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 $\bot$ 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 $x \prec^{\mathsf{kb}}_n t$ or $t \prec^{\mathsf{kb}}_n x$ 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 $\prec^{\mathsf{kb}}_n$ $(n \geq 0)$ such that*

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

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

A gap order $u \prec^{\sharp}_n v$ $(n \geq 1)$ states that "u is less than v w.r.t. $\prec^{\sharp}$, and there are *at least* $n-1$ elements in between." Similarly, $u \leqq^{\sharp}_n v$ $(n \geq 1)$ states that "u is less than v w.r.t. $\prec^{\sharp}$, and there are *exactly* $n-1$ elements in between". Note that $\prec^{\sharp}_1$ is just $\prec^{\sharp}$, $\prec^{\sharp}_0$ is $\preceq^{\sharp}$, $\leqq^{\sharp}_0$ is $=$, and we have $u \prec^{\sharp}_n v \leftrightarrow u \prec^{\sharp}_{n+1} v \vee u \leqq^{\sharp}_n v$.

Example 1. The formula $\exists x(u \prec^{\mathsf{l}} x \prec^{\mathsf{l}} v)$ reduces to $u \prec^{\mathsf{l}}_2 v$ if u, v do not contain x.

3.4 Boundary Functions

Consider the formula $u \leqq^{\mathsf{w}}_1 v$. 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, $f(n) = \bot$ *and* $f(n, p) = \bot$*, 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*

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

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 $\bar{\boldsymbol{u}} = \langle u_1, \dots, u_k\rangle, \bar{\boldsymbol{v}} = \langle v_1, \dots, v_k\rangle$ 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 $\bar{\boldsymbol{u}} \prec^{k;\mathsf{kb}} \bar{\boldsymbol{v}}$ if and only if* (1) *holds.*

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

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

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

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

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

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

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 $\langle \perp, \ldots \perp \rangle$ 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 $\mathscr{L}^{\mathbb{Z}}_{\mathsf{kb}^+}$.

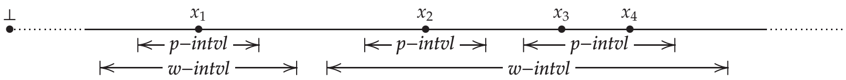
3.8 Delineated Gap Order Completion

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* $\mathscr{L}^{\mathbb{Z}}_{\mathsf{kb}^+}$ *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* $\mathscr{L}^{\mathbb{Z}}_{\mathsf{kb}^+}$*) in which x does not appear and* $\mathsf{s}^{\alpha_p}_i x$ $(1 \le i \le k)$ *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 $\alpha_p = (\mathsf{s}_1^{\alpha_p}, \ldots, \mathsf{s}_k^{\alpha_p})$ 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 $\varphi(\mathsf{s}_1^{\alpha_p} x, \ldots, \mathsf{s}_k^{\alpha_p} x)$ in which x does not occur and $\mathsf{s}_i^{\alpha_p} x$ ($1 \leq i \leq k$) 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 $\star \in \{\prec_n^{k;\mathsf{kb}}, \prec_n^{k;\mathsf{w}}, \prec_n^{k;\mathsf{p}}, \prec_n^{k;\mathsf{l}}, \prec_n^{k;\mathsf{pl}}, \leqq_n^{k;\mathsf{kb}}, \leqq_n^{k;\mathsf{w}}, \leqq_n^{k;\mathsf{p}}, \leqq_n^{k;\mathsf{l}}, \leqq_n^{k;\mathsf{pl}}\}$, and U, V be k-tuples of the same weight.*

1. *If $U = \langle u_1, \ldots, u_k \rangle$ is an ordinary tuple, then $U \star V$ ($V \star U$) can be effectively reduced to an equivalent quantifier-free formula $\varphi(u_1, \ldots, u_k)$ (in $\mathscr{L}_{\mathsf{kb}^+}^{\mathbb{Z}}$) in which u_i ($1 \leq i \leq k$) 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 Th($\mathfrak{A}^{\mathbb{Z}}_{\mathsf{kb}^+}$)

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

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

where $A_i(x, \bar{y})$ are ($1 \leq i \leq n$) 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 $\alpha = (\mathsf{s}^{\alpha}_1, \ldots, \mathsf{s}^{\alpha}_{\mathsf{ar}(\alpha)})$. 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:*
$$\Big(\exists \bar{x}', x_1, \ldots, x_{\mathsf{ar}(\alpha)}, x : \mathsf{TA}\Big) \Big[\mathsf{Is}_{\alpha}(x) \ \wedge \bigwedge_{1 \le i \le \mathsf{ar}(\alpha)} \mathsf{s}^{\alpha}_i(x) = x_i \\ \wedge \ \varphi_{\mathsf{kb}^+}(\bar{x}, \bar{y}, \bar{z}) \ \wedge \ \varphi_{\mathbb{Z}}(\bar{x}, \bar{y}, \bar{z})\Big]. \tag{8}$$
- *(c)* Simplification. *Exhaustively apply the following simplification rules to φ_{kb^+} and $\varphi_{\mathbb{Z}}$ in* (8)*:*
  - *(1) replace $\mathsf{s}^{\alpha}_i(x)$ by x_i $(1 \le i \le \mathsf{ar}(\alpha))$;*
  - *(2) replace x^{w} by $\Sigma^{\mathsf{ar}(\alpha)}_{i=1} x^{\mathsf{w}}_i + \mathsf{W}(\alpha)$;*
  - *(3) replace $x \prec^{\sharp}_n t$ by* depth-reduction $(\ x \prec^{\sharp}_n t\ )$;
  - *(4) similar for $t \prec^{\sharp}_n x$, $x \leqq^{\sharp}_n t$ and $t \leqq^{\sharp}_n x$.*

  The existence of depth-reduction *follows from Lemma 3. Let the resulting formula be*
$$\Big(\exists \bar{x}', x_1, \ldots, x_{\mathsf{ar}(\alpha)}, x : \mathsf{TA}\Big) \Big[\mathsf{Is}_{\alpha}(x) \ \wedge \bigwedge_{1 \le i \le \mathsf{ar}(\alpha)} \mathsf{s}^{\alpha}_i(x) = x_i \\ \varphi'_{\mathsf{kb}^+}(\bar{x}', \mathsf{s}^{\alpha}_1(x), \ldots, \mathsf{s}^{\alpha}_{\mathsf{ar}(\alpha)}(x), \bar{y}, \bar{z}) \ \wedge \ \varphi'_{\mathbb{Z}}(\bar{x}', \mathsf{s}^{\alpha}_1(x), \ldots, \mathsf{s}^{\alpha}_{\mathsf{ar}(\alpha)}(x), \bar{y}, \bar{z})\ \Big]. \tag{9}$$
  It is now clear that if x occurs in φ'_{kb^+} and $\varphi'_{\mathbb{Z}}$ it occurs inside some of $\mathsf{s}^{\alpha}_1(x), \ldots, \mathsf{s}^{\alpha}_{\mathsf{ar}(\alpha)}(x)$. Since
$$\Big(\forall x_1, \ldots, x_{\mathsf{ar}(\alpha)} : \mathsf{TA}\Big)\Big(\exists x : \mathsf{TA}\Big)\Big[\ \mathsf{Is}_{\alpha}(x) \ \wedge \bigwedge_{1 \le i \le \mathsf{ar}(\alpha)} \mathsf{s}^{\alpha}_i(x) = x_i\ \Big]$$

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* $\bigwedge_{1\le i\le \mathsf{ar}(\alpha)} \mathsf{s}_i^\alpha(x) = x_i$*,* $\mathsf{ls}_\alpha(x)$ *together with* $\exists x$*, obtaining*

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

(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 18th 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 1st 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 10th 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 10th 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 2nd 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 17th 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.

A Preliminaries

We assume the first-order syntactic notions of variables, parameters and quantifiers, and semantic notions of structures, satisfiability and validity as in [10].

A *signature* Σ is a set of *parameters* (*function symbols* and *predicate symbols*) each of which is associated with an arity. The function symbols with arity 0 are also called *constants*. The set of *Σ-terms* $\mathcal{T}(\Sigma, \mathcal{X})$ is recursively defined by: (i) every constant $c \in \Sigma$ or variable $x \in \mathcal{X}$ is a term, and (ii) if $f \in \Sigma$ is an n-place function symbol and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term. Equality $=$ is always included as a binary predicate symbol. We reserve $\equiv$ to denote the syntactic equality. If φ is a formula, we use $\mathcal{T}(\varphi)$ to denote the set of terms occurring in φ, $\mathcal{V}(\varphi)$ to denote the set of variables in φ.

An *atomic formula* is a formula of the form $P(t_1, \ldots, t_n)$ where P is an n-place predicate symbol and $t_1, \ldots, t_n$ are terms. A *literal* is an atomic formula or its negation. A variable occurs *free* in a formula if it is not in the scope of a quantifier. A formula without quantifiers is called *quantifier-free*. A *ground formula* is a formula with no variables. A *sentence* is a formula in which no variable occurs free. Every quantifier-free formula can be put into *disjunctive normal form*, that is, a disjunction of conjunctions. A formula $\psi(\bar{x})$ can be put into *prenex form* $Q_1 y_1, \ldots, Q_n y_n\ \varphi(\bar{x}, y_1, \ldots, y_n)$, where Q_i's are either $\exists$ or $\forall$ and $\varphi(\bar{x}, y_1, \ldots, y_n)$ is quantifier-free, called the *matrix* of ψ.

A *Σ-structure* $\mathfrak{A}$ is a tuple $\langle A, I \rangle$ where A is a non-empty domain and I is a function that associates each n-place function symbol f (resp. predicate symbol P) with an n-place function $f^{\mathfrak{A}}$ (resp. relation $P^{\mathfrak{A}}$) on A. We use Gothic letters (like $\mathfrak{A}$) for structures and Roman letters (like A) for the underlying domain. We usually denote $\mathfrak{A}$ by $\langle A; \Sigma \rangle$ which is called the *signature* of $\mathfrak{A}$. A *variable assignment* σ (in $\mathfrak{A}$) is a function that assigns each variable an element of A. We use $[\![x]\!]\sigma$ to denote the assigned values of x under σ and $[\![\varphi]\!]\sigma$ for the truth value of φ under σ. $\mathfrak{A} \vdash [\![\varphi]\!]\sigma$ means φ is true under σ. A formula φ is *satisfiable*, denoted by $\mathfrak{A} \vdash_{\mathsf{s}} \varphi$, if $\mathfrak{A} \vdash [\![\varphi]\!]\sigma$ for some σ; is *unsatisfiable*, denoted by $\mathfrak{A} \nvdash_{\mathsf{s}} \varphi$, if $\mathfrak{A} \vdash [\![\varphi]\!]\sigma$ for no σ; is valid, denoted by $\mathfrak{A} \vdash \varphi$, if $\mathfrak{A} \vdash [\![\varphi]\!]\sigma$ for any σ. A formula φ is valid if and only if $\neg\varphi$ is unsatisfiable.

$\mathfrak{A}$ is a *model* of a set T of sentences if every sentence in T is true in $\mathfrak{A}$. A sentence φ is *(logically) implied* by T (or *T-valid*), written $T \models \varphi$, if φ is true in every model of T. Similarly we say that φ is *T-satisfiable* if φ is true in some model of T and it is *T-unsatisfiable* otherwise. The notions of (T-)validity and (T-)satisfiability naturally extend to a set of formulas. A *theory* T is a set of sentences that is closed under logical implication, that is, if $T \models \varphi$, then $\varphi \in T$. The theory of $\mathfrak{A}$, written $\mathrm{Th}(\mathfrak{A})$, is the set of all valid sentences in $\mathfrak{A}$. By $\mathrm{Th}^{\forall}(\mathfrak{A})$ we denote the quantifier-free subclass of $\mathrm{Th}(\mathfrak{A})$.

We use $\bar{x}$ to denote a set of variables, say, $x_1, \ldots, x_n$, and $\exists\bar{x}$ (resp. $\forall\bar{x}$) as an abbreviation of $\exists x_1, \ldots, \exists x_n$ (resp. $\forall x_1, \ldots, \forall x_n$). We write $\varphi(\bar{x})$ to mean that $\bar{x}$ occur free in φ. $(\exists\bar{x})\varphi(\bar{x})$ and $(\forall\bar{x})\varphi(\bar{x})$ are called *existential closure* and *universal closure* of $\varphi(\bar{x})$, respectively. If $\varphi(\bar{x})$ is quantifier-free, then $(\exists\bar{x})\varphi(\bar{x})$ is called $\exists_1$

and $(\forall \bar{x})\varphi(\bar{x})$ is called $\forall_1$. By satisfiability and validity of a quantifier-free theory, we actually mean the validity of $\exists_1$ and $\forall_1$ formulas, respectively.

A theory T is said to *admit quantifier elimination* if any formula can be equivalently (modulo T) and effectively transformed into a quantifier-free formula. If a theory admits quantifier elimination, then the truth value of any sentence is reducible to the truth value of a ground formula.

A *term algebra* (TA) of Σ with basis $\mathcal{X}$ is the structure $\mathfrak{A}$ whose domain is $\mathcal{T}(\Sigma, \mathcal{X})$ and for any n-place function symbol $f \in \Sigma$ and $t_1, \ldots, t_n \in \mathcal{T}(\Sigma, \mathcal{X})$, $f^{\mathfrak{A}}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$. We assume that Σ does not contain any predicate symbols except equality.

Presburger arithmetic is the first-order theory of addition in the arithmetic of integers. The corresponding structure is denoted by $\mathrm{PA} = \langle \mathbb{Z}; 0, +, < \rangle$. We use $\mathscr{L}_{\mathbb{Z}}$ to denote the formal language of PA.

We adopt conventional notations in discussion of orderings; we write $u \succ v$ for $v \prec u$, $u \preceq v$ for $u \prec v \vee u = v$ and $u \prec v \prec w$ for $u \prec v \wedge v \prec w$.

B Proofs of Lemmas

B.1 Proof of Lemma 1.

Proof. We show that any conjunction of positive literals in $\mathscr{L}_{\mathsf{kb}^+}$ is equivalent to a finite disjunction of delineated gap order completions.

Let φ_{kb^+} be a conjunction of positive literals in $\mathscr{L}_{\mathsf{kb}^+}$ and $\Sigma(\varphi_{\mathsf{kb}^+})$ be the set of all TA-terms in φ_{kb^+}. We describe our algorithm as a construction of a disjunctive decision tree. When we say "guess" we mean going down a disjunctive branch.

We can view φ_{kb^+} as a directed labeled graph G_{kb^+} in which $\Sigma(\varphi_{\mathsf{kb}^+})$ are vertex labels and $u \leqq^{\sharp}_n v$ (or $u \prec^{\sharp}_n v$) represents an *edge* with number label n and color label $\sharp$. An edge of the form $u \prec^{\sharp}_n v$ (resp. $u \leqq^{\sharp}_n v$) is called *stretchable* (resp. *rigid*). A path is *rigid* if every edge on the path is rigid; a path is *stretchable* if it contains a stretchable edge. A *length of a path* is the sum of all number labels on the path.

First we guess an equality partition for $\Sigma(\varphi_{\mathsf{kb}^+})$. All vertexes in the same equivalence class are merged by consolidating corresponding outgoing and incoming edges. We choose an arbitrary representative for each equivalence class as the node label.

Second we guess a linear order for $\Sigma(\varphi_{\mathsf{kb}^+})$. It amounts to adding to G_{kb^+} a Hamilton path, a path between two distinct vertexes of G_{kb^+} that visits each vertex exactly once. We call this path *primary path* and edges on the path *primary edges*. Note that the primary path corresponds to a gap order completion and it is a prototype of the delineated gap order completion we are to construct. At this phase, all primary edges are labeled by 1.

Third we insert boundary terms to delineate the primary path if necessary.

To reduce φ_{kb^+} to a DGOC, we need to remove all non-primary edges. A non-primary edge from u to v is called an *overpath* (resp. *underpath*) if its length is greater than (esp. less than) the length of the primary path from u to v. It is easily seen that a non-primary edge is redundant if it is neither an overpath nor an underpath.

Before removing non-primary edges we need to do the consistency check. There are two types of inconsistency.

1. G_{kb^+} contains a loop.
2. The color of a non-primary edge is incompatible with colors on the corresponding primary path. Let order $\{\mathsf{w}, \mathsf{p}, \mathsf{l}\}$ as $\mathsf{l} \prec^{\mathsf{c}} \mathsf{p} \prec^{\mathsf{c}} \mathsf{w}$. We say that the color c of a non-primary edge is incompatible with colors $\{c_i \mid 1 \leq i \leq k\}$ on the corresponding primary path if $c \neq max\{c_i \mid 1 \leq i \leq k\}$ with respect to $\prec^{\mathsf{c}}$. For example, $u \prec^{\mathsf{l}}_{n_1} v \prec^{\mathsf{p}}_{n_2} w$ is incompatible with neither $u \prec^{\mathsf{w}}_{n_3} w$ nor $u \prec^{\mathsf{l}}_{n_4} w$ while it is compatible with $u \prec^{\mathsf{p}}_{n_5} w$.

If inconsistency is detected, the constraint simplifies to false.

It is not hard to see that the existence of an overpath means the current gap order completion "over-approximates" φ_{kb^+}. To discharge a overpath, we refine the number labels on primary stretchable edges by distributing the extra length to each such edges covered by the overpath. For example consider the following primary path from u to v to w.

$$u \xrightarrow{p_1} v \xrightarrow{p_2} w, \tag{22}$$

where both two edges are stretchable. Suppose that a overpath from u to w labeled with $p > p_1 + p_2$. We replace the subgraph (22) by one of the following.

$$\bigvee_{\substack{n_1+n_2=p-p_1-p_2 \\ n_1,n_2\geq 0}} \qquad u \xrightarrow{p_1+n_1} v \xrightarrow{p_2+n_2} w. \tag{23}$$

Note that we do not add extra length to rigid primary edges, and if the corresponding primary path is not stretchable, then the constraint simplifies to false. After the distribution, we make a stretchable primary edge rigid if the corresponding overpath is rigid.

One by one, we can remove all overpath obtaining a graph in which all non-primary edges are underpath (assuming redundant edges have been removed). A stretchable underpath is also redundant and can be removed. If there exists a rigid underpath, then constraint simplifies to false.

Eventually we obtain a decision tree, each leaf of which, if not simplified to false, corresponds to a delineated gap order completion, and hence φ_{kb^+} can be equivalently transformed to a finite disjunction of GDOCs.

□

Lemma 8. *Let $C_{\mathsf{kb}^+} : \bigwedge_{i=1}^{n} t_i \rhd_i u \wedge \bigwedge_{i=1}^{m} u \rhd_{n+i} t_{n+i}$ be a conjunction of positive literals in $\mathscr{L}_{\mathsf{kb}^+}$ where $\rhd_j$ are gap orders and u is an ordinary term. Then C_{kb^+} is equivalent to a finite disjunction of delineated gap order completions and in each of GDOCs the number of open order literals is no more than that in C_{kb^+}. Moreover, if both the predecessor and the successor of x in the GDOC are boundary terms, then the GDOC has one fewer open order literals than $C_{\mathsf{kb}^+}(x)$ providing $C_{\mathsf{kb}^+}(x)$ contains at least one open order literals.*

Proof. By Lemma 1 it suffices to show that in each of GDOCs the number of open order literals is no more than that in C_{kb^+}. We assume there are $n_1 \leq n$ ordinary terms which are less than u and $m_1 \leq m$ ordinary terms which are greater than u. So in C_{kb^+} there are $m_1 + n_1$ open order literals and $m_1 + n_1 + 1$ ordinary terms. In each of GDOCs obtained by the decision tree construction, the number of ordinary terms can not exceed $m_1 + n_1 + 1$ as no new ordinary terms are introduced. It is not hard to argue that in an linear order containing at most $m_1 + n_1 + 1$ ordinary terms, the number of open order literals are at most $m_1 + n_1$. Note that the introduction of boundary terms (to delineate a gap order) and the merge of two vertexes (to equalize two terms) can only decrease the number of open order literals. Moreover, suppose $C_{\mathsf{kb}^+}(x)$ contains at least one open order literals. Without loss of generality, let us assume $n_1 > 0$. Since both the predecessor and the successor of x in the GDOC are boundary terms, x is separated from other ordinary terms, and so there are at most $n_1 - 1$ open order literals at the left-hand side of x (in the GDOC). Also there are at most m_1 open order literals at the right-hand side of x. Therefore, the total number of open order literals is $m_1 + n_1 - 1$, one less than that in $C_{\mathsf{kb}^+}(x)$. □

B.2 Proof of Lemma 2.

Proof. We show that any formula in $\mathscr{L}^{Z}_{\mathsf{kb}^+}$ can be effectively reduced to an equivalent formula in which no boundary terms appear inside selectors or the weight function.

First we show how to eliminate selectors in front of boundary terms. Suppose φ contains a selector term $\mathsf{s}_i^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m)}$, where $\mathsf{ar}(\alpha_p) = k$. We need to consider four cases.

Case 1. $\mathsf{0}^{\mathsf{w}}_{(m)}$ **is not well-defined.** In this case $\mathsf{0}^{\mathsf{w}}_{(m)} = \perp$, and hence $\mathsf{s}_i^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m)}$ simplifies to $\perp$.

Case 2. $\mathsf{0}^{\mathsf{w}}_{(m)}$ **is well-defined, but** $\mathsf{Tree}^{(\alpha_p)}(m)$ **is false.** Since $\mathsf{Tree}^{(\alpha_p)}(m)$ does not hold, $\mathsf{0}^{\mathsf{w}}_{(m)}$ can not be of type α_p. So $\mathsf{s}_i^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m)}$ simplifies to $\mathsf{0}^{\mathsf{w}}_{(m)}$.

Case 3. $\mathsf{Tree}^{(\alpha_p)}(m)$ **is true but there exists** p' **such that** $p < p' \leq |\Sigma|$ **and** $\mathsf{Tree}^{(\alpha_{p'})}(m)$ **holds.** As in the previous case, $\mathsf{0}^{\mathsf{w}}_{(m)}$ can not be of type α_p, and hence $\mathsf{s}_i^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m)}$ simplifies to $\mathsf{0}^{\mathsf{w}}_{(m)}$.

