 1999.
2. Rolf Backofen. A complete axiomatization of a theory with feature and arity constraints. *Journal of Logical Programming*, 24(1&2):37–71, 1995.
3. Hubert Comon. Solving symbolic ordering constraints. *International Journal of Foundations of Computer Science*, 1(4):387–411, 1990.
4. Hubert Comon and Catherine Delor. Equational formulae with membership constraints. *Information and Computation*, 112(2):167–216, 1994.
5. Hubert Comon and Pierre Lescanne. Equational problems and disunification. *Journal of Symbolic Computation*, 7:371–425, 1989.
6. Hubert Comon and Ralf Treinen. Ordering constraints on trees. In *Proceedings of the* 19*th International Colloquium on Trees in Algebra and Programming (CAAP'94)*, volume 787 of *Lecture Notes in Computer Science*, pages 1–14, Edinburgh, U.K., Apr 1994. Springer-Verlag.
7. Hubert Comon and Ralf Treinen. The first-order theory of lexicographic path orderings is undecidable. *Theoretical Computer Science*, 176(1-2):67–87, 1997.
8. D. C. Cooper. Theorem proving in arithmetic without multiplication. In *Machine Intelligence*, volume 7, pages 91–99. American Elsevier, 1972.
9. Nachum Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 7:279–301, 1982.
10. H. B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, 2001.
11. Wilfrid Hodges. *Model Theory*. Cambridge University Press, Cambridge, UK, 1993.
12. Jean-Pierre Jouannaud and Mitsuhiro Okada. Satisfiability of systems of ordinal notation with the subterm property is decidable. In *Proceedings of the* 18*th International Colloquium on Automata, Languages and Programming (ICALP'91)*, volume 510 of *Lecture Notes in Computer Science*, pages 455–468. Springer-Verlag, 1991.
13. Donald E. Knuth and Peter Bendix. Simple word problems in universal algebras. In *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970. Reprinted in *Automation of Reasoning, Vol. 2* Jürgen Siekmann and G. Wrightson, editors, pp. 342-376, Springer-Verlag, 1983.
14. Konstantin Korovin and Andrei Voronkov. A decision procedure for the existential theory of term algebras with the Knuth-Bendix ordering. In *Proceedings of the* 15*th IEEE Symposium on Logic in Computer Science (LICS'00)*, pages 291 – 302, IEEE Computer Society Press, 2000.

15. Konstantin Korovin and Andrei Voronkov. Knuth-Bendix constraint solving is NP-complete. In *Proceedings of 28th International Colloquium on Automata, Languages and Programming (ICALP'01)*, volume 2076 of *Lecture Notes in Computer Science*, pages 979–992. Springer-Verlag, 2001.
16. Konstantin Korovin and Andrei Voronkov. The decidability of the first-order theory of the Knuth-Bendix order in the case of unary signatures. In *Proceedings of the 22th Conference on Foundations of Software Technology and Theoretical Computer Science, (FSTTCS'02)*, volume 2556 of *Lecture Notes in Computer Science*, pages 230–240. Springer-Verlag, 2002.
17. Viktor Kuncak and Martin Rinard. On the theory of structural subtyping. Technical Report MIT-LCS-TR-879, Massachusetts Institute of Technology, January 2003.
18. Viktor Kuncak and Martin Rinard. The structural subtyping of non-recursive types is decidable. In *Proceedings of the* 18*th IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 96–107. IEEE Computer Society Press, 2003.
19. M. J. Maher. Complete axiomatizations of the algebras of finite, rational and infinite tree. In *Proceedings of the 3th IEEE Symposium on Logic in Computer Science (LICS'88)*, pages 348–357. IEEE Computer Society Press, 1988.
20. A. I. Mal'cev. Axiomatizable classes of locally free algebras of various types. In *The Metamathematics of Algebraic Systems, Collected Papers*, chapter 23, pages 262–281. North Holland, 1971.
21. Paliath Narendran and Michael Rusinowitch. The theory of total unary RPO is decidable. In *Proceedings of the* 1*st International Conference on Computational Logic (CL 2000)*, volume 1861 of *Lecture Notes in Artificial Intelligence*, pages 660–672. Springer-Verlag, 2000.
22. Paliath Narendran, Michael Rusinowitch, and Rakesh M. Verma. RPO constraint solving is in NP. In *Proceedings of the 12th International Workshop on Computer Science Logic (CSL'98)*, volume 1584 of *Lecture Notes in Computer Science*, pages 385 – 398. Springer-Verlag, 1999.
23. Robert Nieuwenhuis. Simple LPO constraint solving methods. *Information Processing Letters*, 47(2):65–69, 1993.
24. Robert Nieuwenhuis and J. Rivero. Solved forms for path ordering constraints. In *Proceedings of* 10*th International Conference on Rewriting Techniques and Applications (RTA'99)*, volume 1631 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 1999.
25. Robert Nieuwenhuis and Albert Rubio. Theorem proving with ordering and equality constrained clauses. *Journal of Symbolic Computation*, 19(4):321–351, 1995.
26. C. R. Reddy and D. W. Loveland. Presburger arithmetic with bounded quantifier alternation. In *Proceedings of the* 10*th Annual Symposium on Theory of Computing*, pages 320–325. ACM Press, 1978.
27. Tatiana Rybina and Andrei Voronkov. A decision procedure for term algebras with queues. *ACM Transactions on Computational Logic*, 2(2):155–181, 2001.
28. Ralf Treinen. A new method for undecidability proofs of first order theories. *Journal of Symbolic Computation*, 14:437–457, 1992.
29. Ting Zhang, Henny Sipma, and Zohar Manna. Decision procedures for recursive data structures with integer constraints. In *Proceedings of the* 2*nd International Joint Conference on Automated Reasoning (IJCAR'04)*, volume 3097 of *Lecture Notes in Computer Science*, pages 152–167. Springer-Verlag, 2004.
30. Ting Zhang, Henny Sipma, and Zohar Manna. Term algebras with length function and bounded quantifier alternation. In *Proceedings of the* 17*th International Conference on Theorem Proving in Higher Order Logics (TPHOLs'04)*, volume 3223 of *Lecture Notes in Computer Science*, pages 321–336. Springer-Verlag, 2004.