Case 4. $\mathsf{Tree}^{(\alpha_p)}(m)$ **is true and for any** p' **such that** $p < p' \leq |\Sigma|$**,** $\mathsf{Tree}^{(\alpha_{p'})}(m)$ **does not hold.** The condition says that $\mathsf{0}^{\mathsf{w}}_{(m)}$ is of type α_p, i.e., $\mathsf{0}^{\mathsf{w}}_{(m)} = \alpha(t_1, \ldots, t_{\mathsf{ar}(\alpha_p)})$. It is not hard to argue that $\langle t_1, \ldots, t_{\mathsf{ar}(\alpha_p)} \rangle$ should be the smallest k-tuple of weight n. Therefore, $\mathsf{s}_i^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m)}$ simplifies to $\mathsf{0}^{\mathsf{w}}_{(\mathsf{SCW}^k_i(m - \mathsf{W}(\alpha_p)))}$, where $\mathsf{SCW}^k_i(n)$ returns the weight of the i^{th} component of a k-tuple which is the smallest (with respect to $\prec^{k;\mathsf{kb}}$) k-tuples of weight n. The definition of $\mathsf{SCW}^k_i(n)$ is given in Appendix E. Note that $\mathsf{SCW}^k_i(n)$ and all these conditions can be expressed in Presburger arithmetic.

Similarly, if a selector term $\mathsf{s}_j^{\alpha_p} \mathsf{0}^{\mathsf{p}}_{(m,p')}$, where $\mathsf{ar}(\alpha_p) = k$, appears in φ, we have three cases to consider.

Case 1. $\mathsf{0}^{\mathsf{p}}_{(m,p')}$ **is not well-defined.** $\mathsf{s}_j^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m,p')}$ simplifies to $\perp$.

Case 2. $\mathsf{0}^{\mathsf{p}}_{(m,p')}$ **is well-defined, but** $p \neq p'$**.** $\mathsf{s}_j^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m,p')}$ simplifies to $\mathsf{0}^{\mathsf{w}}_{(m,p')}$.

Case 3. $\mathsf{0}^{\mathsf{p}}_{(m,p')}$ **is well-defined, and** $p = p'$**.** $\mathsf{s}_j^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m,p')}$ simplifies to $\mathsf{0}^{\mathsf{p}}_{(\mathsf{SCW}^k_j(m - \mathsf{W}(\alpha_p)))}$.

Similarly we can handle terms of the form $\mathsf{s}_j^{\alpha_p} \mathsf{1}^{\mathsf{w}}_{(m)}$ or $\mathsf{s}_j^{\alpha_p} \mathsf{1}^{\mathsf{p}}_{(m,p')}$.

Next we note that boundary terms in the weight function can be reduced to integer terms. For example, $(\mathsf{0}^{\mathsf{w}}_{(m)})^{\mathsf{w}}$ reduces to m if $\mathsf{WD}_{\mathsf{0}^{\mathsf{w}}}(m)$ and to $\mathsf{W}(\perp)$ otherwise.

Though we do the transformation at semantic level by case distinction, the corresponding syntactic transformations (that we really need) can be recovered as follows. Let $\varphi(t)$ be a formula in which term t occurs. Consider there are n cases in each of which under the precondition θ_i, t is reduced to t_i ($1 \leq i \leq n$). Then $\varphi(t)$ is rewritten as

$$\bigwedge_{i=1}^{n} \big(\theta_i \rightarrow \varphi(t_i)\big), \tag{24}$$

or equivalently to

$$\bigvee_{i=1}^{n} \big(\theta_i \wedge \varphi(t_i)\big). \tag{25}$$

It is easy to verify that all preconditions given above are expressible in Presburger arithmetic formulas. For example, consider Case 4 in the reduction of $\mathsf{s}_i^{\alpha_p} 0^{\mathsf{w}}_{(m)}$. The precondition states that

> $\mathsf{Tree}^{(\alpha_p)}(m)$ is true and for any p' such that $p < p' \leq |\Sigma|$, $\mathsf{Tree}^{(\alpha_{p'})}(m)$ does not hold,

which can be expressed formally as

$$\mathsf{Tree}^{(\alpha_p)}(m) \ \wedge \bigwedge_{p<p'\leq|\Sigma|} \neg\mathsf{Tree}^{(\alpha_{p'})}(m). \tag{26}$$

Repeating this transformation, eventually all selectors in front of boundary terms are pealed off and we obtain φ' in which no boundary terms occur inside selectors.

Also it is easily seen from (25) that the transformation involves disjunctive splittings and since θ_i are Presburger formulas, the transformation does not increase the number of open order literals in each of resulting conjunctions. □

B.3 Proof of Lemma 3.

Proof. This lemma states the following: If x is of type α_p with $\alpha_p = (\mathsf{s}_1^{\alpha_p}, \ldots, \mathsf{s}_k^{\alpha_p})$ and t is an arbitrary term, then $x \star t$ where $\star \in \{\prec^{\mathsf{w}}_n, \prec^{\mathsf{p}}_n, \prec^{\mathsf{l}}_n, \leqq^{\mathsf{w}}_n, \leqq^{\mathsf{p}}_n, \leqq^{\mathsf{l}}_n\}$, can be effectively reduced to an equivalent quantifier-free formula $\varphi(\mathsf{s}_1^{\alpha_p} x, \ldots, \mathsf{s}_k^{\alpha_p} x)$ (in $\mathscr{L}^{Z}_{\mathsf{kb}^+}$) in which x does not appear and $\mathsf{s}_i^{\alpha_p} x$ $(1 \leq i \leq k)$ is not inside selectors. The same for $t \star x$.

If t is a boundary term or an ordinary term not containing x, then the result follows directly from Lemma 4. Suppose otherwise t must be in the form Lx where L is non-empty block of selectors. We have two cases depending on whether Lx is at the left-hand side or the right-hand side of $\star$.

Case 1. $x \star Lx$**, for** $\star \in \{\prec^{\mathsf{w}}_n, \prec^{\mathsf{p}}_n, \prec^{\mathsf{l}}_n, \leqq^{\mathsf{w}}_n, \leqq^{\mathsf{p}}_n, \leqq^{\mathsf{l}}_n\}$**.** Then $x \star Lx$ simplifies to false. Since L is non-empty and constraints are type-complete, Lx is a proper subterm of x. For $n = 0$, $x \star Lx$ does not hold as $x \neq Lx$. For $n > 1$, $x \star Lx$ violates the subterm property of KBO [1].

Case 2. $Lx \star x$**, for** $\star \in \{\prec^{\mathsf{w}}_n, \prec^{\mathsf{p}}_n, \prec^{\mathsf{l}}_n, \leqq^{\mathsf{w}}_n, \leqq^{\mathsf{p}}_n, \leqq^{\mathsf{l}}_n\}$**.** We assume $n > 1$ as $x \neq Lx$. If $\star$ is $\prec^{\mathsf{w}}_n$ (resp. $\prec^{\mathsf{p}}_n$), Reduction F.1 (resp. F.2) puts x and Lx in separate literals, and hence the problem goes away. Similarly if $\star$ is $\leqq^{\mathsf{w}}_n$ or $\leqq^{\mathsf{p}}_n$. The last two cases are $Lx \prec^{\mathsf{l}}_n x$ and $Lx \leqq^{\mathsf{l}}_n x$, which are only possible when the language contains the unary function f of weight 0, and L must be a non-empty block of s_f (where s_f is the corresponding selector of f). (If L is not of the form $\{\mathsf{s}_f\}^+$, then $\{Lx\}^{\mathsf{w}} < x^{\mathsf{w}}$

and hence $Lx \prec^{\mathsf{l}}_n x$ simplifies to false.) We first consider $Lx \prec^{\mathsf{l}}_n x$, which is of the form

$$\mathsf{s}^i_f(x) \prec^{\mathsf{l}}_n x \tag{27}$$

where $n > 0$, s^i_f denote i time applications of s_f. First note that (27) certainly implies $\mathsf{ls}_f(\mathsf{s}^i_f(x))$. There are two subcases:

A. There is a non-atom term whose weight is x^{w} and type is not f. Let $g(t_1, \ldots, t_j)$ be such a term; i.e., $g \not\equiv f$ and $(g(t_1, \ldots, t_j))^{\mathsf{w}} = x^{\mathsf{w}}$.

1. If $\mathsf{s}^i_f(x)$ is $f^k(a)$ (where $k > 0$ and a is an atom), then x is $f^{i+k}(a)$. Since $(g(t_1, \ldots, t_j))^{\mathsf{w}} = x^{\mathsf{w}} = a^{\mathsf{w}}$, we have either $a \prec^{\mathsf{p}} g(t_1, \ldots, t_j)$ or $g(t_1, \ldots, t_j) \prec^{\mathsf{p}} a$.
   (a) $a \prec^{\mathsf{p}} g(t_1, \ldots, t_j)$. Then

   $$f^k(a) \prec^{\mathsf{l}} f^k(g(t_1, \ldots, t_j)) \prec^{\mathsf{l}} \ldots \prec^{\mathsf{l}} f^k(g(f^l(t_1), \ldots, t_j)) \prec^{\mathsf{l}} \ldots \prec^{\mathsf{l}} f^{i+k}(a).$$

   (b) $g(t_1, \ldots, t_j) \prec^{\mathsf{p}} a$. Then

   $$f^k(a) \prec^{\mathsf{l}} f^{i+k}(g(t_1, \ldots, t_j)) \prec^{\mathsf{l}} \ldots \prec^{\mathsf{l}} f^{i+k}(g(f^l(t_1), \ldots, t_j)) \prec^{\mathsf{l}} \ldots \prec^{\mathsf{l}} f^{i+k}(a).$$

2. If $\mathsf{s}^i_f(x)$ is $f^k(h(t_1, \ldots, t_j))$ (where $k > 0$ and $h \not\equiv f$), then x is $f^{k+i}(h(t_1, \ldots, t_j))$. We have

   $$f^k(h(t_1, \ldots, t_j)) \prec^{\mathsf{l}} \ldots \prec^{\mathsf{l}} f^k(h(f^l(t_1), \ldots, t_j)) \prec^{\mathsf{l}} \ldots \prec^{\mathsf{l}} f^{i+k}(a)$$

In all of cases, there are infinitely many terms in between $\mathsf{s}^i_f(x)$ and x, and hence (27) simplifies to $\mathsf{ls}_f(\mathsf{s}^i_f(x))$.

B. Any term of weight x^{w} consists of only f-terms (f-type terms) or atoms. The condition says that $\mathsf{s}^i_f(x)$ must be in the form $f^k(a)$ (for some atom a and $k > 0$) and then x is in the form $f^{k+i}(a)$. Note that in this situation, $f^k(a) \leqq^{\mathsf{l}}_r f^{k+i}(a)$ where r is i times the number of trees of weight x^{w} in the signature with f excluded. For example, let $\Sigma = \langle f, a, b \rangle$, we have $f(a) \leqq^{\mathsf{l}}_4 f^3(a)$, namely

$$f(a) \leqq^{\mathsf{l}} f(b) \leqq^{\mathsf{l}} f^2(a) \leqq^{\mathsf{l}} f^2(b) \leqq^{\mathsf{l}} f^3(a).$$

It is not hard to see that (27) simplifies to

$$\mathsf{ls}_f(\mathsf{s}^i_f(x)) \ \wedge \ \mathsf{CNT}_m(x^{\mathsf{w}}),$$

where $m = \lceil n/i \rceil - 1$ and $\mathsf{CNT}_m(x^{\mathsf{w}})$ is defined in the language with f excluded.

Similarly we can simplify $Lx \leqq^{\mathsf{l}}_n x$, which has the form

$$\mathsf{s}^i_f(x) \leqq^{\mathsf{l}}_n x \tag{28}$$

where $n > 0$, s^i_f denote i time applications of s_f. As before (28) certainly implies $\mathsf{ls}_f(\mathsf{s}^i_f(x))$. There are two subcases:

A. There is a non-atom term whose weight is x^{w} and type is not f. Since there are infinitely many terms in between $\mathsf{s}^i_f(x)$ and x, (28) simplifies to false.

B. Any term of weight x^{w} consists of only f-terms (f-type terms) or atoms. It is not hard to see that (28) simplifies to

$$\mathsf{Is}_f(\mathsf{s}^i_f(x)) \ \wedge\ \mathsf{CNT}_m(x^{\mathsf{w}}) \ \wedge\ \neg\mathsf{CNT}_{m+1}(x^{\mathsf{w}}) \ \wedge\ i \mid n,$$

where $m = \lceil n/i \rceil - 1$ and $\mathsf{CNT}_m(x^{\mathsf{w}})$ is defined in the language with f excluded.

At last we note that each of preconditions can be expressed in Presburger arithmetic; the condition for **A** is

$$\bigvee_{\alpha \in C \setminus \mathcal{A} \setminus \{f\}} \mathsf{Tree}^{\alpha}(x^{\mathsf{w}}), \tag{29}$$

and the condition for **B** is just the negation of (29). □

B.4 Proof of Lemma 4.

Proof. This lemma states the following: Let x, t be terms and $\star \in \{\prec^{\mathsf{w}}_n, \prec^{\mathsf{p}}_n, \prec^{\mathsf{l}}_n, \leqq^{\mathsf{w}}_n, \leqq^{\mathsf{p}}_n, \leqq^{\mathsf{l}}_n\}$.

1. If x is an ordinary term of type α_p with $\alpha_p = (\mathsf{s}^{\alpha_p}_1, \ldots, \mathsf{s}^{\alpha_p}_k)$ 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 $\varphi(\mathsf{s}^{\alpha_p}_1 x, \ldots, \mathsf{s}^{\alpha_p}_k x)$ in which x dose not occur and $\mathsf{s}^{\alpha_p}_i x$ ($1 \le i \le k$) 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.

As there are many combinations of $x \star t$, the detailed reductions are given in Appendices F-J. Here we only highlight the main idea.

General Idea: Gap Order Delineation.

The general reduction rule is to delineate a gap order to "indecomposable" intervals by inserting boundary terms. For example, suppose both u and v are ordinary terms, $u \prec^{\mathsf{w}}_n v$ ($n \ge 1$) reduces to

$$\bigvee_{\substack{n_1+n_2+n_3=n \\ n_1,n_3 \ge 0, n_2 \ge 1}} u \prec^{\mathsf{pl}}_{n_1} 1^{\mathsf{w}}_{(u^{\mathsf{w}})} \prec^{\mathsf{w}}_{n_2} 0^{\mathsf{w}}_{(v^{\mathsf{w}})} \prec^{\mathsf{pl}}_{n_3} v. \tag{30}$$

Recall that $x \prec^{\mathsf{pl}} y \equiv x \prec^{\mathsf{p}} y \vee x \prec^{\mathsf{l}} y$. Suppose we choose the branch

$$u \prec^{\mathsf{p}}_{n_1} 1^{\mathsf{w}}_{(u^{\mathsf{w}})} \prec^{\mathsf{w}}_{n_2} 0^{\mathsf{w}}_{(v^{\mathsf{w}})} \prec^{\mathsf{p}}_{n_3} v. \tag{31}$$

Assuming u is of type α_p, v is of type α_q, we can further delineate (31) to

$$\bigvee_{\substack{n_1=n_{11}+n_{12} \\ n_3=n_{31}+n_{32}}} u \prec^{\mathsf{l}}_{n_{11}} 1^{\mathsf{p}}_{(u^{\mathsf{w}},p)} \prec^{\mathsf{p}}_{n_{12}} 1^{\mathsf{w}}_{(u^{\mathsf{w}})} \prec^{\mathsf{w}}_{n_2} 0^{\mathsf{w}}_{(v^{\mathsf{w}})} \prec^{\mathsf{p}}_{n_{31}} 0^{\mathsf{p}}_{(v^{\mathsf{w}},q)} \prec^{\mathsf{pl}}_{n_{32}} v. \tag{32}$$

In general suppose we have $u \prec_n^\sharp v$ and insert m boundary terms $t_1, \ldots, t_m$ in between u and v satisfying

$$u \preceq^{\sharp_0} t_1 \preceq^{\sharp_1} \cdots \preceq^{\sharp_{m-1}} t_m \preceq^{\sharp_m} v, \tag{33}$$

where $\sharp_i \in \{\mathsf{w}, \mathsf{p}, \mathsf{l}\}$ $(i \leq m)$ and the maximum of $\{\sharp_i \mid i \leq m\}$ is $\sharp$. Recall that we define $\prec^{\mathsf{c}}$ such that $\mathsf{l} \prec^{\mathsf{c}} \mathsf{p} \prec^{\mathsf{c}} \mathsf{w}$. For example, $max\{\mathsf{w}, \mathsf{p}\} = \mathsf{w}$ and $min\{\mathsf{p}, \mathsf{l}\} = \mathsf{l}$. It is not hard to argue that $u \prec_n^\sharp v$ is equivalent to

$$\bigvee_{\substack{n_0+\ldots+n_m=n \\ n_0,\ldots,n_m \geq 0}} u \prec_{n_0}^{\sharp_0} t_1 \prec_{n_1}^{\sharp_1} t_2 \prec_{n_2}^{\sharp_2} \cdots \prec_{n_{m-1}}^{\sharp_{m-1}} t_m \prec_{n_m}^{\sharp_m} v. \tag{34}$$

Note that in such a delineation, $\prec_0^\sharp$ can be read as $=$. This saves an unnecessary splittings for $\preceq^\sharp$. But it is not essential as the termination of our reductions does not rely on the decrease of gap counts.

Similarly, $u \leqq_n^\sharp v$ can be delineated as

$$\bigvee_{\substack{n_0+\ldots+n_m=n \\ n_0,\ldots,n_m \geq 0}} u \leqq_{n_0}^{\sharp_0} t_1 \leqq_{n_1}^{\sharp_1} t_2 \leqq_{n_2}^{\sharp_2} \cdots \leqq_{n_{m-1}}^{\sharp_{m-1}} t_m \leqq_{n_m}^{\sharp_m} v. \tag{35}$$

As an example for closed relations, we consider $\mathsf{1}_{(m)}^{\mathsf{w}} \prec_n^{\mathsf{w}} \mathsf{0}_{(m')}^{\mathsf{w}}$. If $n = 0$, $\mathsf{1}_{(m)}^{\mathsf{w}} \prec_n^{\mathsf{w}} \mathsf{0}_{(m')}^{\mathsf{w}}$ reduces to

$$\mathsf{1}_{(m)}^{\mathsf{w}} \prec_0^{\mathsf{w}} \mathsf{0}_{(m')}^{\mathsf{w}} \leftrightarrow \mathsf{1}_{(m)}^{\mathsf{w}} = \mathsf{0}_{(m')}^{\mathsf{w}} \ \vee\ \mathsf{1}_{(m)}^{\mathsf{w}} \prec^{\mathsf{w}} \mathsf{0}_{(m')}^{\mathsf{w}}.$$

As was shown in Reduction I.3, $\mathsf{1}_{(m)}^{\mathsf{w}} = \mathsf{0}_{(m')}^{\mathsf{w}}$ reduces to $m = m' \wedge \neg\mathsf{CNT}_1(m)$. If $n = 1$, $\mathsf{1}_{(m)}^{\mathsf{w}} \prec_n^{\mathsf{w}} \mathsf{0}_{(m')}^{\mathsf{w}}$ reduces to $m < m'$. If $n > 1$, $\mathsf{1}_{(m)}^{\mathsf{w}} \prec_n^{\mathsf{w}} \mathsf{0}_{(m')}^{\mathsf{w}}$ reduces to

$$\bigvee_{r=1}^{n-1} (\exists z_1)\ldots(\exists z_r)\Big[m < z_1 < \ldots < z_r < m' \ \wedge \bigvee_{\substack{n_1+\ldots+n_r=n-1 \\ n_1,\ldots,n_r>0}} \bigwedge_{i=1}^{r} \mathsf{CNT}_{n_i-1}(z_i)\Big]. \tag{36}$$

We briefly explain the meaning of (36). Since $\mathsf{1}_{(m)}^{\mathsf{w}}$ is the largest term of weight m and $\mathsf{0}_{(m')}^{\mathsf{w}}$ is the smallest term of weight of m', $\mathsf{1}_{(m)}^{\mathsf{w}} \prec_n^{\mathsf{w}} \mathsf{0}_{(m')}^{\mathsf{w}}$ means that there are at least $n-1$ terms whose weights are in between (but excluding) m and m'. We can assume all terms in between $\mathsf{1}_{(m)}^{\mathsf{w}}$ and $\mathsf{0}_{(m')}^{\mathsf{w}}$ are distributed in r different weights with $1 \leq r \leq n-1$. For fixed r we can also assume the $n-1$ terms are distributed into r levels, which is represented formally as

$$\bigvee_{\substack{n_1+\ldots+n_r=n-1 \\ n_1,\ldots,n_r>0}} \bigwedge_{i=1}^{r} \mathsf{CNT}_{n_i-1}(z_i).$$

Notice that (36) is not in in the syntax of (first-order) logic. However, it is straightforward to translate this abbreviation back to a first-order formula.

Termination

The termination of reductions does not reply on the decrease of gap counts. Rather, a gap order is delineated to "indecomposable" p-intervals. A closed equality literal or a closed gap order literal can be reduced to a Presburger formula (Appendices I and J). Eventually half open gap order literals or open gap order literals involves terms in the same p-interval; i.e., they are in the form of $x \prec_n^{\mathsf{I}} t$ ($n \geq 0$). Suppose x and t are of type α_p with $\alpha_p = (\mathsf{s}_1^{\alpha_p}, \ldots, \mathsf{s}_k^{\alpha_p})$.

Let us first assume t is an ordinary term. Then $x = \alpha_p(\mathsf{s}_1^{\alpha_p} x, \ldots, \mathsf{s}_k^{\alpha_p} x)$, $t = \alpha_p(\mathsf{s}_1^{\alpha_p} t, \ldots, \mathsf{s}_k^{\alpha_p} t)$, and $x \prec_n^{\mathsf{I}} t$ is equivalent to

$$\langle \mathsf{s}_1^{\alpha_p} x, \ldots, \mathsf{s}_k^{\alpha_p} x \rangle \prec_n^{k;\mathsf{kb}} \langle \mathsf{s}_1^{\alpha_p} t, \ldots, \mathsf{s}_k^{\alpha_p} t \rangle,$$

which, by Lemma 5, further reduces to a formula in $\mathscr{L}^{\mathbb{Z}}_{\mathsf{kb}^+}$ in terms of $\mathsf{s}_i^{\alpha_p} x$, $\mathsf{s}_i^{\alpha_p} t$ ($1 \leq i \leq k$).

Now let us assume t is a boundary term, say $t \equiv \mathsf{0}^{\mathsf{w}}_{(m)}$. We can rewrite t as $\alpha_p(\mathsf{s}_1^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m)}, \ldots, \mathsf{s}_k^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m)})$. So $x \prec_n^{\mathsf{I}} t$ is equivalent to

$$\langle \mathsf{s}_1^{\alpha_p} x, \ldots, \mathsf{s}_k^{\alpha_p} x \rangle \prec_n^{k;\mathsf{kb}} \langle \mathsf{s}_1^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m)}, \ldots, \mathsf{s}_k^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m)} \rangle,$$

which, by Lemma 5, further reduces to a formula in $\mathscr{L}^{\mathbb{Z}}_{\mathsf{kb}^+}$ in terms of $\mathsf{s}_i^{\alpha_p} x$, $\mathsf{s}_i^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m)}$ ($1 \leq i \leq k$). By Lemma 2, $\mathsf{s}_1^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m)}, \ldots, \mathsf{s}_k^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m)}$ are simplified to boundary terms not inside selectors, say $\mathsf{0}^{\mathsf{w}}_{(f_1(m))}, \ldots, \mathsf{0}^{\mathsf{w}}_{(f_k(m))}$ where $f_1, \ldots, f_k$ are integer functions definable in Presburger arithmetic. (Note that each $\mathsf{s}_i^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m)}$ must be the smallest (w.r.t. $\prec^{\mathsf{kb}}$) of weight $(\mathsf{s}_i^{\alpha_p} \mathsf{0}^{\mathsf{w}}_{(m)})^{\mathsf{w}}$.)

Similarly if $t \prec_n^{\mathsf{I}} x$ is a half-open or open gap order literal. Note that we never need to reduce a half-open or open equality literal in that a direct quantifier elimination can be carried out (Part 1, Lemma 6).

Well-definedness of boundary terms

For simplicity, reductions listed in Appendices G, H, I, and J implicitly assumes well-definedness of boundary terms. The final reduction result should incorporate predicates WD (Appendix E). For example, suppose $n > 1$, $\mathsf{1}^{\mathsf{w}}_{(m)} \prec_n^{\mathsf{w}} \mathsf{0}^{\mathsf{w}}_{(m')}$ should be equivalent to

$$\begin{aligned} &\mathsf{WD}_{\mathsf{1}^{\mathsf{w}}}(m) \wedge \mathsf{WD}_{\mathsf{0}^{\mathsf{w}}}(m') \wedge (36) \\ &\qquad\qquad \vee \neg\mathsf{WD}_{\mathsf{1}^{\mathsf{w}}}(m) \wedge \mathsf{WD}_{\mathsf{0}^{\mathsf{w}}}(m') \wedge \bot \prec_n^{\mathsf{w}} \mathsf{0}^{\mathsf{w}}_{(m')} \\ &\qquad\qquad \vee \mathsf{WD}_{\mathsf{1}^{\mathsf{w}}}(m) \wedge \neg\mathsf{WD}_{\mathsf{0}^{\mathsf{w}}}(m') \wedge \mathsf{1}^{\mathsf{w}}_{(m)} \prec_n^{\mathsf{w}} \bot \\ &\qquad\qquad\qquad\qquad \vee \neg\mathsf{WD}_{\mathsf{1}^{\mathsf{w}}}(m) \wedge \neg\mathsf{WD}_{\mathsf{0}^{\mathsf{w}}}(m') \wedge \bot \prec_n^{\mathsf{w}} \bot. \end{aligned} \tag{37}$$

Note that we do not need to go through this detour if the parameters in boundary terms comes from weights TA-terms, as predicates WD are valid. For example, for $n > 1$, the final reduction result of $\mathsf{1}^{\mathsf{w}}_{(x^{\mathsf{w}})} \prec^{\mathsf{w}}_{n} \mathsf{0}^{\mathsf{w}}_{(y^{\mathsf{w}})}$ is just

$$\bigvee_{r=1}^{n-1} (\exists z_1)\ldots(\exists z_r)\Big[x^{\mathsf{w}} < z_1 < \ldots < z_r < y^{\mathsf{w}} \;\wedge \bigvee_{\substack{n_1+\ldots+n_r=n-1\\ n_1,\ldots,n_r>0}} \bigwedge_{i=1}^{r} \mathsf{CNT}_{n_i-1}(z_i)\Big]. \tag{38}$$

Also note that we chose $\perp$ as the "sink value" to make boundary functions total. In general, the choice of a sink value affects the truth values of formulas in $\mathscr{L}^{\mathbb{Z}}_{\mathsf{kb}^+}$; i.e., $\mathrm{Th}(\mathfrak{A}^{\mathbb{Z}}_{\mathsf{kb}^+})$ varies as the sink value does. However, $\mathrm{Th}(\mathfrak{A}_{\mathsf{kb}})$ remains the same thanks to the incorporation of well-definedness predicates.

Reduction of Relations Involving Atoms.

There is no separate lists for reductions involving atoms, in that atoms are boundary terms. For example, $\mathsf{0}^{\mathsf{p}}_{(\mathsf{W}(\alpha_p),p)}$ is another name for an atom α_p. So is $\mathsf{1}^{\mathsf{p}}_{(\mathsf{W}(\alpha_p),p)}$.

Presence of the Unary Function of Weight 0.

Up to now we assume that the language does not contain the unary function of weight 0. However, 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 unless the left-hand operand is an atom. 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, $\mathscr{L}^{\mathbb{Z}}_{\mathsf{kb}^+}$ only needs to include lower boundary functions in order to decompose gap orders. All reductions listed in the appendices can be changed accordingly by eliminating upper boundary terms and merging the neighboring gap orders. For example, consider the reduction of $u \prec^{\mathsf{w}}_{n} v$ ($n \geq 1$). Instead of (30), we have two cases.

Case A. If u is not an atom (asserted by tester literals), $u \prec^{\mathsf{w}}_{n} v$ becomes $u \prec^{\mathsf{w}} v$.

Case B. If u is an atom, we can instantiate u with one of atoms as our signature is finite. Suppose $u \prec^{\mathsf{w}}_{n} v$ reduces to $a \prec^{\mathsf{w}}_{n} v$ for some atom a. Then we check if there exists a non-atom term with the weight of $\mathsf{W}(a)$. This is obviously computable due to finiteness of our signature. If so, $a \prec^{\mathsf{w}}_{n} v$ reduces $a \prec^{\mathsf{w}} v$ for the same reason as Case A. Otherwise, $\mathsf{1}^{\mathsf{w}}_{(\mathsf{W}(a))}$ must also be an atom, say b, and so are any terms between a and b. Suppose there are $m-1$ atoms in between a and b. It is easily verified that $a \prec^{\mathsf{w}}_{n} v$ is just $b \prec^{\mathsf{w}}_{n-m} v$ (i.e., $\mathsf{1}^{\mathsf{w}}_{(\mathsf{W}(a))} \prec^{\mathsf{w}}_{n-m} v$), which further reduces to

$$\bigvee_{\substack{n_1+n_2=n-m\\ n_2\geq 0, n_1\geq 1}} \mathsf{1}^{\mathsf{w}}_{(\mathsf{W}(a))} \prec^{\mathsf{w}}_{n_1} \mathsf{0}^{\mathsf{w}}_{(v^{\mathsf{w}})} \prec^{\mathsf{pl}}_{n_2} v. \tag{39}$$

In summary,

1. Appendices I and J list reductions for closed equalities and closed gap order literals.
2. Appendices G and H list reductions for half-open equalities and half-open gap order literals.

3. Appendices F list reductions for open gap literals.

□

B.5 Proof of Lemma 5.

Proof. This lemma states the following: Let U, V be k-tuples of the same weight, and $\star \in \{\prec_n^{k;\mathsf{w}}, \prec_n^{k;\mathsf{p}}, \prec_n^{k;\mathsf{l}}, \leqq_n^{k;\mathsf{w}}, \leqq_n^{k;\mathsf{p}}, \leqq_n^{k;\mathsf{l}}\}$.

1. If $U = \langle u_1, \ldots, u_k \rangle$ is an ordinary tuple, then $U \star V$ ($V \star U$) can be effectively reduced to an equivalent quantifier-free formula $\varphi(u_1, \ldots, u_k)$ (in $\mathscr{L}_{\mathsf{kb}^+}^{\mathsf{Z}}$) in which v_i ($1 \leq i \leq k$) 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.

There are many types of tuple literals, especially the closed tuple literals. However, as tuples are only used in intermediate reduction steps, we only meet a small portion of combinations. The detailed reductions are given in Appendices K-O. Here we only highlight the main points.

Notations

First note that we define tuple relations between tuples of the same weight. So tuple relations are not only parameterized by k, the tuple length, but also parameterized by m, the total weight.

Second note that we define suborders on tuples $\langle u_1, \ldots, u_k \rangle \prec^{k;\sharp} \langle v_1, \ldots, v_k \rangle$ ($\sharp \in \{\mathsf{w}, \mathsf{p}, \mathsf{l}\}$) as

$$u_1 \prec^{\sharp} v_1 \ \vee \ u_1 = v_1 \ \wedge \ \langle u_2, \ldots, u_k \rangle \prec^{k;\mathsf{kb}} \langle v_2, \ldots, v_k \rangle, \tag{40}$$

instead as

$$\exists i (1 \leq i \leq k) \big[u_i \prec^{\sharp} v_i \ \wedge \ \forall j (1 \leq j < i) u_j = v_j \big]. \tag{41}$$

As we shall see soon, this technical choice can give reduction formulas uniform appearance.

Third recall that $\langle u_1, \ldots, u_k \rangle \prec_n^{k;\sharp} \langle v_1, \ldots, v_k \rangle$ is *proper* if $u_1 \prec^{\sharp} v_1$. We claim that improper orders can be decomposed to a Boolean combination of proper orders between k-tuples and proper orders between tuples of shorter length. Consider $U \prec_n^{k;\sharp} V$.

1. U, V are ordinary tuples. Let $U = \langle u_1, \ldots, u_k \rangle$ and $V = \langle v_1, \ldots, v_k \rangle$. So $U \prec_n^{k;\sharp} V$ is equivalent to

$$\Big((u_1 = v_1) \ \wedge \ \langle u_2, \ldots, u_k \rangle \prec_n^{k;\mathsf{kb}} \langle v_2, \ldots, v_k \rangle \Big) \\ \vee \ \Big((u_1 \prec^{\sharp} v_1) \ \wedge \ \langle u_1, \ldots, u_k \rangle \prec_n^{k;\sharp} \langle v_1, \ldots, v_k \rangle \Big).$$

2. U is an ordinary tuple and V is a boundary tuple. It is easily seen that $V = \langle v_1, \ldots, v_k \rangle$ where $v_1, \ldots, v_k$ are boundary terms. For example, $\bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,m)}$ is

$$\langle 1^{\mathsf{w}}_{(m)}, 1^{\mathsf{w}}_{(\mathsf{LCW}_1^{k-1}(sum-m))}, \ldots, 1^{\mathsf{w}}_{(\mathsf{LCW}_{k-1}^{k-1}(sum-m))} \rangle,$$

where $\mathsf{LCW}_i^k(n)$ returns the weight of the i^{th} component of the largest k-tuple (w.r.t. $\prec^{k;\mathsf{kb}}$) of weight n (see Sec. E.13). Then $U \prec_n^{k;\sharp} V$ reduces to

$$\Big((u_1 = v_1) \ \wedge \ \langle u_2, \ldots, u_k \rangle \prec_n^{k;\mathsf{kb}} \langle v_2, \ldots, v_k \rangle\Big) \\ \vee \ \Big((u_1 \prec^{\sharp} v_1) \ \wedge \ \langle u_1, \ldots, u_k \rangle \prec_n^{k;\sharp} V\Big).$$

3. U is a boundary tuple and V is an ordinary tuple. Similar to Case 2.
4. Both U and V are boundary tuples. Similar to Case 2.

In each case, $U \prec_n^{k;\sharp} V$ reduces to a proper tuple order or an order between tuples of shorter length. Iteratively applying the process, an improper k-tuple order can be decomposed to a Boolean combination of a proper k-tuple order and proper orders between tuples of shorter length. Therefore in order to reduce tuple orders to non-tuple order literals (in terms of components of those tuples), it suffices to show *proper reductions*, the reductions for proper orders between tuples of the same weight. Those proper reductions are given in Appendices K-O. Note that there is an implicit side conditions $u_1 \prec^{\sharp} v_1$ in redexes (reduction targets) of all reduction rules.

General Idea: Gap Order Delineation and Tuple Length Reduction.

The general reduction rule is the same as before, namely, using boundary tuples to delineate a tuple order to "indecomposable" intervals. For example, Let $U = \langle u_1, \ldots, u_k \rangle$, $V = \langle v_1, \ldots, v_k \rangle$ be ordinary tuples, $n \geq 1$ and $sum = \sum_{i=1}^k u_i^{\mathsf{w}} = \sum_{i=1}^k v_i^{\mathsf{w}}$. $U \prec_n^{k;\mathsf{w}} V$ reduces to (Reduction K.1)

$$\bigvee_{\substack{n_1+n_2+n_3=n \\ n_1,n_3\geq 0, n_2 \geq 1}} \langle u_1, \ldots, u_k \rangle \prec_{n_1}^{k;\mathsf{pl}} \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,u_1^{\mathsf{w}})} \prec_{n_2}^{k;\mathsf{w}} \bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum,v_1^{\mathsf{w}})} \prec_{n_3}^{k;\mathsf{pl}} \langle v_1, \ldots, v_k \rangle. \tag{42}$$

Now it is clear why we define $\langle u_1, \ldots, u_k \rangle \prec_n^{k;\sharp} \langle v_1, \ldots, v_k \rangle$ by (40). other than by (41). Suppose we did it in the other way. Then in the decomposition (42), we can not have $\langle u_1, \ldots, u_k \rangle \prec_{n_1}^{k;\mathsf{pl}} \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,u_1^{\mathsf{w}})}$, because the order between $\langle u_1, \ldots, u_k \rangle$ and $\bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,u_1^{\mathsf{w}})}$ may be determined by the weights of the i^{th} ($i > 1$) respective components.

Besides gap order delineation, we also need to decompose tuple orders using term order and tuple order in the smaller dimension. For example, consider $\langle u_1, \ldots, u_k \rangle \prec_{n_1}^{k;\mathsf{p}} \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,u_1^{\mathsf{w}})}$, a literal in a branch of (42), which reduces to (Reduction

M.8)

$$\bigvee_{\substack{n_{11}(n_{12}+1)+n_{13}\geq n_1,\\ 0\leq n_{11},n_{12},n_{13}\leq n_1}} u_1 \prec^{\mathsf{p}}_{n_{11}} \mathsf{1}^{\mathsf{w}}_{(u_1^{\mathsf{w}})} \wedge \bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(sum-u_1^{\mathsf{w}})} \prec^{k-1;\mathsf{kb}}_{n_{12}} \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum-u_1^{\mathsf{w}})}$$
$$\wedge \langle u_2,\ldots,u_k\rangle \prec^{k-1;\mathsf{kb}}_{n_{13}} \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum-u_1^{\mathsf{w}})}. \quad (43)$$

We briefly explain the reason behind (43). The tuples in between $\langle u_1,\ldots,u_k\rangle$ and $\bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,u_1^{\mathsf{w}})}$ can be divided into $n_{11}+1$ segments where $n_{11}+1$ is the number of terms in between (and including) u_1 and $\mathsf{1}^{\mathsf{w}}_{(u_1^{\mathsf{w}})}$. Each of the first n_{11} segments contains $n_{12}+1$ k-tuples, where $n_{12}+1$ is the number of $k-1$-tuples of weight $sum-u_1^{\mathsf{w}}$. The last segment contains n_{13} k-tuples because where n_{13} is the number of $k-1$-tuples greater than or equal to $\langle u_2,\ldots,u_k\rangle$ but less than $\bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum-u_1^{\mathsf{w}})}$. We require $n_{11}(n_{12}+1)+n_{13}\geq n_1$ and $0\leq n_{11},n_{12},n_{13}\leq n_1$.

As an example for closed relations, $\bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n} \bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum,m')}$ reduces to (Reduction O.25)

$$\bigvee_{r=1}^{n-1}\Big[m < \mathsf{CW}^k_1(sum,m,m') < \ldots < \mathsf{CW}^k_r(sum,m,m') < m' \wedge$$
$$\bigvee_{\substack{\sum_{i=1}^r n_i=n-1\\ n_i\geq 1}} \bigwedge_{i=1}^{r} \bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum,\mathsf{CW}^k_i(sum,m,m'))} \prec^{k;\mathsf{w}}_{n_i-1} \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,\mathsf{CW}^k_i(sum,m,m'))}\Big]. \quad (44)$$

We explain the reason for (44). Since $\bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,m)}$ is the largest k-tuple of weight sum with the first component having weight m and $\mathsf{0}^{\mathsf{w}}_{(m')}$ is the smallest tuple of weight of sum with the first component having weight m' $\bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n} \bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum,m')}$ means that there are at least $n-1$ k-tuples of weight sum for which the weights of the first components are greater than m and smaller than m'. We can assume all k-tuples in between $\bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,m)}$ and $\bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum,m')}$ are distributed in r different weights with $1\leq r\leq n-1$. For a fixed r we can also assume the $n-1$ k-tuples are distributed into r levels, which is represented formally as

$$\bigvee_{\substack{\sum_{i=1}^r n_i=n-1\\ n_i\geq 1}} \bigwedge_{i=1}^{r} \bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum,\mathsf{CW}^k_i(sum,m,m'))} \prec^{k;\mathsf{w}}_{n_i-1} \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,\mathsf{CW}^k_i(sum,m,m'))}.$$

Note that $\mathsf{CW}^k_i(n,r,r')$ returns the i^{th} smallest integer in (r,r') which is the weight of a component in a k-tuple having weight n, and hence

$$\bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum,\mathsf{CW}^k_i(sum,m,m'))} \prec^{k;\mathsf{w}}_{n_i-1} \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,\mathsf{CW}^k_i(sum,m,m'))}$$

express that there are n_i k-tuples of weight sum with the first component having weight $\mathsf{CW}^k_i(sum,m,m')$. We require $\sum_{i=1}^r n_i = n-1$ and $n_i\geq 1$ for $1\leq i\leq r$.

Termination

As before the termination of reductions does not reply on the decrease of gap counts. Rather, a tuple gap order is first delineated to "indecomposable" intervals which is turn reduced to gap order between tuples of shorter length. So eventually a tuple order is reduced to a formula in $\mathscr{L}^{\mathbb{Z}}_{\mathsf{kb}^+}$. In particular, a closed tuple gap order or a closed tuple equality literals can be reduced to a Presburger formula. Iteratively applying the reduction, eventually only term order $\prec^{\sharp}_{n}$ ($\sharp \in \{\mathsf{w}, \mathsf{p}, \mathsf{l}\}$) appears and the reduction terminates.

Well-definedness of Boundary Tuples

Reductions listed in Appendices G, H, I, and J implicitly assume well-definedness of boundary tuples. This is safe as parameters of boundary tuples comes from weights of well-defined TA-terms.

Reduction of Literals Involving $\bar{\mathbf{0}}^{k;\mathsf{kb}}_{(sum)}$ or $\bar{\mathbf{1}}^{k;\mathsf{kb}}_{(sum)}$.

There is no separate lists for reductions of literals involving $\bar{\mathbf{0}}^{k;\mathsf{kb}}_{(sum)}$ or $\bar{\mathbf{1}}^{k;\mathsf{kb}}_{(sum)}$. as they can be expressed, respectively, by

$$\bar{\mathbf{0}}^{k;\mathsf{kb}}_{(sum)} = \bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum,\mathsf{MinCW}^k(sum))}, \qquad \bar{\mathbf{1}}^{k;\mathsf{kb}}_{(sum)} = \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,\mathsf{MaxCW}^k(sum))}.$$

Here $\mathsf{MinCW}^k(sum))$ (resp. $\mathsf{MaxCW}^k(sum))$ abbreviates $\mathsf{MaxCW}^k(sum, 0, sum+1)$ (resp. $\mathsf{MinCW}^k(sum, 0, sum+1))$, the largest integer (resp. the smallest integer) in $(0, sum+1)$ which is the weight of a component in a k-tuple having weight sum.

The Number of Open Gap Order Literals.

As we mentioned before, tuple orders are reduced to term orders. The reduction does not increase the number of open gap order literals. The only reduction generating open gap order literals is reduction of open tuple order $\langle u_1, \ldots, u_k\rangle \prec^{k;\mathsf{l}}_{n} \langle v_1, \ldots, v_k\rangle$ (Reduction K.3), copied below

$$\begin{aligned}\bigvee_{\substack{n_1+n_2+n_3=n,\\ n_1,n_2\geq 0, n_3\geq 1}} \Big[\langle u_2, \ldots, u_k\rangle \prec^{k-1;\mathsf{kb}}_{n_1} \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(rem)} \wedge \bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(rem)} \prec^{k-1;\mathsf{kb}}_{n_2} \langle v_2, \ldots, v_k\rangle \\ \wedge \bigvee_{\substack{(m_1+1)(m_2-1)\geq(n_3-1)\\ m_1\geq 0, m_2>0, m_1,m_2<n_3}} \Big(\bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(rem)} \prec^{k-1;\mathsf{kb}}_{m_1} \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(rem)} \wedge u_1 \prec^{\mathsf{l}}_{m_2} v_1\Big)\Big],\end{aligned} \tag{45}$$

where $rem = \sum_{i=2}^{k} u_i^{\mathsf{w}} = \sum_{i=2}^{k} v_i^{\mathsf{w}}$. It is not hard to argue that in the final result of (45), each conjunction only contains one open gap order literals. As the reduction is triggered by a open gap order literal, it does not add more open gap order literals. Note that though $u_1 \prec^{\sharp} v_1$ is an implicit term gap order in redexes, it does not included in reducts (results of reductions). In particular, the implicit condition $u_1 \prec^{\mathsf{l}} v_1$ for $\langle u_1, \ldots, u_k\rangle \prec^{k;\mathsf{l}}_{n} \langle v_1, \ldots, v_k\rangle$ is replaced by more precise relation $u_1 \prec^{\mathsf{l}}_{m_2} v_1$.

In summary,

1. Appendices N and O, respectively, list reductions for closed equality literals and closed tuple order literals.
2. Appendices L and M, respectively, list reductions for half-open equality literals and half-open tuple order literals.
3. Appendices K list reductions for open tuple order literals. □

B.6 Proof of Lemma 7.

Proof. The lemma states the following claim: Let z be an integer variable, $C_{\mathsf{kb}^+}(z)$ a conjunction of literals in $\mathscr{L}_{\mathsf{kb}^+}$ where z occurs inside boundary terms, and $\varphi_{\mathbb{Z}}(z)$ a Presburger formula. Then $(\exists z : \mathbb{Z})[C_{\mathsf{kb}^+}(z) \wedge \varphi_{\mathbb{Z}}(z)]$ can be effectively reduced to $C'_{\mathsf{kb}^+} \wedge \varphi'_{\mathbb{Z}}$ where no z occurs and C'_{kb^+} is quantifier-free.

Let z, $C_{\mathsf{kb}^+}(z)$ and $\varphi_{\mathbb{Z}}(z)$ as stated above. Let us assume z has n occurrences in $C_{\mathsf{kb}^+}(z)$, namely

$$f_1(z),\ \ldots,\ f_n(z),$$

where f_i $(1 \leq i \leq n)$ are arbitrary integer function definable in Presburger arithmetic, and $f_i(z)$ $(1 \leq i \leq n)$ occur inside boundary terms. Note that f_i's are introduced during reductions in Appendices F-O, and they are built up using usual integer functions as well as those defined in Appendix E.

For each $1 \leq i \leq n$, we do simplification as follows. Let us assume $f_i(z)$ occurs inside a boundary term t. Without loss of generality, let assume $f_i(z)$ is the outmost integer function, i.e., t has one of the following forms:

$$\mathsf{0}^{\mathsf{w}}_{(f_i(z))}, \qquad \mathsf{0}^{\mathsf{p}}_{(f_i(z),\_)}, \qquad \mathsf{1}^{\mathsf{w}}_{(f_i(z))}, \qquad \mathsf{1}^{\mathsf{p}}_{(f_i(z),\_)}$$

Consider the occurrences of t in $C_{\mathsf{kb}^+}(z)$.

Case 1 t **occurs in literals of the forms**

$$t <^{\sharp}_{n} t', \qquad t \leqq^{\sharp}_{n} t', \qquad t' <^{\sharp}_{n} t, \qquad t' \leqq^{\sharp}_{n} t,$$

where $\sharp \in \{\mathsf{w}, \mathsf{p}, \mathsf{l}\}$ **and** t' **is also a boundary term.** By Lemma 4, we can remove all such closed literals from $C_{\mathsf{kb}^+}(z)$ and add the corresponding equivalent Presburger formulas to $\varphi_{\mathbb{Z}}(z)$.

Case 2 t **occurs in literals of the forms**

$$t <^{*}_{n} t', \qquad t \leqq^{*}_{n} t', \qquad t' <^{*}_{n} t, \qquad t' \leqq^{*}_{n} t,$$

where $* \in \{\mathsf{p}, \mathsf{l}\}$ **and** t' **is an ordinary term.** Then we replace every occurrence of $f_i(z)$ in $C_{\mathsf{kb}^+}(z)$ by $(t')^{\mathsf{w}}$ and add $f_i(z) = (t')^{\mathsf{w}}$ to $\varphi_Z(z)$, and set

$$\begin{aligned} C_{\mathsf{kb}^+}(z) &:= C_{\mathsf{kb}^+}(z)\Big[f_i(z)/(t')^{\mathsf{w}}\Big], \\ \varphi_{\mathbb{Z}}(z) &:= \varphi_{\mathbb{Z}}(z) \wedge f_i(z) = (t')^{\mathsf{w}}. \end{aligned}$$

Case 3 t **occurs in literals of the forms**

$$t <^{\mathsf{w}}_{n} t', \qquad t \leqq^{\mathsf{w}}_{n} t', \qquad t' <^{\mathsf{w}}_{n} t, \qquad t' \leqq^{\mathsf{w}}_{n} t,$$

where t' is an ordinary term. In this case, we introduce new boundary terms to delineate those gap orders. For example, $t \prec^{\mathsf{w}}_{n} t'$ is replaced by

$$\bigvee_{\substack{n_1+n_2=n \\ n_1\geq 0, n_2\geq 0}} t \prec^{\mathsf{w}}_{n_1} \mathsf{0}^{\mathsf{w}}_{((t')^{\mathsf{w}})} \prec^{\mathsf{pl}}_{n_2} t'$$

Similar transformations apply to $t \leqq^{\mathsf{w}}_{n} t'$, $t' \prec^{\mathsf{w}}_{n} t$ and $t' \leqq^{\mathsf{w}}_{n} t$. By the transformation we reduce this case to Case 1.

In this way we can remove all $f_i(z)$ from C_{kb^+}, obtaining

$$(\exists z\!:\!\mathbb{Z}) \left[C'_{\mathsf{kb}^+} \wedge \varphi'_{\mathbb{Z}}(z)\right],$$

where C'_{kb^+} does not contain z. Now we can move C'_{kb^+} out of $\exists z$, obtaining

$$C'_{\mathsf{kb}^+} \wedge (\exists z\!:\!\mathbb{Z})\, \varphi'_{\mathbb{Z}}(z),$$

which is our desired result. □

B.7 Proof of Lemma 6.

Proof. This lemma states the following: Let x be a term variable, $C_{\mathsf{kb}^+}(x)$ a conjunction of literals in $\mathscr{L}_{\mathsf{kb}^+}$ in which all term occurrences of x have depth 0, and $\varphi_{\mathbb{Z}}(x)$ a Presburger formula in which x occurs inside the weight function. Then $(\exists x\!:\!\mathsf{TA})[C_{\mathsf{kb}^+}(x) \wedge \varphi_{\mathbb{Z}}(x)]$ can be effectively reduced to $C'_{\mathsf{kb}^+} \wedge \varphi'_{\mathbb{Z}}$ in which x does not occur and C'_{kb^+} is quantifier-free.

Note that in the antecedent of the lemma, we only require that all **term occurrences** of x have depth 0 and x can have arbitrarily many integer occurrences of the form $(Lx)^{\mathsf{w}}$ where L is a (possibly empty) selector sequence.

We divide the proof into two parts.

Part 1. $C_{\mathsf{kb}^+}(x)$ contains $x = t$ where t may contain x.

We rewrite $(\exists x\!:\!\mathsf{TA})[C_{\mathsf{kb}^+}(x) \wedge \varphi_{\mathbb{Z}}(x)]$ as

$$(\exists x\!:\!\mathsf{TA}) \left[\, x = t \ \wedge \ \varphi(x)\right], \tag{46}$$

where $\varphi(x)$ is an arbitrary conjunction of literals in $\mathscr{L}^{\mathbb{Z}}_{\mathsf{kb}^+}$.

If t does not contain x, then (46) simplifies to $\varphi(t)$. Otherwise, we have three cases to consider.

Case 1. $t \equiv x$. Then $(\exists x)[x = x \wedge \varphi(x)]$ simplifies to $(\exists x)\varphi(x)$.

Case 2. $t \equiv Lx$ **where** L **is a non-empty selector block.** Then $(\exists x)(x = Lx \wedge \varphi(x))$ simplifies to false. Note that here we assume that terms like Lx are proper (see Def. 16)

Case 3. t **is a boundary term in which** $(L_i x)^{\mathsf{w}}$ **($i \leq n$) occur (as parts of an integer function).** In this case (46) has the form

$$(\exists x\!:\!\mathsf{TA}) \left[\, x = t[f((L_0x)^{\mathsf{w}}, \ldots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}})] \ \wedge \ \varphi(x)\,\right], \tag{47}$$

where $f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},x^{\mathsf{w}})$ denotes the outmost Presburger function containing $(L_ix)^{\mathsf{w}}$ ($i \le n$) and x^{w}. We assume $t[f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},x^{\mathsf{w}})]$ is a well-defined boundary term (see Sec. E.15). Otherwise $t = \bot$ and hence (46) reduces to $\varphi(\bot)$. Without loss of generality, we assume L_ix enumerate all selector terms containing x in (47) and they form a tree (with x being the root, L_ix including all leaves). In addition, in this tree sibling nodes are coexistent, called *sibling-complete*. For example, in $\mathfrak{A}_{\mathsf{list}}$ if $\mathsf{car}(x)$ appear, we can assume that $\mathsf{cdr}(x)$ also appear. The induced tree is $\mathsf{cons}(\mathsf{car}(x),\mathsf{cdr}(x))$ where x is the root, $\mathsf{car}(x)$, $\mathsf{cdr}(x)$ are two leaves. Note that L_ix also include intermediate nodes in the induced tree. In general, if we need $L'x$ to appear, we can just set

$$f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},(L'x)^{\mathsf{w}},x^{\mathsf{w}}) := f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},x^{\mathsf{w}}) + (L'x)^{\mathsf{w}} - (L'x)^{\mathsf{w}}.$$

It is easily seen that (47) is equivalent to

$$(\exists x\!:\!\mathsf{TA})\Big[\, x = t[f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},x^{\mathsf{w}})] \;\wedge\; \varphi(t[f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},x^{\mathsf{w}})])\,\Big]. \tag{48}$$

Note that in (48) x only has one term occurrence, namely, as the left-hand side of

$$x = t[f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},x^{\mathsf{w}})]. \tag{49}$$

We claim that (48) is equivalent to

$$(\exists (L_0x)^{\mathsf{w}},\ldots(L_nx)^{\mathsf{w}},x^{\mathsf{w}}\!:\!\mathbb{Z})\Big[\,\varphi(t[f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},x^{\mathsf{w}})]) \wedge \varDelta\,\Big], \tag{50}$$

where $\varDelta$ is the constraint to describe the weight relations in the induced tree. More precisely, it satisfies the following conditions.

1. If L_ix is of type α_p and has immediate children $\mathsf{s}_1^{\alpha_p}L_ix,\ldots,\mathsf{s}_k^{\alpha_p}L_ix$, then $\varDelta$ includes

$$(L_ix)^{\mathsf{w}} = \Big(\sum_{j=1}^{k}(\mathsf{s}_j^{\alpha_p}L_ix)^{\mathsf{w}}\Big) + \mathsf{W}(\alpha_p).$$

   In particular, it includes

$$x^{\mathsf{w}} = \sum_{i\in I\subseteq\{0..n\}}(L_ix)^{\mathsf{w}} + c.$$

   where I is the index of all leaf nodes and c is the sum of the weight of intermediate nodes of the tree with root x.
2. For $i \in I$, $\varDelta$ includes $\mathsf{Tree}((L_ix)^{\mathsf{w}})$; if L_ix is α-typed, $\varDelta$ includes $\mathsf{Tree}^{\alpha}((L_ix)^{\mathsf{w}})$. These assertions constrains each leaf to have a well-defined weight.

3. Δ also includes constraints that require that $L_0x, \ldots, L_nx, x$ form a boundary term. For example, suppose $t \equiv 0^{\mathsf{w}}_{(f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},x^{\mathsf{w}}))}$. Then each node L_ix must be the smallest (w.r.t. $\prec^{\mathsf{kb}}$) of weight $(L_ix)^{\mathsf{w}}$. If L_ix is of type α_p, then there exists no p' such that $p < p' \leq |\Sigma|$ and $\mathsf{Tree}^{\alpha_{p'}}((L_ix)^{\mathsf{w}})$ holds. This constraint is formally expressed by

$$\bigwedge_{p<p'\leq|\Sigma|} \neg\mathsf{Tree}^{\alpha_{p'}}((L_ix)^{\mathsf{w}}).$$

A similar constraint is added for x. Besides that, if L_ix has k siblings, namely, in the order from left to right, $L_{i_1}x, \ldots, L_{i_k}x$ (where $L_{i_j} \equiv \mathsf{s}_j^{\alpha_p}$ for $1 \leq j \leq k$), we require that $\langle L_{i_1}x, \ldots, L_{i_k}\rangle$ be the smallest k-tuple (w.r.t. $\prec^{k;\mathsf{kb}}$) of weight $n = \sum_{j=1}^{k}(L_{i_j}x)^{\mathsf{w}}$. This constraint is formally expressed by

$$\bigwedge_{j=1}^{k}(L_{i_j}x)^{\mathsf{w}} = \mathsf{SCW}_j^k(n),$$

where $\mathsf{SCW}_j^k(n)$ returns the weight of the j^{th} component of the smallest k-tuple (w.r.t. $\prec^{k;\mathsf{kb}}$) of weight n (see Sec. E.11). Similarly we obtain constraints for

$$t \equiv 0^{\mathsf{p}}_{(f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},x^{\mathsf{w}}),\_)}, \quad t \equiv 1^{\mathsf{w}}_{(f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},x^{\mathsf{w}}))}, \quad t \equiv 1^{\mathsf{p}}_{(f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},x^{\mathsf{w}}),\_)}.$$

Notice that here $(L_ix)^{\mathsf{w}}$, x^{w} serve as purely syntactical variables. A variable renaming can bring (50) to

$$(\exists z_0, \ldots z_n, z\!:\!\mathbb{Z})\Big[\,\varphi'(t[f(z0, \ldots, z_n, z)]) \;\wedge\; \Delta(z_0, \ldots, z_n, z)\,\Big]. \tag{51}$$

We show the equivalence between (48) and (50). First it is obvious that (48) implies (50) because Δ describes weight constraints on the boundary term tree rooted at x. To show that (50) implies (48), it suffices to show there exists x such that (49) holds. Again because of Δ, $(L_0x)^{\mathsf{w}}, \ldots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}}$ satisfies weight constraints of a boundary term (tree) and $f((L_0x)^{\mathsf{w}}, \ldots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}})$ makes t well-defined. Therefore $(L_0x)^{\mathsf{w}}, \ldots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}}$ can be "realized" by a TA-term x which is a boundary term. Intuitively, when a term x occurs inside the weight function, its term properties is deprived.
We like to point out three things about this elimination by substitution. First we can proceed with this elimination procedure even if x has other term occurrences with depths great than 1. Second the substitution from (47) to (48) may put a boundary term inside a selector term. In fact, it is only the substitution that makes a boundary term appear inside a selector. However, Lemma 2 eliminates this superficial complication. Third, the substitution obviously does not increase the number of open order literals.

Part 2. $C_{\mathsf{kb}^+}(x)$ does not contain equalities like $x = t$.

Since x is the only term occurrence of x in C_{kb^+}, we can move other constraints not containing x out of $(\exists x : \mathsf{TA})$. So we can assume that C_{kb^+} is in the form

$$\bigwedge_{i=1}^{l} u_i \rhd_i x \;\wedge\; \bigwedge_{j=1}^{m} x \rhd_{l+j} v_j, \tag{52}$$

where $\rhd_i$ ($1 \leq i \leq l + m$) are gap orders. By Lemma 8, C_{kb^+} is equivalent to a disjunction of delineated gap order completions, each of which contains no more open order literals than $C_{\mathsf{kb}^+}(x)$ does. Now we assume $C_{\mathsf{kb}^+}(x)$ is a DGOC, denoted by $\mathsf{DGOC}(x, (L_0x)^{\mathsf{w}}, \ldots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}})$ where $(L_0x)^{\mathsf{w}}, \ldots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}}$ enumerate all integer occurrence of x in $C_{\mathsf{kb}^+}(x)$ and $\varphi_{\mathbb{Z}}(x)$. Note that $(L_0x)^{\mathsf{w}}, \ldots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}}$ could be parts of integer functions which are in turn inside boundary terms. Let $\{x\}^{\mathsf{w}}$ abbreviate $(L_0x)^{\mathsf{w}}, \ldots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}}$. As in Part 1, we also assume $L_0x, \ldots, L_nx, x$ enumerate all selector terms containing x and they form a sibling-complete tree (with x being the root and L_ix including all leaves).

Without loss of generality, we assume that $\mathsf{DGOC}(x, \{x\}^{\mathsf{w}})$ has the following form:

$$t_1 \;\rhd_1\; \ldots \;\rhd_{i-2}\; t_{i-1} \;\rhd_{i-1}\; x \;\rhd_i\; t_{i+1} \;\rhd_{i+1}\; \ldots \;\rhd_{n-1}\; t_n.$$

Let $\mathsf{Head}(\{x\}^{\mathsf{w}})$ (resp. $\mathsf{Tail}(\{x\}^{\mathsf{w}})$) denote the linear order before x (resp. after x) in the above sequence. We can decompose $\mathsf{DGOC}(x, \{x\}^{\mathsf{w}})$ as follows.

$$\underbrace{t_1 \;\rhd_1\; \ldots \;\rhd_{i-2}\; t_{i-1}}_{\mathsf{Head}(x^{\mathsf{w}})} \;\rhd_{i-1}\; x \;\rhd_i\; \underbrace{t_{i+1} \;\rhd_{i+1}\; \ldots \;\rhd_{n-1}\; t_n}_{\mathsf{Tail}(x^{\mathsf{w}})}.$$

Since $\mathsf{DGOC}(x, \{x\}^{\mathsf{w}})$ is a DGOC, we can further assume $\rhd_{i-1}$ is $\prec^{\sharp}_{n_1}$ and $\rhd_i$ is $\prec^{\sharp}_{n_2}$ where $\sharp \in \{\mathsf{pl}, \mathsf{l}\}$. So $\mathsf{DGOC}(x, \{x\}^{\mathsf{w}})$ is in the form

$$\mathsf{Head}(\{x\}^{\mathsf{w}}) \prec^{\sharp}_{n_1} x \prec^{\sharp}_{n_2} \mathsf{Tail}(\{x\}^{\mathsf{w}}).$$

We write $\mathsf{Head}(\{x\}^{\mathsf{w}}) \prec^{\sharp}_{n_1+n_2} \mathsf{Tail}(\{x\}^{\mathsf{w}})$ to denote the ordering obtained from $\mathsf{DGOC}(x, \{x\}^{\mathsf{w}})$ by removing the *term occurrence* of x and accordingly joining two gap orders before and after such occurrence, namely,

$$\underbrace{t_1 \;\rhd_1\; \ldots \;\rhd_{i-2}\; t_{i-1}}_{\mathsf{Head}(\{x\}^{\mathsf{w}})} \;\prec^{\sharp}_{n_1+n_2}\; \underbrace{t_{i+1} \;\rhd_{i+1}\; \ldots \;\rhd_{n-1}\; t_n}_{\mathsf{Tail}(\{x\}^{\mathsf{w}})}.$$

Let us first assume that $\sharp \equiv \mathsf{l}$; the case that $\sharp \equiv \mathsf{pl}$ is similar. We further assume all terms in the p-interval where x is have type α_p. Recall that $\mathsf{DGOC}(x, \{x\}^{\mathsf{w}})$ is type-complete and tester literals are not listed for simplicity. We claim that

$$(\exists x : \mathsf{TA}) \left[\mathsf{Head}(\{x\}^{\mathsf{w}}) \prec^{\mathsf{l}}_{n_1} x \prec^{\mathsf{l}}_{n_2} \mathsf{Tail}(\{x\}^{\mathsf{w}}) \;\wedge\; \varphi_{\mathbb{Z}}(\{x\}^{\mathsf{w}})\right] \tag{53}$$

is equivalent to

$$(\exists\{x\}^{\mathsf{w}}\!:\!\mathbb{Z})\left[\mathsf{Head}(\{x\}^{\mathsf{w}}) \prec^{\mathsf{l}}_{n_1+n_2} \mathsf{Tail}(\{x\}^{\mathsf{w}}) \ \wedge\ \Delta(\{x\}^{\mathsf{w}}) \ \wedge\ \varphi_{\mathbb{Z}}(\{x\}^{\mathsf{w}})\right], \tag{54}$$

where as in Part 1, $\Delta(\{x\}^{\mathsf{w}})$ are weight constraints describing the tree structure formed by $L_0x, \ldots, L_nx, x$. Note that $\{x\}^{\mathsf{w}}$ now are purely syntactical integer variable.

We show the equivalence between (53) and (54). First it is obvious that (53) implies (54). On the other hand, (54) implies

$$(\exists\{x\}^{\mathsf{w}}\!:\!\mathbb{Z})(\exists x\!:\!\mathsf{TA})\left[\mathsf{Is}_{\alpha_p}(x) \ \wedge\ \Delta(\{x\}^{\mathsf{w}}) \ \wedge\ \varphi_{\mathbb{Z}}(\{x\}^{\mathsf{w}}) \wedge \right.$$
$$\left. \mathsf{Head}(\{x\}^{\mathsf{w}}) \prec^{\mathsf{l}}_{n_1} x \prec^{\mathsf{l}}_{n_2} \mathsf{Tail}(\{x\}^{\mathsf{w}})\right]. \tag{55}$$

which is equivalent to

$$(\exists x\!:\!\mathsf{TA})(\exists\{x\}^{\mathsf{w}}\!:\!\mathbb{Z})\left[\mathsf{Is}_{\alpha_p}(x) \ \wedge\ \Delta(\{x\}^{\mathsf{w}}) \ \wedge\ \varphi_{\mathbb{Z}}(\{x\}^{\mathsf{w}}) \wedge \right.$$
$$\left. \mathsf{Head}(\{x\}^{\mathsf{w}}) \prec^{\mathsf{l}}_{n_1} x \prec^{\mathsf{l}}_{n_2} \mathsf{Tail}(\{x\}^{\mathsf{w}})\right]. \tag{56}$$

which obviously implies (53). Therefore (53) is equivalent to (54). By variable renaming we rewrite (54) as

$$(\exists\bar{z}\!:\!\mathbb{Z})\left[\mathsf{Head}(\bar{z}) \prec^{\mathsf{l}}_{n_1+n_2} \mathsf{Tail}(\bar{z}) \ \wedge\ \Delta(\bar{z}) \ \wedge\ \varphi_{\mathbb{Z}}(\bar{z})\right]. \tag{57}$$

Now by Lemma 7 we can eliminate $\exists\bar{z}$ from (57).

□

C Proofs of Theorems

We devote this section to prove the correctness of the elimination algorithm.

Lemma 9. *Each transformation step in the elimination procedure is sound, i.e., it preserves equivalence.*

Proof. Preservation of equivalence has been established by lemmas in Appendix B. More precisely, Lemma 7 justifies transformation equivalence in Algorithm 1; Lemmas 4 and 6 justifies subprocedures [2] and [2], respectively, in Algorithm 2. Axioms in Proposition 1 justifies basic simplification steps in [2]. □

Lemma 10. *The elimination procedure terminates.*

Proof. It is clear that Alg. 1 terminates. The termination of Alg. 2 is a bit involved. The difficulty comes from Step [2]. It is easily seen that each run of Step [2] terminates as the decomposition is done in depth-first manner (Step [2]), which eventually produces a variable with depth 0. However this process in general introduces new quantified variables (Step [2]). In addition the DEPTH-REDUCTION subprocedure (in Step [2]) may increase the depth of the term on the other side of an order relation. By Lemmas 3, 4 and 5, the increase of depths of terms only happens to the reductions of open order literals. Checking all reductions in Appendices F-O, we see this actually happens in Reduction F.3. Moreover this reduction can also increase the number of equalities and the substitution in elimination of equality (Part 1, Lemma 6), may also increase depths of terms if the equality to be eliminated is open. But the source of the depth increase is also the existence of open order literals. So to show the termination of the whole procedure it suffices to show the decrease of the number of open order literals.

First we notice that negation elimination (Sec. D) and substitution in eliminating equalities (Part 1 in Lemma. 6) do not introduce new open order literals (as conjuncts) into the resulting matrix. Second it is not hard to argue that Reduction F.3, which comes from a reduction of an open order literal, replaces the old open order literal by a new one in each resulting conjunction. Third when the depth of a variable x is reduced to 0, we meet conjunction of the form (52), denoted by $C_{\mathsf{kb}^+}(x)$. We assume $C_{\mathsf{kb}^+}(x)$ contains at least one open order literals. By Part 2 in Lemma 6 and Lemma 8 the corresponding $\mathsf{DGOC}(x, \{x\}^{\mathsf{w}})$ has no more open order literals than $C_{\mathsf{kb}^+}(x)$ does. Moreover, let $\mathsf{DGOC}(x, \{x\}^{\mathsf{w}})$ be

$$\underbrace{t_1 \ \rhd_1 \ \dots \ \rhd_{i-2} \ t_{i-1}}_{\mathsf{Head}(x^{\mathsf{w}})} \rhd_{i-1} x \rhd_i \underbrace{t_{i+1} \ \rhd_{i+1} \ \dots \ \rhd_{n-1} \ t_n}_{\mathsf{Tail}(x^{\mathsf{w}})}. \tag{58}$$

The elimination of x will give $\mathsf{Head}(\{x\}^{\mathsf{w}}) \rhd_\star \mathsf{Tail}(\{x\}^{\mathsf{w}})$, the ordering obtained by removing the *term occurrence* of x. It is not hard to argue that if either t_{i-1} or t_{i+1} is an ordinary term, $\mathsf{Head}(\{x\}^{\mathsf{w}}) \rhd_\star \mathsf{Tail}(\{x\}^{\mathsf{w}})$ has one less open order literals than $\mathsf{DGOC}(x, \{x\}^{\mathsf{w}})$. If both t_{i-1} and t_{i+1} are boundary terms, then by Lemma 8 $\mathsf{DGOC}(x, \{x\}^{\mathsf{w}})$ has one fewer open order literals than $C_{\mathsf{kb}^+}(x)$ providing $C_{\mathsf{kb}^+}(x)$ contains at least one open order literals. Note that the delineation process could assert x to be a boundary term. For example, $\rhd_{i-1}$ is $\prec_0^{\mathsf{l}}$ and t_{i-1} is a boundary term. Recall $t_{i-1} \prec^{\mathsf{l}}_{0} x$ is equivalent to $t_{i-1} \prec^{\mathsf{l}} x \vee x = t_{i-1}$. If we choose $x = t_{i-1}$, then we proceed as Part 1 in Lemma. 6. As we mentioned before, this process does not add more open order literals and in case that x occurs in at least one open order literals in $C_{\mathsf{kb}^+}(x)$, the substitution will reduce the number of open order literals.

We showed that the quantifier elimination of $(\exists x : \mathsf{TA})$ where x involves in an open order literal will result in a quantifier-free formula which contains fewer open order literals in each conjunction. When all open order literals are gone, the depth reduction procedure will not increase the depths of terms on the other side of an order relation. Note that the depth reduction can still generate more equalities. However, those generated equalities are half-open equalities (i.e., one side of an equality is a boundary term), and hence the subsequent substitutions (in elimination of equalities) will not increase the depths of terms either. Eventually all term occurrences of TA-terms have depth 0. The termination follows. □

C.1 Proof of Theorem 1.

Proof. By Lemmas 9 and 10. □

C.2 A Variant Elimination Procedure.

We present a variant elimination procedure. Other than guessing a delineated gap order completion, we only need to guess a smallest upper bound and a largest lower bound. We assume no $x = t$ occurs inside $(\exists x : \mathsf{TA})$ as this case is identical to Part 1 in Lemma 6.

Lemma 11. *Formulas of the form*

$$(\exists x : \mathsf{TA}) \Big[\bigwedge_{i=1}^{m} u_i \trianglerighteq^{\sharp_i}_{p_i} x \;\wedge\; \bigwedge_{j=1}^{n} x \trianglerighteq^{*_j}_{q_j} v_j \Big], \tag{59}$$

where $\trianglerighteq$ stands for $\prec$ or $\leqq$, $\sharp_i, *_j \in \{\mathsf{w}, \mathsf{p}, \mathsf{l}\}$, and $p_i, q_j \geq 1$, can be effectively reduced to a quantifier-free formula which has fewer open literals than the original formula.

Proof. We first introduce the following notations. For a term t, let $t + n$ denote the n^{th} term after t (w.r.t. $\prec^{\mathsf{kb}}$). So $t + 0$, $t - 0$ are just t itself. Let $t - n$ denote the term s such that $t = s + n$. In case of presence of unary function symbol of weight 0 and there exists no s such that $s + n = t$, let $t - n$ $(n \geq 1)$ denote an arbitrary fixed term s' such that $s' \prec^{\mathsf{kb}}_{n} t$. Note that $t + n$, $t - n$ are just syntactic sugars. Gap orders between these terms can be brought back to $\mathscr{L}^{\mathbb{Z}}_{\mathsf{kb}}$. For example,

$$t + 3 \prec^{\mathsf{kb}} s - 5 \quad \leftrightarrow \quad t + 8 \prec^{\mathsf{kb}} s \quad \leftrightarrow \quad t \prec^{\mathsf{kb}}_{9} s.$$

As another example,

$$t \leqq^{\mathsf{kb}}_{2} s + 3 \quad \leftrightarrow \quad t \prec^{\mathsf{kb}} s \;\vee\; t = s \;\vee\; s \leqq^{\mathsf{kb}} t.$$

It is easily seen that for $n \geq 1$, $t \prec^{\sharp}_{n} s$ is equivalent to $t+(n-1) \prec^{\sharp} s$ or $t \prec^{\sharp} s-(n-1)$. Therefore we can rewrite (59) to

$$(\exists x : \mathsf{TA}) \Big[\bigwedge_{i=1}^{m} u_i + (p_i - 1) \trianglerighteq^{\sharp_i} x \ \wedge \ \bigwedge_{j=1}^{n} x \trianglerighteq^{*_j} v_j - (q_j - 1) \Big], \tag{60}$$

Now we guess the largest term from all lower bounds in

$$\Big\{ u_i + (p_i - 1) \mid 1 \leq i \leq m \Big\}$$

and dually, the smallest term from all upper bounds in

$$\Big\{ v_j - (q_j - 1) \mid 1 \leq j \leq n \Big\}.$$

Note that the combination is finite and in general, there could be multiple terms equal to the largest lower bound or the smallest upper bound. Without loss of generality we assume $u_1 + (p_1 - 1)$ (resp. $v_1 - (q_1 - 1)$) is only the largest lower bound (resp. the smallest upper bound) of x. Then (60) can be rewritten as

$$\begin{aligned} (\exists x : \mathsf{TA}) \Big[\bigwedge_{i=2}^{m} u_i + (p_i - 1) \prec^{\mathsf{kb}} u_1 + (p_1 - 1) \ \wedge \ u_1 + (p_1 - 1) \trianglerighteq^{\sharp_1} x \\ \wedge \ x \trianglerighteq^{*_1} v_1 - (q_1 - 1) \ \wedge \ \bigwedge_{j=2}^{n} v_1 - (q_1 - 1) \prec^{\mathsf{kb}} v_j - (n_j - 1) \Big], \end{aligned} \tag{61}$$

which in turn is equivalent to

$$\begin{aligned} \bigwedge_{i=2}^{m} u_i + (p_i - 1) \prec^{\mathsf{kb}} u_1 + (p_1 - 1) \\ \wedge \ (\exists x : \mathsf{TA}) \Big[u_1 + (p_1 - 1) \trianglerighteq^{\sharp_1} x \ \wedge \ x \trianglerighteq^{*_1} v_1 - (q_1 - 1) \Big] \\ \wedge \ \bigwedge_{j=2}^{n} v_1 - (q_1 - 1) \prec^{\mathsf{kb}} v_j - (n_j - 1) \end{aligned} \tag{62}$$

Let us consider

$$(\exists x : \mathsf{TA}) \Big[u_1 + (p_1 - 1) \trianglerighteq^{\sharp_1} x \ \wedge \ x \trianglerighteq^{*_1} v_1 - (q_1 - 1) \Big]$$

If necessary, we introduce boundary terms to delineate

$$u_1 + (p_1 - 1) \trianglerighteq^{\sharp_1} x \ \wedge \ x \trianglerighteq^{*_1} v_1 - (q_1 - 1).$$

This process does not increase open order literals. If the delineation process could assert x to be a boundary term, then we proceed as in Part 1 in Lemma 6. This also reduces the number of open order literals providing that x occurs in at least one open order literals in (59).

Now let us count the number of open order literals in (62). Suppose in (59) there are m' ordinary lower bounds and n' ordinary upper bounds. We assume $m', n' \geq 1$. Otherwise, obviously there is no open order literal left after eliminating x. Hence there are $m' + n'$ open literals in (59). We first count the number of open order literals outside $(\exists x : \mathsf{TA})$. There are four cases.

Case 1. Both $u_1 + (p_1 - 1)$ and $v_1 - (q_1 - 1)$ are ordinary. (More precisely, both u_1 and v_1 are.) Then there are $m' - 1 + n' - 1$ open literals.

Case 2. Both $u_1 + (p_1 - 1)$ and $v_1 - (q_1 - 1)$ are boundary terms. Then there are no open literals.

Case 3. $u_1 + (p_1 - 1)$ is an ordinary term and $v_1 - (q_1 - 1)$ is a boundary term. Then there are $m' - 1$ open literals.

Case 4. $u_1 + (p_1 - 1)$ is a boundary term and $v_1 - (q_1 - 1)$ is an ordinary term. Then there are $n' - 1$ open literals.

In any case, there are at least two fewer open order literals in the quantifier-free part of (62) than in (59). The the elimination of $(\exists x : \mathsf{TA})$ can generate at most one open literal. Therefore, the transformation from (59) to (62) reduces the number of open order literals. The rest of work is the same as that in Part 2 in Alg. 6.

□

D Negation Elimination

In this section we show the elimination of negative literals (in $\mathscr{L}_{\mathsf{kb}^+}$). Note that we do not care about negative integer literals.

First it is easily seen that

$$u \neq v \leftrightarrow u \prec^{\mathsf{w}} v \ \vee \ u \prec^{\mathsf{p}} v \ \vee \ u \prec^{\mathsf{l}} v \ \vee \ v \prec^{\mathsf{w}} u \ \vee \ v \prec^{\mathsf{p}} u \ \vee \ v \prec^{\mathsf{l}} u$$

Let $\sharp \in \{\mathsf{w}, \mathsf{p}, \mathsf{l}\}$. Recall that for $n \geq 0$.

$$u \leqq^{\sharp}_{n} v \leftrightarrow (u \prec^{\sharp}_{n} v) \wedge \neg(u \prec^{\sharp}_{n+1} v).$$

For $n > 0$, we have

$$\neg(u \prec^{\mathsf{w}}_{n} v) \leftrightarrow v \prec^{\mathsf{w}} u \ \vee \ \bigvee_{i=0}^{n-1} u \leqq^{\mathsf{w}}_{i} v,$$

$$\neg(u \prec^{\mathsf{p}}_{n} v) \leftrightarrow u^{\mathsf{w}} \neq v^{\mathsf{w}} \ \vee \ v \prec^{\mathsf{p}} u \ \vee \ \bigvee_{i=0}^{n-1} u \leqq^{\mathsf{p}}_{i} v,$$

$$\neg(u \prec^{\mathsf{l}}_{n} v) \leftrightarrow u^{\mathsf{w}} \neq v^{\mathsf{w}} \ \vee \ \neg\Big(\bigwedge_{1 \leq p \leq |\Sigma|} \mathsf{Is}_{\alpha_p}(u) \leftrightarrow \mathsf{Is}_{\alpha_p}(v)\Big) \ \vee \ v \prec^{\mathsf{l}} u \ \vee \ \bigvee_{i=0}^{n-1} u \leqq^{\mathsf{l}}_{i} v,$$

$$\neg(u \leqq^{\mathsf{w}}_{n} v) \leftrightarrow v \prec^{\mathsf{w}} u \ \vee \ \bigvee_{i=0}^{n-1} u \leqq^{\mathsf{w}}_{i} v \ \vee \ u \prec^{\mathsf{w}}_{n+1} v,$$

$$\neg(u \leqq^{\mathsf{p}}_{n} v) \leftrightarrow u^{\mathsf{w}} \neq v^{\mathsf{w}} \ \vee \ v \prec^{\mathsf{p}} u \ \vee \ \bigvee_{i=0}^{n-1} u \leqq^{\mathsf{p}}_{i} v \ \vee \ u \prec^{\mathsf{p}}_{n+1} v,$$

$$\begin{aligned}\neg(u \leqq^{\mathsf{l}}_{n} v) \leftrightarrow u^{\mathsf{w}} \neq v^{\mathsf{w}} \ \vee \ \neg\Big(\bigwedge_{1 \leq p \leq |\Sigma|} \mathsf{Is}_{\alpha_p}(u) \leftrightarrow \mathsf{Is}_{\alpha_p}(v)\Big) \ \vee \ v \prec^{\mathsf{l}} u \\ \vee \ \bigvee_{i=0}^{n-1} u \leqq^{\mathsf{l}}_{i} v \ \vee \ u \prec^{\mathsf{l}}_{n+1} v.\end{aligned}$$

Note that $u \leqq^{\sharp}_{0} v$ is just $u = v$ and $u \prec^{\sharp}_{0} v$ is just $u \prec^{\sharp} v \vee u = v$. So without loss of generality we assume that all literals occur positively.

We can also assume no negation of test literals occurs because

$$\neg\mathsf{Is}_{\alpha_p}(x) \leftrightarrow \bigvee_{\alpha_q \in \mathcal{C}, \alpha_p \neq \alpha_q} \mathsf{Is}_{\alpha_q}(x), \qquad \neg\mathsf{Is}_{\mathsf{A}}(x) \leftrightarrow \bigvee_{\alpha_p \in \mathcal{C} \setminus \mathcal{A}} \mathsf{Is}_{\alpha_p}(x).$$

E Definitions

In this section, we show the the integer functions used in reductions in Appendices F-O can be defined in Presburger arithmetic. As a consequence, we can freely use the following functions to form any integer terms.

E.1 Definition of $\mathsf{IsTW}^k(n)$.

$\mathsf{IsTW}^k(n)$ says that n is the weight of a k-tuple.

$$\exists z_1, \ldots, \exists z_k > 0 \Big(\sum_{i=1}^{k} z_i = n \wedge \bigwedge_{i=1}^{k} \mathsf{Tree}(z_i) \Big).$$

E.2 Definition of $\mathsf{IsTW}^k(n, p)$.

$\mathsf{IsTW}^k(n, p)$ says that n is the weight of a k-tuple whose first component is of type α_p.

$$\exists z_1, \ldots, \exists z_k > 0 \Big(\sum_{i=1}^{k} z_i = n \wedge \bigwedge_{i=1}^{k} \mathsf{Tree}(z_i) \wedge \mathsf{Tree}^{\alpha_p}(z_1) \Big).$$

E.3 Definition of $\mathsf{IsCW}^k(n, m)$.

$\mathsf{IsCW}^k(n, m)$ says that m is the weight of a component in a k-tuple of weight n.

$$\mathsf{IsTW}^k(n) \wedge \mathsf{IsTW}^{k-1}(n - m).$$

E.4 Definition of $\mathsf{IsCW}^k(n, m, p)$.

$\mathsf{IsCW}^k(n, m, p)$ says that m is the weight of a component in a k-tuple of weight n whose first component is of type α_p.

$$\mathsf{IsTW}^k(n, p) \wedge \mathsf{IsTW}^{k-1}(n - m)$$

E.5 Definition of $x = \mathsf{MinCW}^k(n, r, r')$.

$\mathsf{MinCW}^k(n, r, r')$ returns the smallest integer in (r, r') which is the weight of a component in a k-tuple of weight n. It returns r if there is no such integer.

$$\begin{aligned} &\Big[\neg \mathsf{IsTW}^k(n) \vee \forall z(r < z < r' \to \neg \mathsf{IsCW}^k(n, z)) \to x = r\Big] \vee \\ &\quad \Big[\mathsf{IsTW}^k(n) \wedge \exists z(r < z < r' \wedge \mathsf{IsCW}^k(n, z)) \\ &\qquad \to \mathsf{IsCW}^k(n, x) \wedge \forall z(r < z < x \to \neg \mathsf{IsCW}^k(n, z))\Big]. \end{aligned}$$

E.6 Definition of $x = \mathsf{MinCW}^k(n, p, r, r')$.

$\mathsf{MinCW}^k(n, p, r, r')$ returns the smallest integer in (r, r') which is the weight of a component in a k-tuple of weight n and with the first component be of type α_p. It returns r if there is no such integer.

$$\begin{aligned} &\Big[\neg \mathsf{IsTW}^k(n, p) \vee \forall z(r < z < r' \to \neg \mathsf{IsCW}^k(n, z, p)) \to x = r\Big] \vee \\ &\quad \Big[\mathsf{IsTW}^k(n, p) \wedge \exists z(r < z < r' \wedge \mathsf{IsCW}^k(n, z, p)) \\ &\qquad \to \mathsf{IsCW}^k(n, x, p) \wedge \forall z(r < z < x \to \neg \mathsf{IsCW}^k(n, z, p))\Big]. \end{aligned}$$

E.7 Definition of $x = \mathsf{MaxCW}^k(n, r, r')$.

$\mathsf{MaxCW}^k(n, r, r')$ returns the largest integer in (r, r') which is the weight of a component in a k-tuple of weight n. It returns r if there is no such integer.

$$\begin{aligned}&\left[\neg\mathsf{IsTW}^k(n) \lor \forall z(r < z < r' \to \neg\mathsf{IsCW}^k(n, z)) \to x = r\right] \lor \\ &\quad\left[\mathsf{IsTW}^k(n) \land \exists z(r < z < r' \land \mathsf{IsCW}^k(n, z))\right. \\ &\qquad\qquad \left.\to \mathsf{IsCW}^k(n, x) \land \forall z(x < z < r' \to \neg\mathsf{IsCW}^k(n, z))\right].\end{aligned}$$

E.8 Definition of $x = \mathsf{MaxCW}^k(n, p, r, r')$.

$\mathsf{MaxCW}^k(n, p, r, r')$ returns the largest integer in (r, r') which is the weight of a component in a k-tuple of weight n and with the first component be of type α_p. It returns r if there is no such integer.

$$\begin{aligned}&\left[\neg\mathsf{IsTW}^k(n, p) \lor \forall z(r < z < r' \to \neg\mathsf{IsCW}^k(n, z, p)) \to x = r\right] \lor \\ &\quad\left[\mathsf{IsTW}^k(n, p) \land \exists z(r < z < r' \land \mathsf{IsCW}^k(n, z, p))\right. \\ &\qquad\qquad \left.\to \mathsf{IsCW}^k(n, x, p) \land \forall z(x < z < r' \to \neg\mathsf{IsCW}^k(n, z, p))\right].\end{aligned}$$

E.9 Definition of $x = \mathsf{CW}^k_i(n, r, r')$ ($i > 0$).

$\mathsf{CW}^k_i(n, r, r')$ returns the i^{th} smallest integer in (r, r') which is the weight of a component in a k-tuple of weight n. Suppose that there are m ($m \geq 0$) such distinct integers in (r, r') which are weights of components in a k-tuple of weight n. If $i > m$, then $\mathsf{CW}^k_i(n, r, r') = \mathsf{MaxCW}^k(n, r, r')$. We give the definition by induction on m.
Case $i = 1$.

$$x = \mathsf{MinCW}^k(n, r, r').$$

Case $i > 1$.

$$\begin{aligned}&\left[x > \mathsf{CW}^k_{i-1}(n, r, r') \land r < x < r' \land \mathsf{IsCW}^k(n, x)\right. \\ &\quad\left.\land \forall z(\mathsf{CW}^k_{i-1}(n, r, r') < z < x \to \neg\mathsf{IsCW}^k(n, z)\right] \lor \\ &\qquad\left[\forall z(z > \mathsf{CW}^k_{i-1}(n) \land r < z < r' \to \neg\mathsf{IsCW}^k(n, z)) \land x = \mathsf{CW}^k_{i-1}(n)\right].\end{aligned}$$

E.10 Definition of $x = \mathsf{CW}^k_i(n, p, r, r')$ ($i > 0$).

$\mathsf{CW}^k_i(n, p, r, r')$ returns the i^{th} smallest integer in (r, r') which is the weight of a component in a k-tuple of weight n and with the first component of type α_p. Suppose that there are m ($m \geq 0$) such distinct integers in (r, r') which are weights of components in a k-tuple of weight n. If $i > m$, then $\mathsf{CW}^k_i(n, p, r, r') = \mathsf{MaxCW}^k(n, p, r, r')$. We give the definition by induction on m.
Case $i = 1$.

$$x = \mathsf{MinCW}^k(n, p, r, r').$$

Case $i > 1$.

$$\Big[x > \mathsf{CW}^k_{i-1}(n,p,r,r') \wedge r < x < r' \wedge \mathsf{IsCW}^k(n,x,p)$$
$$\wedge\ \forall z(\mathsf{CW}^k_{i-1}(n,p,r,r') < z < x \rightarrow \neg\mathsf{IsCW}^k(n,z,p)\Big] \vee$$
$$\Big[\forall z(z > \mathsf{CW}^k_{i-1}(n,p) \wedge r < z < r' \rightarrow \neg\mathsf{IsCW}^k(n,z,p)) \wedge x = \mathsf{CW}^k_{i-1}(n,p)\Big].$$

With no risk of confusion, we write $\mathsf{MinCW}^k(n)$, $\mathsf{MaxCW}^k(n)$ and $\mathsf{CW}^k_i(n)$ for $\mathsf{MinCW}^k(n,0,n+1)$, $\mathsf{MaxCW}^k(n,0,n+1)$ and $\mathsf{CW}^k_i(n,0,n+1)$ respectively. Similarly $\mathsf{MinCW}^k(n,p)$, $\mathsf{MaxCW}^k(n,p)$ and $\mathsf{CW}^k_i(n,p)$ stand for $\mathsf{MinCW}^k(n,p,0,n+1)$, $\mathsf{MaxCW}^k(n,p,0,n+1)$ and $\mathsf{CW}^k_i(n,p,0,n+1)$ respectively.

E.11 Definition of $x = \mathsf{SCW}^k_i(n)$ $(1 \leq i \leq k)$.

$\mathsf{SCW}^k_i(n)$ returns the weight of the i^{th} component of the smallest k-tuple (w.r.t. $\prec^{k;\mathsf{kb}}$) of weight n. It returns 0 if no such weight exists. We define it inductively as follows.
Case $i = 1$.

$$x = \mathsf{MinCW}^k(n).$$

Case $1 < i \leq k$.

$$\neg\mathsf{IsTW}^k(n) \rightarrow x = 0 \wedge$$
$$\mathsf{IsTW}^k(n) \rightarrow x = \mathsf{MinCW}^{k-i+1}\Big(n - \sum_{j=1}^{i-1} \mathsf{SCW}^k_j(n)\Big).$$

E.12 Definition of $x = \mathsf{SCW}^k_i(n,p)$ $(1 \leq i \leq k)$.

$\mathsf{SCW}^k_i(n)$ returns the weight of the i^{th} component of the smallest k-tuple (w.r.t. $\prec^{k;\mathsf{kb}}$) of weight n and with the first component of type α_p. It returns 0 if no such weight exists. We define it inductively as follows.
Case $i = 1$.

$$x = \mathsf{MinCW}^k(n,p).$$

Case $1 < j \leq k$.

$$\neg\mathsf{IsTW}^k(n,p) \rightarrow x = 0 \wedge$$
$$\mathsf{IsTW}^k(n,p) \rightarrow x = \mathsf{MinCW}^{k-i+1}\Big(n - \sum_{j=1}^{i-1} \mathsf{SCW}^k_j(n,p)\Big).$$

E.13 Definition of $x = \mathsf{LCW}^k_i(n)$ $(1 \leq i \leq k)$.

$\mathsf{LCW}^k_i(n)$ returns the weight of the i^{th} component of the largest k-tuple (w.r.t. $\prec^{k;\mathsf{kb}}$) of weight n. It returns 0 if no such weight exists. We define it inductively as follows.
Case $i = 1$.

$$x = \mathsf{MaxCW}^k(n).$$

Case $1 < i \leq k$.

$$\neg \mathsf{IsTW}^k(n) \rightarrow x = 0 \wedge$$
$$\mathsf{IsTW}^k(n) \rightarrow x = \mathsf{MaxCW}^{k-i+1}\Big(n - \sum_{j=1}^{i-1} \mathsf{LCW}_j^k(n)\Big).$$

E.14 Definition of $x = \mathsf{LCW}_i^k(n, p)$ $(1 \leq i \leq k)$.

$\mathsf{LCW}_i^k(n, p)$ returns the weight of the i^{th} component of the largest k-tuple (w.r.t. $\prec^{k;\mathsf{kb}}$) of weight n and with the first component of type α_p. It returns 0 if no such weight exists. We define it inductively as follows.
Case $i = 1$.

$$x = \mathsf{MaxCW}^k(n, p).$$

Case $1 < i \leq k$.

$$\neg \mathsf{IsTW}^k(n, p) \rightarrow x = 0 \wedge$$
$$\mathsf{IsTW}^k(n, p) \rightarrow x = \mathsf{MaxCW}^{k-i+1}\Big(n - \sum_{j=1}^{i-1} \mathsf{LCW}_j^k(n, p)\Big).$$

E.15 Definition of Well-definedness.

The following are well-definedness predicates for boundary terms or boundary tuples.

$$\begin{aligned}
\mathsf{WD}_{0^{\mathsf{w}}}(m) &\leftrightarrow \mathsf{Tree}(m)\\
\mathsf{WD}_{1^{\mathsf{w}}}(m) &\leftrightarrow \mathsf{Tree}(m)\\
\mathsf{WD}_{0^{\mathsf{p}}}(m, p) &\leftrightarrow \mathsf{Tree}^{(\alpha_p)}(m)\\
\mathsf{WD}_{1^{\mathsf{p}}}(m, p) &\leftrightarrow \mathsf{Tree}^{(\alpha_p)}(m)\\
\mathsf{WD}_{\bar{\mathbf{0}}^{k;\mathsf{kb}}}(n) &\leftrightarrow \mathsf{IsTW}^k(n)\\
\mathsf{WD}_{\bar{\mathbf{1}}^{k;\mathsf{kb}}}(n) &\leftrightarrow \mathsf{IsTW}^k(n)\\
\mathsf{WD}_{\bar{\mathbf{0}}^{k;\mathsf{w}}}(n, m) &\leftrightarrow \mathsf{IsCW}^k(n, m)\\
\mathsf{WD}_{\bar{\mathbf{1}}^{k;\mathsf{w}}}(n, m) &\leftrightarrow \mathsf{IsCW}^k(n, m)\\
\mathsf{WD}_{\bar{\mathbf{0}}^{k;\mathsf{p}}}(n, m, p) &\leftrightarrow \mathsf{IsCW}^k(n, m) \wedge \mathsf{Tree}^{(\alpha_p)}(m)\\
\mathsf{WD}_{\bar{\mathbf{1}}^{k;\mathsf{p}}}(n, m, p) &\leftrightarrow \mathsf{IsCW}^k(n, m) \wedge \mathsf{Tree}^{(\alpha_p)}(m)
\end{aligned}$$

F Reductions of Open Gap Orders

From from this section to Sec. O, we define reductions on $\prec_n^\sharp$ and $\prec_n^{k;\sharp}$ ($\sharp \in \{\mathsf{w}, \mathsf{p}, \mathsf{l}\}$, $n \geq 0$, $k > 0$). The reductions on $\leqq_n^\sharp$ and $\leqq_n^{k;\sharp}$ can be similarly defined.

F.1 Reduction of $u \prec_n^{\mathsf{w}} v$.

$$\bigvee_{\substack{n_1+n_2+n_3=n \\ n_1,n_3\geq 0, n_2 \geq 1}} u \prec_{n_1}^{\mathsf{pl}} \mathsf{1}^{\mathsf{w}}_{(u^{\mathsf{w}})} \prec_{n_2}^{\mathsf{w}} \mathsf{0}^{\mathsf{w}}_{(v^{\mathsf{w}})} \prec_{n_3}^{\mathsf{pl}} v.$$

F.2 Reduction of $u \prec_n^{\mathsf{p}} v$.

Let $\mathsf{type}(u) = \alpha_p$ and $\mathsf{type}(v) = \alpha_q$.

$$\bigvee_{\substack{n_1+n_2+n_3=n \\ n_1,n_3\geq 0, n_2 \geq 1}} u \prec_{n_1}^{\mathsf{l}} \mathsf{1}^{\mathsf{p}}_{u^{\mathsf{w}},p} \prec_{n_2}^{\mathsf{p}} \mathsf{0}^{\mathsf{p}}_{v^{\mathsf{w}},q} \prec_{n_3}^{\mathsf{l}} v.$$

F.3 Reduction of $u \prec_n^{\mathsf{l}} v$.

Let $u = \alpha(u_1, \ldots, u_k)$ and $v = \alpha(v_1, \ldots, v_k)$.

$$\langle u_1, \ldots, u_k \rangle \prec_n^{k;\mathsf{kb}} \langle v_1, \ldots, v_k \rangle.$$

G Reduction of Half Open Equalities.

Let $u = \alpha_q(u_1, \ldots, u_k)$.

G.1 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} = u$.

$$\bigwedge_{i=1}^{q-1} \neg\mathsf{Tree}^{(\alpha_i)}(u^{\mathsf{w}}) \wedge m = u^{\mathsf{w}} \wedge \mathsf{1}^{\mathsf{kb}}_{(u^{\mathsf{w}}-\mathsf{W}(\alpha_q))} = \langle u_1, \ldots, u_k \rangle$$

G.2 Reduction of $\mathbf{1}^{\mathsf{p}}_{(m,p)} = u$.

$$p = q \wedge m = u^{\mathsf{w}} \wedge \mathsf{1}^{\mathsf{kb}}_{(u^{\mathsf{w}}-\mathsf{W}(\alpha_q))} = \langle u_1, \ldots, u_k \rangle$$

G.3 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} = u$.

$$\bigwedge_{i=q+1}^{|\Sigma|} \neg\mathsf{Tree}^{(\alpha_i)}(u^{\mathsf{w}}) \wedge m = u^{\mathsf{w}} \wedge \mathsf{0}^{\mathsf{kb}}_{(u^{\mathsf{w}}-\mathsf{W}(\alpha_q))} = \langle u_1, \ldots, u_k \rangle$$

G.4 Reduction of $\mathbf{0}^{\mathsf{p}}_{(m,p)} = u$.

$$p = q \wedge m = u^{\mathsf{w}} \wedge \mathsf{0}^{\mathsf{kb}}_{(u^{\mathsf{w}}-\mathsf{W}(\alpha_p))} = \langle u_1, \ldots, u_k \rangle$$

H Reduction of Half Open Gap Orders

Let $u = \alpha_q(u_1, \dots, u_k)$.

H.1 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n} u$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 1, n_2\geq 0}} \mathsf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_1} \mathsf{0}^{\mathsf{w}}_{(u^{\mathsf{w}})} \prec^{\mathsf{pl}}_{n_2} u.$$

H.2 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{p}}_{n} u$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 1, n_2\geq 0}} \mathsf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{p}}_{n_1} \mathsf{0}^{\mathsf{p}}_{(m,q)} \prec^{\mathsf{l}}_{n_2} u.$$

H.3 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{l}}_{n} u$.

$$\mathsf{Is}_{\alpha_q}(u) \wedge m = u^{\mathsf{w}} \wedge \bigwedge_{i=q+1}^{|\Sigma|} \neg\mathsf{Tree}^{(\alpha_i)}(u^{\mathsf{w}}) \wedge \bar{\mathbf{0}}^{k;\mathsf{kb}}_{(m-\mathsf{W}(\alpha_q))} \prec^{k;\mathsf{kb}}_{n} \langle u_1, \dots, u_k\rangle.$$

H.4 Reduction of $\mathbf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{w}}_{n} u$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 1, n_2\geq 0}} \mathsf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{w}}_{n_1} \mathsf{0}^{\mathsf{w}}_{(u^{\mathsf{w}})} \prec^{\mathsf{pl}}_{n_2} u.$$

H.5 Reduction of $\mathbf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{p}}_{n} u$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 1, n_2\geq 0}} \mathsf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{p}}_{n_1} \mathsf{0}^{\mathsf{p}}_{(m,q)} \prec^{\mathsf{l}}_{n_2} u.$$

H.6 Reduction of $\mathbf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{l}}_{n} u$.

$$p = q \wedge \mathsf{Is}_{\alpha_q}(u) \wedge m = u^{\mathsf{w}} \wedge \bar{\mathbf{0}}^{k;\mathsf{kb}}_{(m-\mathsf{W}(\alpha_q))} \prec^{k;\mathsf{kb}}_{n} \langle u_1, \dots, u_k\rangle.$$

H.7 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n} u$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 1, n_2\geq 0}} \mathsf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_1} \mathsf{0}^{\mathsf{w}}_{(u^{\mathsf{w}})} \prec^{\mathsf{pl}}_{n_2} u.$$

H.8 Reduction of $\mathbf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{w}}_{n} u$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 1, n_2\geq 0}} \mathsf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{w}}_{n_1} \mathsf{0}^{\mathsf{w}}_{(u^{\mathsf{w}})} \prec^{\mathsf{pl}}_{n_2} u.$$

H.9 Reduction of $u \prec^{\mathsf{w}}_{n} \mathsf{1}^{\mathsf{w}}_{(m)}$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 0, n_2\geq 1}} u \prec^{\mathsf{pl}}_{n_1} \mathsf{1}^{\mathsf{w}}_{(u^{\mathsf{w}})} \prec^{\mathsf{w}}_{n_2} \mathsf{1}^{\mathsf{w}}_{(m)}.$$

H.10 Reduction of $u \prec^{\mathsf{p}}_{n} \mathsf{1}^{\mathsf{w}}_{(m)}$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 0, n_2\geq 1}} u \prec^{\mathsf{l}}_{n_1} \mathsf{1}^{\mathsf{p}}_{(m,q)} \prec^{\mathsf{p}}_{n_2} \mathsf{1}^{\mathsf{w}}_{(m)}.$$

H.11 Reduction of $u \prec^{\mathsf{l}}_{n} \mathsf{1}^{\mathsf{w}}_{(m)}$.

$$\mathsf{Is}_{\alpha_q}(u) \wedge m = u^{\mathsf{w}} \wedge \bigwedge_{i=1}^{q-1} \neg \mathsf{Tree}^{(\alpha_i)}(u^{\mathsf{w}}) \wedge \langle u_1, \ldots, u_k \rangle \prec^{k;\mathsf{kb}}_{n} \bar{\mathbf{1}}^{k;\mathsf{kb}}_{(m-\mathsf{W}(\alpha_q))}.$$

H.12 Reduction of $u \prec^{\mathsf{w}}_{n} \mathsf{1}^{\mathsf{p}}_{(m,p)}$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 0, n_2\geq 1}} u \prec^{\mathsf{pl}}_{n_1} \mathsf{1}^{\mathsf{w}}_{(u^{\mathsf{w}})} \prec^{\mathsf{w}}_{n_2} \mathsf{1}^{\mathsf{p}}_{(m,p)}.$$

H.13 Reduction of $u \prec^{\mathsf{p}}_{n} \mathsf{1}^{\mathsf{p}}_{(m,p)}$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 0, n_2\geq 1}} u \prec^{\mathsf{l}}_{n_1} \mathsf{1}^{\mathsf{p}}_{(m,q)} \prec^{\mathsf{p}}_{n_2} \mathsf{1}^{\mathsf{p}}_{(m,p)}.$$

H.14 Reduction of $u \prec^{\mathsf{l}}_{n} \mathsf{1}^{\mathsf{p}}_{(m,p)}$.

$$p = q \wedge \mathsf{Is}_{\alpha_q}(u) \wedge m = u^{\mathsf{w}} \wedge \langle u_1, \ldots, u_k \rangle \prec^{k;\mathsf{kb}}_{n} \bar{\mathbf{1}}^{k;\mathsf{kb}}_{(m-\mathsf{W}(\alpha_q))}.$$

H.15 Reduction of $u \prec^{\mathsf{w}}_{n} \mathsf{0}^{\mathsf{w}}_{(m)}$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 0, n_2\geq 1}} u \prec^{\mathsf{pl}}_{n_1} \mathsf{1}^{\mathsf{w}}_{(u^{\mathsf{w}})} \prec^{\mathsf{w}}_{n_2} \mathsf{0}^{\mathsf{w}}_{(m)}.$$

H.16 Reduction of $u \prec^{\mathsf{w}}_{n} \mathsf{0}^{\mathsf{p}}_{(m,p)}$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 0, n_2\geq 1}} u \prec^{\mathsf{pl}}_{n_1} \mathsf{1}^{\mathsf{w}}_{(u^{\mathsf{w}})} \prec^{\mathsf{w}}_{n_2} \mathsf{0}^{\mathsf{p}}_{(m,p)}.$$

I Reduction of Closed Equalities

I.1 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} = \mathbf{0}^{\mathsf{w}}_{(m')}$.

$$m = m'.$$

I.2 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} = \mathbf{0}^{\mathsf{p}}_{(m',p')}$.

$$m = m' \wedge \bigwedge_{i=p'+1}^{|\Sigma|} \neg\mathsf{Tree}^{(\alpha_i)}(m).$$

I.3 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} = \mathbf{1}^{\mathsf{w}}_{(m')}$.

$$m = m' \wedge \neg\mathsf{CNT}_1(m).$$

I.4 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} = \mathbf{1}^{\mathsf{p}}_{(m',p')}$.

$$m = m' \wedge \bigwedge_{i=p'+1}^{|\Sigma|} \neg\mathsf{Tree}^{(\alpha_i)}(m) \wedge \neg\mathsf{CNT}_1^{(\alpha_{p'})}(m).$$

I.5 Reduction of $\mathbf{0}^{\mathsf{p}}_{(m,p)} = \mathbf{0}^{\mathsf{p}}_{(m',p')}$.

$$m = m' \wedge p = p'.$$

I.6 Reduction of $\mathbf{0}^{\mathsf{p}}_{(m,p)} = \mathbf{1}^{\mathsf{w}}_{(m')}$.

$$m = m' \wedge \bigwedge_{i=1}^{p-1} \neg\mathsf{Tree}^{(\alpha_i)}(m) \wedge \neg\mathsf{CNT}_1^{(\alpha_p)}(m).$$

I.7 Reduction of $\mathbf{0}^{\mathsf{p}}_{(m,p)} = \mathbf{1}^{\mathsf{p}}_{(m',p')}$.

$$m = m' \wedge p = p' \wedge \neg\mathsf{CNT}_1^{(\alpha_p)}(m).$$

I.8 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} = \mathbf{1}^{\mathsf{w}}_{(m')}$.

$$m = m'.$$

I.9 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} = \mathbf{1}^{\mathsf{p}}_{(m',p')}$.

$$m = m' \wedge \bigwedge_{i=1}^{p'-1} \neg\mathsf{Tree}^{(\alpha_i)}(m).$$

I.10 Reduction of $\mathbf{1}^{\mathsf{p}}_{(m,p)} = \mathbf{1}^{\mathsf{p}}_{(m',p')}$.

$$m = m' \wedge p = p'$$

J Reduction of Closed Gap Orders

J.1 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n} \mathbf{0}^{\mathsf{w}}_{(m')}$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 0, n_2\geq 1}} \mathsf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{pl}}_{n_1} \mathsf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_2} \mathsf{0}^{\mathsf{w}}_{(m')}.$$

J.2 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{p}}_{n} \mathbf{0}^{\mathsf{w}}_{(m')}$.

$$\mathsf{false}.$$

J.3 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{l}}_{n} \mathbf{0}^{\mathsf{w}}_{(m')}$.

$$\mathsf{false}.$$

J.4 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n} \mathbf{0}^{\mathsf{p}}_{(m',p')}$.

$$\bigvee_{\substack{n_1+n_2+n_3=n\\ n_1,n_3\geq 0, n_2\geq 1}} \mathsf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{pl}}_{n_1} \mathsf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_2} \mathsf{0}^{\mathsf{w}}_{(m')} \prec^{\mathsf{w}}_{n_3} \mathsf{0}^{\mathsf{p}}_{(m',p')}.$$

J.5 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{p}}_{n} \mathbf{0}^{\mathsf{p}}_{(m',p')}$.

$$m = m' \wedge \bigvee_{\substack{n_{p'+1}+\ldots+n_{|\Sigma|}=n\\ n_{p'+1},\ldots,n_{|\Sigma|}\geq 0}} \Big[\bigwedge_{i=p'+1}^{|\Sigma|} \big(n_i > 0 \rightarrow \mathsf{CNT}^{(\alpha_i)}_{n_i-1}(m)\big)\Big].$$

J.6 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{l}}_{n} \mathbf{0}^{\mathsf{p}}_{(m',p')}$.

$$\mathsf{false}.$$

J.7 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n} \mathbf{1}^{\mathsf{w}}_{(m')}$.

$$\bigvee_{\substack{n_1+n_2+n_3=n\\ n_1,n_3\geq 0, n_2\geq 1}} \mathsf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{pl}}_{n_1} \mathsf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_2} \mathsf{0}^{\mathsf{w}}_{(m')} \prec^{\mathsf{pl}}_{n_3} \mathsf{1}^{\mathsf{w}}_{(m')}.$$

J.8 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{p}}_{n} \mathbf{1}^{\mathsf{w}}_{(m')}$.

$$\begin{aligned} m = m' \wedge \bigvee_{\substack{p,p'\in\{1,\ldots,|\Sigma|\}\\ p\neq p'}} & \Big[\mathsf{Tree}^{(\alpha_p)}(m) \wedge \mathsf{Tree}^{(\alpha_{p'})}(m)\Big] \wedge \\ & \bigvee_{\substack{n_1+\ldots+n_{|\Sigma|}=n+1\\ n_1,\ldots,n_{|\Sigma|}\geq 0}} \Big[\bigwedge_{i=1}^{|\Sigma|} \big(n_i > 0 \rightarrow \mathsf{CNT}^{(\alpha_i)}_{n_i-1}(m)\big)\Big]. \end{aligned}$$

J.9 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{l}}_{n} \mathbf{1}^{\mathsf{w}}_{(m')}$.

$$m = m' \wedge \bigwedge_{\substack{p,p' \in \{1,\ldots,|\Sigma|\} \\ p \neq p'}} \left[\mathsf{Tree}^{(\alpha_p)}(m) \rightarrow \neg\mathsf{Tree}^{(\alpha_{p'})}(m)\right] \wedge$$
$$\bigvee_{\substack{n_1+\ldots+n_{|\Sigma|}=n+1 \\ n_1,\ldots,n_{|\Sigma|}\geq 0}} \left[\bigwedge_{i=1}^{|\Sigma|} \left(n_i > 0 \rightarrow \mathsf{CNT}^{(\alpha_i)}_{n_i-1}(m)\right)\right].$$

J.10 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n} \mathbf{1}^{\mathsf{p}}_{(m',p')}$.

$$\bigvee_{\substack{n_1+n_2+n_3+n_4=n \\ n_1,n_3,n_4\geq 0, n_2\geq 1}} \mathsf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{pl}}_{n_1} \mathsf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_2} \mathsf{0}^{\mathsf{w}}_{(m')} \prec^{\mathsf{pl}}_{n_3} \mathsf{0}^{\mathsf{w}}_{(m',p')} \prec^{\mathsf{pl}}_{n_4} \mathsf{1}^{\mathsf{w}}_{(m',p')}.$$

J.11 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{p}}_{n} \mathbf{1}^{\mathsf{p}}_{(m',p')}$.

$$m = m' \wedge \bigvee_{p \in \{p'+1,\ldots,|\Sigma|\}} \left[\mathsf{Tree}^{(\alpha_p)}(m)\right] \wedge$$
$$\bigvee_{\substack{n_{p'}+\ldots+n_{|\Sigma|}=n+1 \\ n_{p'},\ldots,n_{|\Sigma|}\geq 0}} \left[\bigwedge_{i=p'}^{|\Sigma|} \left(n_i > 0 \rightarrow \mathsf{CNT}^{(\alpha_i)}_{n_i-1}(m)\right)\right].$$

J.12 Reduction of $\mathbf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{l}}_{n} \mathbf{1}^{\mathsf{p}}_{(m',p')}$.

$$m = m' \wedge \bigwedge_{p \in \{p'+1,\ldots,|\Sigma|\}} \left[\neg\mathsf{Tree}^{(\alpha_p)}(m)\right] \wedge \mathsf{CNT}^{(\alpha_{p'})}_{n}(m).$$

J.13 Reduction of $\mathbf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{w}}_{n} \mathbf{0}^{\mathsf{w}}_{(m')}$.

$$\bigvee_{\substack{n_1+n_2=n \\ n_1\geq 0, n_2\geq 1}} \mathsf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{pl}}_{n_1} \mathsf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_2} \mathsf{0}^{\mathsf{w}}_{(m')}.$$

J.14 Reduction of $\mathbf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{p}}_{n} \mathbf{0}^{\mathsf{w}}_{(m')}$.

$$\mathsf{false}.$$

J.15 Reduction of $\mathbf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{l}}_{n} \mathbf{0}^{\mathsf{w}}_{(m')}$.

$$\mathsf{false}.$$

J.16 Reduction of $\mathsf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{w}}_{n} \mathsf{0}^{\mathsf{p}}_{(m',p')}$.

$$\bigvee_{\substack{n_1+n_2+n_3+n_4=n \\ n_1,n_2,n_4\geq 0, n_3\geq 1}} \mathsf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{pl}}_{n_1} \mathsf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{pl}}_{n_2} \mathsf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_3} \mathsf{0}^{\mathsf{w}}_{(m')} \prec^{\mathsf{pl}}_{n_4} \mathsf{0}^{\mathsf{p}}_{(m',p')}.$$

J.17 Reduction of $\mathsf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{p}}_{n} \mathsf{0}^{\mathsf{p}}_{(m',p')}$.

$$m = m' \wedge p' < p \wedge \bigvee_{\substack{n_{p'}+\ldots+n_{p-1}=n \\ n_{p'},\ldots,n_{p-1}\geq 0}} \Big[\bigwedge_{i=p'}^{p-1} \big(n_i > 0 \rightarrow \mathsf{CNT}^{(\alpha_i)}_{n_i-1}(m) \big) \Big].$$

J.18 Reduction of $\mathsf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{l}}_{n} \mathsf{0}^{\mathsf{p}}_{(m',p')}$.

false.

J.19 Reduction of $\mathsf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{w}}_{n} \mathsf{1}^{\mathsf{w}}_{(m')}$.

$$\bigvee_{\substack{n_1+n_2+n_3+n_4=n \\ n_1,n_2,n_4\geq 0, n_3\geq 1}} \mathsf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{pl}}_{n_1} \mathsf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{pl}}_{n_2} \mathsf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_3} \mathsf{0}^{\mathsf{w}}_{(m')} \prec^{\mathsf{pl}}_{n_4} \mathsf{1}^{\mathsf{w}}_{(m')}.$$

J.20 Reduction of $\mathsf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{p}}_{n} \mathsf{1}^{\mathsf{w}}_{(m')}$.

$$m = m' \wedge \bigvee_{p'\in\{1,\ldots,p-1\}} \Big[\mathsf{Tree}^{(\alpha_{p'})}(m) \Big] \wedge \bigvee_{\substack{n_1+\ldots+n_p=n+1 \\ n_1,\ldots,n_p\geq 0}} \Big[\bigwedge_{i=1}^{p} \big(n_i > 0 \rightarrow \mathsf{CNT}^{(\alpha_i)}_{n_i-1}(m) \big) \Big].$$

J.21 Reduction of $\mathsf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{l}}_{n} \mathsf{1}^{\mathsf{w}}_{(m')}$.

$$m = m' \wedge \bigwedge_{p'\in\{1,\ldots,p-1\}} \Big[\neg\mathsf{Tree}^{(\alpha_{p'})}(m) \Big] \wedge \mathsf{CNT}^{(\alpha_p)}_{n}(m).$$

J.22 Reduction of $\mathsf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{w}}_{n} \mathsf{1}^{\mathsf{p}}_{(m',p')}$.

$$\bigvee_{\substack{n_1+n_2+n_3+n_4+n_5=n \\ n_1,n_2,n_4,n_5\geq 0, n_3\geq 1}} \mathsf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{pl}}_{n_1} \mathsf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{pl}}_{n_2} \mathsf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_3} \mathsf{0}^{\mathsf{w}}_{(m')} \prec^{\mathsf{pl}}_{n_4} \mathsf{0}^{\mathsf{p}}_{(m',p')} \prec^{\mathsf{pl}}_{n_5} \mathsf{1}^{\mathsf{p}}_{(m',p')}.$$

J.23 Reduction of $\mathbf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{p}}_{n} \mathbf{1}^{\mathsf{p}}_{(m',p')}$.

$$m = m' \wedge p' < p \wedge \bigvee_{\substack{n_{p'}+\ldots+n_p=n+1\\ n_{p'},\ldots,n_p\geq 0}} \Big[\bigwedge_{i=p'}^{p} \big(n_i > 0 \rightarrow \mathsf{CNT}^{(\alpha_i)}_{n_i-1}(m)\big)\Big].$$

J.24 Reduction of $\mathbf{0}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{l}}_{n} \mathbf{1}^{\mathsf{p}}_{(m',p')}$.

$$m = m' \wedge p = p' \wedge \mathsf{CNT}^{(\alpha_p)}_{n}(m).$$

J.25 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n} \mathbf{0}^{\mathsf{w}}_{(m')}$.

Case $n = 1$.

$$m < m'.$$

Case $n > 1$.

$$\bigvee_{r=1}^{n-1} \exists z_1 \ldots \exists z_r \Big[m < z_1 < \ldots < z_r < m' \wedge \bigvee_{\substack{n_1+\ldots+n_r=n-1\\ n_1,\ldots,n_r>0}} \bigwedge_{i=1}^{r} \mathsf{CNT}_{n_i-1}(z_i)\Big].$$

J.26 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{p}}_{n} \mathbf{0}^{\mathsf{w}}_{(m')}$.

$$\mathsf{false}.$$

J.27 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{l}}_{n} \mathbf{0}^{\mathsf{w}}_{(m')}$.

$$\mathsf{false}.$$

J.28 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n} \mathbf{0}^{\mathsf{p}}_{(m',p')}$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_2\geq 0, n_1\geq 1}} \mathsf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_1} \mathsf{0}^{\mathsf{w}}_{(m')} \prec^{\mathsf{pl}}_{n_2} \mathsf{0}^{\mathsf{p}}_{(m',p')}.$$

J.29 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{p}}_{n} \mathbf{0}^{\mathsf{p}}_{(m',p')}$.

$$\mathsf{false}.$$

J.30 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{l}}_{n} \mathbf{0}^{\mathsf{p}}_{(m',p')}$.

$$\mathsf{false}.$$

J.31 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n} \mathbf{1}^{\mathsf{w}}_{(m')}$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_2\geq 0, n_1\geq 1}} \mathsf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_1} \mathsf{0}^{\mathsf{w}}_{(m')} \prec^{\mathsf{pl}}_{n_2} \mathsf{1}^{\mathsf{w}}_{(m')}.$$

J.32 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{p}}_{n} \mathbf{1}^{\mathsf{w}}_{(m')}$.

false.

J.33 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{l}}_{n} \mathbf{1}^{\mathsf{w}}_{(m')}$.

false.

J.34 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n} \mathbf{1}^{\mathsf{p}}_{(m',p')}$.

$$\bigvee_{\substack{n_1+n_2+n_3=n\\ n_2,n_3\geq 0, n_1\geq 1}} 1^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_1} 0^{\mathsf{w}}_{(m')} \prec^{\mathsf{pl}}_{n_2} 0^{\mathsf{w}}_{(m',p')} \prec^{\mathsf{pl}}_{n_3} 1^{\mathsf{w}}_{(m',p')}.$$

J.35 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{p}}_{n} \mathbf{1}^{\mathsf{p}}_{(m',p')}$.

false.

J.36 Reduction of $\mathbf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{l}}_{n} \mathbf{1}^{\mathsf{p}}_{(m',p')}$.

false.

J.37 Reduction of $\mathbf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{w}}_{n} \mathbf{0}^{\mathsf{w}}_{(m')}$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 0, n_2\geq 1}} 1^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{pl}}_{n_1} 1^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_2} 0^{\mathsf{w}}_{(m')}.$$

J.38 Reduction of $\mathbf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{p}}_{n} \mathbf{0}^{\mathsf{w}}_{(m')}$.

false.

J.39 Reduction of $\mathbf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{l}}_{n} \mathbf{0}^{\mathsf{w}}_{(m')}$.

false.

J.40 Reduction of $\mathbf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{w}}_{n} \mathbf{0}^{\mathsf{p}}_{(m',p')}$.

$$\bigvee_{\substack{n_1+n_2+n_3=n\\ n_1,n_3\geq 0, n_2\geq 1}} 1^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{pl}}_{n_1} 1^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_2} 0^{\mathsf{w}}_{(m')} \prec^{\mathsf{w}}_{n_3} 0^{\mathsf{p}}_{(m',p')}.$$

J.41 Reduction of $\mathsf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{p}}_{n} \mathsf{0}^{\mathsf{p}}_{(m',p')}$.

Case $n = 1$.

$$m = m' \wedge p' < p.$$

Case $n > 1$.

$$m = m' \wedge p' < p - 1 \wedge \bigvee_{\substack{n_{p'+1}+\ldots+n_{p-1}=n-1 \\ n_{p'+1},\ldots,n_{p-1}\geq 0}} \Big[\bigwedge_{i=p'+1}^{p-1} \Big(n_i > 0 \rightarrow \mathsf{CNT}^{(\alpha_i)}_{n_i-1}(m) \Big) \Big].$$

J.42 Reduction of $\mathsf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{l}}_{n} \mathsf{0}^{\mathsf{p}}_{(m',p')}$.

$$\mathsf{false}.$$

J.43 Reduction of $\mathsf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{w}}_{n} \mathsf{1}^{\mathsf{w}}_{(m')}$.

$$\bigvee_{\substack{n_1+n_2+n_3=n \\ n_1,n_3\geq 0, n_2\geq 1}} \mathsf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{pl}}_{n_1} \mathsf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_2} \mathsf{0}^{\mathsf{w}}_{(m')} \prec^{\mathsf{pl}}_{n_3} \mathsf{1}^{\mathsf{w}}_{(m')}.$$

J.44 Reduction of $\mathsf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{p}}_{n} \mathsf{1}^{\mathsf{w}}_{(m')}$.

$$m = m' \wedge \bigvee_{\substack{n_1+\ldots+n_{p-1}=n \\ n_1,\ldots,n_{p-1}\geq 0}} \Big[\bigwedge_{i=1}^{p-1} \Big(n_i > 0 \rightarrow \mathsf{CNT}^{(\alpha_i)}_{n_i-1}(m) \Big) \Big].$$

J.45 Reduction of $\mathsf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{l}}_{n} \mathsf{1}^{\mathsf{w}}_{(m')}$.

$$\mathsf{false}.$$

J.46 Reduction of $\mathsf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{w}}_{n} \mathsf{1}^{\mathsf{p}}_{(m',p')}$.

$$\bigvee_{\substack{n_1+n_2+n_3+n_4=n \\ n_1,n_3,n_4\geq 0, n_2\geq 1}} \mathsf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{pl}}_{n_1} \mathsf{1}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_2} \mathsf{0}^{\mathsf{w}}_{(m')} \prec^{\mathsf{pl}}_{n_3} \mathsf{0}^{\mathsf{w}}_{(m',p')} \prec^{\mathsf{pl}}_{n_4} \mathsf{1}^{\mathsf{w}}_{(m',p')}.$$

J.47 Reduction of $\mathsf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{p}}_{n} \mathsf{1}^{\mathsf{p}}_{(m',p')}$.

$$m = m' \wedge p' < p \wedge \bigvee_{\substack{n_{p'}+\ldots+n_{p-1}=n \\ n_{p'}\ldots,n_{p-1}\geq 0}} \Big[\bigwedge_{i=p'}^{p-1} \Big(n_i > 0 \rightarrow \mathsf{CNT}^{(\alpha_i)}_{n_i-1}(m) \Big) \Big].$$

J.48 Reduction of $\mathsf{1}^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{l}}_{n} \mathsf{1}^{\mathsf{p}}_{(m',p')}$.

$$\mathsf{false}.$$

K Reductions of Open Gap Orders on Tuples

Note that we define suborders on tuples $\langle u_1,\ldots,u_k\rangle \prec^{k;\sharp} \langle v_1,\ldots,v_k\rangle$ ($\sharp \in \{\mathsf{w},\mathsf{p},\mathsf{l}\}$) as

$$u_1 \prec^{\sharp} v_1 \ \vee \ u_1 = v_1 \ \wedge \ \langle u_2,\ldots,u_k\rangle \prec^{k;\mathsf{kb}} \langle v_2,\ldots,v_k\rangle,$$

instead as

$$\exists i(1 \le i \le k)\Big[u_i \prec^{\sharp} v_i \ \wedge \ \forall j(1 \le j < i) u_j = v_j\Big].$$

As we showed, this is a purely technical choice to give reduction formulas uniform appearance.

Recall that $\langle u_1,\ldots,u_k\rangle \prec_n^{k;\sharp} \langle v_1,\ldots,v_k\rangle$ is *proper* if $u_1 \prec^{\sharp} v_1$. All reductions shown in Appendices K-O are reductions of *proper* orders between tuples of the same weight. This amounts to assume implicit side conditions $u_1 \prec^{\sharp} v_1$ in redexes (reduction targets).

Let $sum = \sum_{i=1}^{k} u_i^{\mathsf{w}} = \sum_{i=1}^{k} v_i^{\mathsf{w}}$.

K.1 Reduction of $\langle u_1,\ldots,u_k\rangle \prec_n^{k;\mathsf{w}} \langle v_1,\ldots,v_k\rangle$.

$$\bigvee_{\substack{n_1+n_2+n_3=n\\ n_1,n_3\ge 0, n_2\ge 1}} \langle u_1,\ldots,u_k\rangle \prec_{n_1}^{k;\mathsf{pl}} \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,u_1^{\mathsf{w}})} \prec_{n_2}^{k;\mathsf{w}} \bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum,v_1^{\mathsf{w}})} \prec_{n_3}^{k;\mathsf{pl}} \langle v_1,\ldots,v_k\rangle.$$

K.2 Reduction of $\langle u_1,\ldots,u_k\rangle \prec_n^{k;\mathsf{p}} \langle v_1,\ldots,v_k\rangle$.

Let $\mathsf{type}(u_1) = \alpha_p$ and $\mathsf{type}(v_1) = \alpha_{p'}$.

$$\bigvee_{\substack{n_1+n_2+n_3=n,\\ n_1,n_3\ge 0, n_2\ge 1}} \langle u_1,\ldots,u_k\rangle \prec_{n_1}^{k;\mathsf{l}} \bar{\mathbf{1}}^{k;\mathsf{p}}_{(sum,u_1^{\mathsf{w}},p)} \prec_{n_2}^{k;\mathsf{p}} \bar{\mathbf{0}}^{k;\mathsf{p}}_{(sum,v_1^{\mathsf{w}},p')} \prec_{n_3}^{k;\mathsf{l}} \langle v_1,\ldots,v_k\rangle.$$

K.3 Reduction of $\langle u_1,\ldots,u_k\rangle \prec_n^{k;\mathsf{l}} \langle v_1,\ldots,v_k\rangle$.

Let $rem = \sum_{i=2}^{k} u_i^{\mathsf{w}} = \sum_{i=2}^{k} v_i^{\mathsf{w}}$.

$$\bigvee_{\substack{n_1+n_2+n_3=n,\\ n_1,n_2\ge 0, n_3\ge 1}} \Big[\langle u_2,\ldots,u_k\rangle \prec_{n_1}^{k-1;\mathsf{kb}} \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(rem)} \ \wedge \ \bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(rem)} \prec_{n_2}^{k-1;\mathsf{kb}} \langle v_2,\ldots,v_k\rangle$$
$$\wedge \bigvee_{\substack{(m_1+1)(m_2-1)\ge(n_3-1)\\ m_1\ge 0, m_2>0, m_1,m_2<n_3}} \Big(\bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(rem)} \prec_{m_1}^{k-1;\mathsf{kb}} \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(rem)} \ \wedge \ u_1 \prec_{m_2}^{\mathsf{l}} v_1\Big)\Big],$$

L Reduction of Half Open Equalities on Tuples.

L.1 Reduction of $\bar{0}^{k;\mathsf{kb}}_{(sum)} = \langle u_1, \ldots, u_k \rangle$.

$$\bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum,\mathsf{MinCW}^k(sum))} = \langle u_1, \ldots, u_k \rangle.$$

L.2 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} = \langle u_1, \ldots, u_k \rangle$.

$$\mathsf{0}^{\mathsf{w}}_{(m)} = u_1 \wedge \bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(sum-m)} = \langle u_2, \ldots, u_k \rangle.$$

L.3 Reduction of $\bar{0}^{k;\mathsf{p}}_{(sum,m,p)} = \langle u_1, \ldots, u_k \rangle$.

$$\mathsf{0}^{\mathsf{p}}_{(m,p)} = u_1 \wedge \bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(sum-m)} = \langle u_2, \ldots, u_k \rangle$$

L.4 Reduction of $\bar{1}^{k;\mathsf{kb}}_{(sum)} = \langle u_1, \ldots, u_k \rangle$.

$$\bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,\mathsf{MaxCW}^k(sum))} = \langle u_1, \ldots, u_k \rangle.$$

L.5 Reduction of $\bar{1}^{k;\mathsf{w}}_{(sum,m)} = \langle u_1, \ldots, u_k \rangle$.

$$\mathsf{1}^{\mathsf{w}}_{(m)} = u_1 \wedge \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum-m)} = \langle u_2, \ldots, u_k \rangle$$

L.6 Reduction of $\bar{1}^{k;\mathsf{p}}_{(sum,m,p)} = \langle u_1, \ldots, u_k \rangle$.

$$\mathsf{1}^{\mathsf{w}}_{(m,p)} = u_1 \wedge \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum-m)} = \langle u_2, \ldots, u_k \rangle$$

M Reductions of Half Open Gap Orders on Tuples

M.1 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n} \langle u_1, \ldots, u_k \rangle$.

$$\bigvee_{\substack{n_1+n_2=n,\\ n_1\geq 1, n_2\geq 0}} \bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n_1} \bar{0}^{k;\mathsf{w}}_{(sum,u_1^{\mathsf{w}})} \prec^{k;\mathsf{pl}}_{n_2} \langle u_1, \ldots, u_k \rangle.$$

M.2 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{p}}_{n} \langle u_1, \ldots, u_k \rangle$.

$$\bigvee_{\substack{n_1(n_2+1)+n_3\geq n,\\ 0\leq n_1,n_2,n_3\leq n}} 0^{\mathsf{w}}_{(m)} \prec^{\mathsf{p}}_{n_1} u_1 \ \wedge\ \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{(sum-m)}$$
$$\wedge\ \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_3} \langle u_2, \ldots, u_k \rangle.$$

M.3 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{l}}_{n} \langle u_1, \ldots, u_k \rangle$.

$$\bigvee_{\substack{n_1(n_2+1)+n_3\geq n,\\ 0\leq n_1,n_2,n_3\leq n}} 0^{\mathsf{w}}_{(m)} \prec^{\mathsf{l}}_{n_1} u_1 \ \wedge\ \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{(sum-m)}$$
$$\wedge\ \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_3} \langle u_2, \ldots, u_k \rangle.$$

M.4 Reduction of $\bar{0}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{w}}_{n} \langle u_1, \ldots, u_k \rangle$.

$$\bigvee_{\substack{n_1+n_2=n,\\ n_1\geq 1, n_2\geq 0}} \bar{0}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{w}}_{n_1} \bar{0}^{k;\mathsf{w}}_{(sum,u_1^{\mathsf{w}})} \prec^{k;\mathsf{pl}}_{n_2} \langle u_1, \ldots, u_k \rangle.$$

M.5 Reduction of $\bar{0}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{p}}_{n} \langle u_1, \ldots, u_k \rangle$.

$$\bigvee_{\substack{n_1(n_2+1)+n_3\geq n,\\ 0\leq n_1,n_2,n_3\leq n}} 0^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{p}}_{n_1} u_1 \ \wedge\ \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{(sum-m)}$$
$$\wedge\ \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_3} \langle u_2, \ldots, u_k \rangle.$$

M.6 Reduction of $\bar{0}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{l}}_{n} \langle u_1, \ldots, u_k \rangle$.

$$\bigvee_{\substack{n_1(n_2+1)+n_3\geq n,\\ 0\leq n_1,n_2,n_3\leq n}} 0^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{l}}_{n_1} u_1 \ \wedge\ \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{(sum-m)}$$
$$\wedge\ \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_3} \langle u_2, \ldots, u_k \rangle.$$

M.7 Reduction of $\langle u_1, \ldots, u_k \rangle \prec^{k;\mathsf{w}}_n \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,m)}$.

$$\bigvee_{\substack{n_1+n_2=n,\\ n_1\geq 0, n_2\geq 1}} \langle u_1, \ldots, u_k \rangle \prec^{k;\mathsf{pl}}_{n_1} \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,u_1^{\mathsf{w}})} \prec^{k;\mathsf{w}}_{n_2} \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,m)}.$$

M.8 Reduction of $\langle u_1, \ldots, u_k \rangle \prec^{k;\mathsf{p}}_n \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,m)}$.

$$\begin{aligned}\bigvee_{\substack{n_1(n_2+1)+n_3\geq n,\\ 0\leq n_1,n_2,n_3\leq n}} u_1 \prec^{\mathsf{p}}_{n_1} \mathsf{1}^{\mathsf{w}}_{(m)} \ \wedge\ & \bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum-m)} \\ & \wedge\ \langle u_2, \ldots, u_k \rangle \prec^{k-1;\mathsf{kb}}_{n_3} \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum-m)}.\end{aligned}$$

M.9 Reduction of $\langle u_1, \ldots, u_k \rangle \prec^{k;\mathsf{l}}_n \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,m)}$.

$$\begin{aligned}\bigvee_{\substack{n_1(n_2+1)+n_3\geq n,\\ 0\leq n_1,n_2,n_3\leq n}} u_1 \prec^{\mathsf{l}}_{n_1} \mathsf{1}^{\mathsf{w}}_{(m)} \ \wedge\ & \bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum-m)} \\ & \wedge\ \langle u_2, \ldots, u_k \rangle \prec^{k-1;\mathsf{kb}}_{n_3} \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum-m)}.\end{aligned}$$

M.10 Reduction of $\langle u_1, \ldots, u_k \rangle \prec^{k;\mathsf{w}}_n \bar{\mathbf{1}}^{k;\mathsf{p}}_{(sum,m,p)}$.

$$\bigvee_{\substack{n_1+n_2=n,\\ n_1\geq 0, n_2\geq 1}} \langle u_1, \ldots, u_k \rangle \prec^{k;\mathsf{pl}}_{n_1} \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,u_1^{\mathsf{w}})} \prec^{k;\mathsf{w}}_{n_2} \bar{\mathbf{1}}^{k;\mathsf{p}}_{(sum,m,p)}.$$

M.11 Reduction of $\langle u_1, \ldots, u_k \rangle \prec^{k;\mathsf{p}}_n \bar{\mathbf{1}}^{k;\mathsf{p}}_{(sum,m,p)}$.

$$\begin{aligned}\bigvee_{\substack{n_1(n_2+1)+n_3\geq n,\\ 0\leq n_1,n_2,n_3\leq n}} u_1 \prec^{\mathsf{p}}_{n_1} \mathsf{1}^{\mathsf{p}}_{(m,p)} \ \wedge\ & \bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum-m)} \\ & \wedge\ \langle u_2, \ldots, u_k \rangle \prec^{k-1;\mathsf{kb}}_{n_3} \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum-m)}.\end{aligned}$$

M.12 Reduction of $\langle u_1, \ldots, u_k \rangle \prec^{k;\mathsf{l}}_n \bar{\mathbf{1}}^{k;\mathsf{p}}_{(sum,m,p)}$.

$$\begin{aligned}\bigvee_{\substack{n_1(n_2+1)+n_3\geq n,\\ 0\leq n_1,n_2,n_3\leq n}} u_1 \prec^{\mathsf{l}}_{n_1} \mathsf{1}^{\mathsf{p}}_{(m,p)} \ \wedge\ & \bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum-m)} \\ & \wedge\ \langle u_2, \ldots, u_k \rangle \prec^{k-1;\mathsf{kb}}_{n_3} \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum-m)}.\end{aligned}$$

N Reduction of Closed Equalities on Tuples

N.1 Reduction of $\bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum,m)} = \bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum',m')}$.

$$sum = sum' \wedge m = m'.$$

N.2 Reduction of $\bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum,m)} = \bar{\mathbf{0}}^{k;\mathsf{p}}_{(sum',m',p')}$.

$$sum = sum' \wedge m = m' \wedge \bigwedge_{i=p'+1}^{|\Sigma|} \neg\mathsf{Tree}^{(\alpha_i)}(m).$$

N.3 Reduction of $\bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum,m)} = \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum',m')}$.

$$sum = sum' \wedge m = m' \wedge \neg\mathsf{CNT}_1(m) \wedge \bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(sum-m)} = \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum'-m')}.$$

N.4 Reduction of $\bar{\mathbf{0}}^{k;\mathsf{w}}_{(sum,m)} = \bar{\mathbf{1}}^{k;\mathsf{p}}_{(sum',m',p')}$.

$$\begin{aligned} sum = sum' \wedge m = m' \wedge \bigwedge_{i=p'+1}^{|\Sigma|} \neg\mathsf{Tree}^{(\alpha_i)}(m) \wedge \\ \neg\mathsf{CNT}_1^{(\alpha_{p'})}(m) \wedge \bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(sum-m)} = \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum'-m')}. \end{aligned}$$

N.5 Reduction of $\bar{\mathbf{0}}^{k;\mathsf{p}}_{(sum,m,p)} = \bar{\mathbf{0}}^{k;\mathsf{p}}_{(sum',m',p')}$.

$$sum = sum' \wedge m = m' \wedge p = p'.$$

N.6 Reduction of $\bar{\mathbf{0}}^{k;\mathsf{p}}_{(sum,m,p)} = \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum',m')}$.

$$\begin{aligned} sum = sum' \wedge m = m' \wedge \bigwedge_{i=1}^{p-1} \neg\mathsf{Tree}^{(\alpha_i)}(m) \wedge \\ \neg\mathsf{CNT}_1^{(\alpha_p)}(m) \wedge \bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(sum-m)} = \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum'-m')}. \end{aligned}$$

N.7 Reduction of $\bar{\mathbf{0}}^{k;\mathsf{p}}_{(sum,m,p)} = \bar{\mathbf{1}}^{k;\mathsf{p}}_{(sum',m',p')}$.

$$\begin{aligned} sum = sum' \wedge m = m' \wedge p = p' \wedge \\ \neg\mathsf{CNT}_1^{(\alpha_p)}(m) \wedge \bar{\mathbf{0}}^{k-1;\mathsf{kb}}_{(sum-m)} = \bar{\mathbf{1}}^{k-1;\mathsf{kb}}_{(sum'-m')}. \end{aligned}$$

N.8 Reduction of $\bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum,m)} = \bar{\mathbf{1}}^{k;\mathsf{w}}_{(sum',m')}$.

$$sum = sum' \wedge m = m'.$$

N.9 Reduction of $\bar{1}^{k;\mathsf{w}}_{(sum,m)} = \bar{1}^{k;\mathsf{p}}_{(sum',m',p')}$.

$$sum = sum' \wedge m = m' \wedge \bigwedge_{i=1}^{p'-1} \neg \mathsf{Tree}^{(\alpha_i)}(m).$$

N.10 Reduction of $\bar{1}^{k;\mathsf{p}}_{(sum,m,p)} = \bar{1}^{k;\mathsf{p}}_{(sum',m',p')}$.

$$sum = sum' \wedge m = m' \wedge p = p'.$$

O Reductions of Closed Gap Orders on Tuples

As relations between tuples are syntactic sugar, we only need a small portion of reductions in the following list.

O.1 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n} \bar{0}^{k;\mathsf{w}}_{(sum,m')}$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 0, n_2\geq 1}} \bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{pl}}_{n_1} \bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n_2} \bar{0}^{k;\mathsf{w}}_{(sum,m')}.$$

O.2 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{p}}_{n} \bar{0}^{k;\mathsf{w}}_{(sum,m')}$.

$$\mathsf{false}.$$

O.3 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{l}}_{n} \bar{0}^{k;\mathsf{w}}_{(sum,m')}$.

$$\mathsf{false}.$$

O.4 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{n_1+n_2+n_3=n\\ n_1,n_3\geq 0, n_2\geq 1}} \bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{pl}}_{n_1} \bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n_2} \bar{0}^{k;\mathsf{w}}_{(sum,m')} \prec^{k;\mathsf{pl}}_{n_3} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')}.$$

O.5 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{p}}_{n} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{n_1(n_2+1)\geq n\\ n_1>0, n_2\geq 0, n_1,n_2\leq n}} \mathsf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{p}}_{n_1} \mathsf{0}^{\mathsf{p}}_{(m',p')} \wedge \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{(sum-m)}.$$

O.6 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{l}}_{n} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')}$.

$$\mathsf{false}.$$

O.7 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n} \bar{1}^{k;\mathsf{w}}_{(sum,m')}$.

Case $m = m'$.

$$\bigvee_{\substack{(n_1+1)(n_2+1)\geq n+1\\ 0\leq n_1,n_2\leq (n+1)}} \mathsf{0}^{\mathsf{w}}_{(m)} \prec^{\mathsf{w}}_{n_1} \mathsf{1}^{\mathsf{w}}_{(m)} \wedge \bar{0}^{k-1;\mathsf{kb}}_{sum-m} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{sum-m}.$$

Case $m < m'$.

$$\bigvee_{r=1}^{n-1} \Big[m \leq \mathsf{CW}^k_1(sum,m,m') < \ldots < \mathsf{CW}^k_r(sum,m,m') \leq m' \wedge$$

$$\bigvee_{\substack{\sum_{i=1}^r n_i = n+1\\ n_i\geq 1}} \bigwedge_{i=1}^{r} \bar{0}^{k;\mathsf{w}}_{(sum,\mathsf{CW}^k_i(sum,m,m'))} \prec^{k;\mathsf{w}}_{n_i-1} \bar{1}^{k;\mathsf{w}}_{(sum,\mathsf{CW}^k_i(sum,m,m'))} \Big].$$

O.8 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{p}}_{n} \bar{1}^{k;\mathsf{w}}_{(sum,m')}$.

$$\bigvee_{\substack{(n_1+1)(n_2+1)\geq n+1\\ n_1>0,n_2\geq 0,n_1,n_2\leq n+1}} 0^{\mathsf{w}}_{(m)} \prec^{\mathsf{p}}_{n_1} 1^{\mathsf{w}}_{(m')} \wedge \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{(sum-m)}.$$

O.9 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{l}}_{n} \bar{1}^{k;\mathsf{w}}_{(sum,m')}$.

$$\bigvee_{\substack{(n_1+1)(n_2+1)\geq n+1\\ n_1>0,n_2\geq 0,n_1,n_2\leq n+1}} 0^{\mathsf{w}}_{(m)} \prec^{\mathsf{l}}_{n_1} 1^{\mathsf{w}}_{(m')} \wedge \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{(sum-m)}.$$

O.10 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n} \bar{1}^{k;\mathsf{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{n_1+n_2+n_3+n_4=n\\ n_1,n_3,n_4\geq 0,n_2\geq 1}} \bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{pl}}_{n_1} \bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n_2} \bar{0}^{k;\mathsf{w}}_{(sum,m')} \prec^{k;\mathsf{pl}}_{n_3} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')} \prec^{k;\mathsf{pl}}_{n_4} \bar{1}^{k;\mathsf{p}}_{(sum,m',p')}.$$

O.11 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{p}}_{n} \bar{1}^{k;\mathsf{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{(n_1+1)(n_2+1)\geq n+1\\ n_1>0,n_2\geq 0,n_1,n_2\leq n+1}} 0^{\mathsf{w}}_{(m)} \prec^{\mathsf{p}}_{n_1} 1^{\mathsf{p}}_{(m',p')} \wedge \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{(sum-m)}.$$

O.12 Reduction of $\bar{0}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{l}}_{n} \bar{1}^{k;\mathsf{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{(n_1+1)(n_2+1)\geq n+1\\ n_1>0,n_2\geq 0,n_1,n_2\leq n+1}} 0^{\mathsf{w}}_{(m)} \prec^{\mathsf{l}}_{n_1} 1^{\mathsf{p}}_{(m',p')} \wedge \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{(sum-m)}.$$

O.13 Reduction of $\bar{0}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{w}}_{n} \bar{0}^{k;\mathsf{w}}_{(sum,m')}$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 0,n_2\geq 1}} \bar{0}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{pl}}_{n_1} \bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n_2} \bar{0}^{k;\mathsf{w}}_{(sum,m')}.$$

O.14 Reduction of $\bar{0}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{p}}_{n} \bar{0}^{k;\mathsf{w}}_{(sum,m')}$.

false.

O.15 Reduction of $\bar{0}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{l}}_{n} \bar{0}^{k;\mathsf{w}}_{(sum,m')}$.

false.

O.16 Reduction of $\bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m,p)} \prec^{k;\mathrm{w}}_{n} \bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{n_1+n_2+n_3+n_4=n\\ n_1,n_2,n_4\geq 0,n_3\geq 1}} \bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m,p)} \prec^{k;\mathrm{pl}}_{n_1} \bar{\mathbf{1}}^{k;\mathrm{p}}_{(sum,m,p)} \prec^{k;\mathrm{pl}}_{n_2} \bar{\mathbf{1}}^{k;\mathrm{w}}_{(sum,m)} \prec^{k;\mathrm{w}}_{n_3} \bar{\mathbf{0}}^{k;\mathrm{w}}_{(sum,m')} \prec^{k;\mathrm{pl}}_{n_4} \bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m',p')}.$$

O.17 Reduction of $\bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m,p)} \prec^{k;\mathrm{p}}_{n} \bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{n_1(n_2+1)\geq n\\ n_1>0,n_2\geq 0,n_1,n_2\leq n}} 0^{\mathrm{p}}_{(m,p)} \prec^{\mathrm{p}}_{n_1} 0^{\mathrm{p}}_{(m',p')} \wedge \bar{\mathbf{0}}^{k-1;\mathrm{kb}}_{(sum-m)} \prec^{k-1;\mathrm{kb}}_{n_2} \bar{\mathbf{1}}^{k-1;\mathrm{kb}}_{(sum-m)}.$$

O.18 Reduction of $\bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m,p)} \prec^{k;\mathrm{l}}_{n} \bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m',p')}$.

false.

O.19 Reduction of $\bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m,p)} \prec^{k;\mathrm{w}}_{n} \bar{\mathbf{1}}^{k;\mathrm{w}}_{(sum,m')}$.

$$\bigvee_{\substack{n_1+n_2+n_3+n_4=n\\ n_1,n_2,n_4\geq 0,n_3\geq 1}} \bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m,p)} \prec^{k;\mathrm{pl}}_{n_1} \bar{\mathbf{1}}^{k;\mathrm{p}}_{(sum,m,p)} \prec^{k;\mathrm{w}}_{n_2} \bar{\mathbf{1}}^{k;\mathrm{w}}_{(sum,m)} \prec^{k;\mathrm{pl}}_{n_3} \bar{\mathbf{0}}^{k;\mathrm{w}}_{(sum,m')} \prec^{k;\mathrm{pl}}_{n_4} \bar{\mathbf{1}}^{k;\mathrm{w}}_{(sum,m')}.$$

O.20 Reduction of $\bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m,p)} \prec^{k;\mathrm{p}}_{n} \bar{\mathbf{1}}^{k;\mathrm{w}}_{(sum,m')}$.

$$\bigvee_{\substack{(n_1+1)(n_2+1)\geq n+1\\ n_1>0,n_2\geq 0,n_1,n_2\leq n+1}} 0^{\mathrm{p}}_{(m,p)} \prec^{\mathrm{p}}_{n_1} 1^{\mathrm{w}}_{(m')} \wedge \bar{\mathbf{0}}^{k-1;\mathrm{kb}}_{(sum-m)} \prec^{k-1;\mathrm{kb}}_{n_2} \bar{\mathbf{1}}^{k-1;\mathrm{kb}}_{(sum-m)}.$$

O.21 Reduction of $\bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m,p)} \prec^{k;\mathrm{l}}_{n} \bar{\mathbf{1}}^{k;\mathrm{w}}_{(sum,m')}$.

$$\bigvee_{\substack{(n_1+1)(n_2+1)\geq n+1\\ n_1>0,n_2\geq 0,n_1,n_2\leq n+1}} 0^{\mathrm{p}}_{(m,p)} \prec^{\mathrm{l}}_{n_1} 1^{\mathrm{w}}_{(m')} \wedge \bar{\mathbf{0}}^{k-1;\mathrm{kb}}_{(sum-m)} \prec^{k-1;\mathrm{kb}}_{n_2} \bar{\mathbf{1}}^{k-1;\mathrm{kb}}_{(sum-m)}.$$

O.22 Reduction of $\bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m,p)} \prec^{k;\mathrm{w}}_{n} \bar{\mathbf{1}}^{k;\mathrm{p}}_{(sum,m',p')}$.

$$\begin{aligned}\bigvee_{\substack{n_1+n_2+n_3+n_4+n+5=n\\ n_1,n_2,n_4,n_5\geq 0,n_3\geq 1}} & \bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m,p)} \prec^{k;\mathrm{pl}}_{n_1} \bar{\mathbf{1}}^{k;\mathrm{p}}_{(sum,m,p)} \prec^{k;\mathrm{pl}}_{n_2} \bar{\mathbf{1}}^{k;\mathrm{w}}_{(sum,m,p)} \\ & \prec^{k;\mathrm{w}}_{n_3} \bar{\mathbf{0}}^{k;\mathrm{w}}_{(sum,m')} \prec^{k;\mathrm{pl}}_{n_4} \bar{\mathbf{0}}^{k;\mathrm{p}}_{(sum,m',p')} \prec^{k;\mathrm{pl}}_{n_5} \bar{\mathbf{1}}^{k;\mathrm{p}}_{(sum,m',p')}.\end{aligned}$$

O.23 Reduction of $\bar{0}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{p}}_{n} \bar{1}^{k;\mathsf{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{(n_1+1)(n_2+1)\geq n+1\\ n_1>0,n_2\geq 0,n_1,n_2\leq n+1}} 0^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{p}}_{n_1} 1^{\mathsf{p}}_{(m',p')} \wedge \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{(sum-m)}.$$

O.24 Reduction of $\bar{0}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{l}}_{n} \bar{1}^{k;\mathsf{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{(n_1+1)(n_2+1)\geq n+1\\ n_1>0,n_2\geq 0,n_1,n_2\leq n+1}} 0^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{l}}_{n_1} 1^{\mathsf{p}}_{(m',p')} \wedge \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{(sum-m)}.$$

O.25 Reduction of $\bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n} \bar{0}^{k;\mathsf{w}}_{(sum,m')}$.

$$\bigvee_{r=1}^{n-1} \Big[m < \mathsf{CW}^k_1(sum,m,m') < \ldots < \mathsf{CW}^k_r(sum,m,m') < m' \wedge$$
$$\bigvee_{\substack{\sum_{i=1}^r n_i = n-1\\ n_i\geq 1}} \bigwedge_{i=1}^{r} \bar{0}^{k;\mathsf{w}}_{(sum,\mathsf{CW}^k_i(sum,m,m'))} \prec^{k;\mathsf{w}}_{n_i-1} \bar{1}^{k;\mathsf{w}}_{(sum,\mathsf{CW}^k_i(sum,m,m'))}\Big].$$

O.26 Reduction of $\bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{p}}_{n} \bar{0}^{k;\mathsf{w}}_{(sum,m')}$.

false.

O.27 Reduction of $\bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{l}}_{n} \bar{0}^{k;\mathsf{w}}_{(sum,m')}$.

false.

O.28 Reduction of $\bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{n_1+n_2+n_3=n\\ n_1\geq 1,n_2\geq 0}} \bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n_1} \bar{0}^{k;\mathsf{w}}_{(sum,m')} \prec^{k;\mathsf{pl}}_{n_2} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')}.$$

O.29 Reduction of $\bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{p}}_{n} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')}$.

false

O.30 Reduction of $\bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{l}}_{n} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')}$.

false.

O.31 Reduction of $\bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n} \bar{1}^{k;\mathsf{w}}_{(sum,m')}$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_2\geq 0, n_1\geq 1}} \bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n_1} \bar{0}^{k;\mathsf{w}}_{(sum,m')} \prec^{k;\mathsf{pl}}_{n_2} \bar{1}^{k;\mathsf{w}}_{(sum,m')}.$$

O.32 Reduction of $\bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{p}}_{n} \bar{1}^{k;\mathsf{w}}_{(sum,m')}$.

false

O.33 Reduction of $\bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{l}}_{n} \bar{1}^{k;\mathsf{w}}_{(sum,m')}$.

false.

O.34 Reduction of $\bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n} \bar{1}^{k;\mathsf{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{n_1+n_2+n_3=n\\ n_2,n_3\geq 0, n_1\geq 1}} \bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n_1} \bar{0}^{k;\mathsf{w}}_{(sum,m')} \prec^{k;\mathsf{pl}}_{n_2} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')}. \prec^{k;\mathsf{pl}}_{n_3} \bar{1}^{k;\mathsf{p}}_{(sum,m',p')}.$$

O.35 Reduction of $\bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{p}}_{n} \bar{1}^{k;\mathsf{p}}_{(sum,m',p')}$.

false.

O.36 Reduction of $\bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{l}}_{n} \bar{1}^{k;\mathsf{p}}_{(sum,m',p')}$.

false.

O.37 Reduction of $\bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{w}}_{n} \bar{0}^{k;\mathsf{w}}_{(sum,m')}$.

$$\bigvee_{\substack{n_1+n_2=n\\ n_1\geq 0, n_2\geq 1}} \bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{pl}}_{n_1} \bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n_2} \bar{0}^{k;\mathsf{w}}_{(sum,m')}.$$

O.38 Reduction of $\bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{p}}_{n} \bar{0}^{k;\mathsf{w}}_{(sum,m')}$.

false.

O.39 Reduction of $\bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{l}}_{n} \bar{0}^{k;\mathsf{w}}_{(sum,m')}$.

false.

O.40 Reduction of $\bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{w}}_{n} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{n_1+n_2+n_3=n\\ n_1,n_3\geq 0, n_2\geq 1}} \bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{pl}}_{n_1} \bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n_2} \bar{0}^{k;\mathsf{w}}_{(sum,m')} \prec^{k;\mathsf{pl}}_{n_3} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')}.$$

O.41 Reduction of $\bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{p}}_{n} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{(n_1-1)(n_2+1)\geq n-1\\ n_1>0,n_2\geq 0,n_1,n_2\leq n-1}} 0^{\mathsf{p}}_{(m)} \prec^{\mathsf{p}}_{n_1} 1^{\mathsf{p}}_{(m)} \wedge \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{(sum-m)}.$$

O.42 Reduction of $\bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{l}}_{n} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')}$.

false.

O.43 Reduction of $\bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{w}}_{n} \bar{1}^{k;\mathsf{w}}_{(sum,m')}$.

$$\bigvee_{\substack{n_1+n_2+n_3+n_4=n\\ n_1,n_3\geq 0,n_2\geq 1}} \bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{pl}}_{n_1} \bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n_2} \bar{0}^{k;\mathsf{w}}_{(sum,m')} \prec^{k;\mathsf{pl}}_{n_3} \bar{1}^{k;\mathsf{w}}_{(sum,m')}$$

O.44 Reduction of $\bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{p}}_{n} \bar{1}^{k;\mathsf{w}}_{(sum,m')}$.

$$\bigvee_{\substack{(n_1)(n_2+1)\geq n\\ n_1>0,n_2\geq 0,n_1,n_2\leq n}} 1^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{p}}_{n_1} 1^{\mathsf{w}}_{(m')} \wedge \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{(sum-m)}.$$

O.45 Reduction of $\bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{l}}_{n} \bar{1}^{k;\mathsf{w}}_{(sum,m')}$.

false.

O.46 Reduction of $\bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{w}}_{n} \bar{1}^{k;\mathsf{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{n_1+n_2+n_3+n_4=n\\ n_1,n_3,n_4\geq 0,n_2\geq 1}} \bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{pl}}_{n_1} \bar{1}^{k;\mathsf{w}}_{(sum,m)} \prec^{k;\mathsf{w}}_{n_2} \bar{0}^{k;\mathsf{w}}_{(sum,m')} \prec^{k;\mathsf{pl}}_{n_3} \bar{0}^{k;\mathsf{p}}_{(sum,m',p')} \prec^{k;\mathsf{pl}}_{n_4} \bar{1}^{k;\mathsf{p}}_{(sum,m',p')}.$$

O.47 Reduction of $\bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{p}}_{n} \bar{1}^{k;\mathsf{p}}_{(sum,m',p')}$.

$$\bigvee_{\substack{n_1(n_2+1)\geq n\\ n_1>0,n_2\geq 0,n_1,n_2\leq n}} 1^{\mathsf{p}}_{(m,p)} \prec^{\mathsf{p}}_{n_1} 1^{\mathsf{p}}_{(m',p')} \wedge \bar{0}^{k-1;\mathsf{kb}}_{(sum-m)} \prec^{k-1;\mathsf{kb}}_{n_2} \bar{1}^{k-1;\mathsf{kb}}_{(sum-m)}.$$

O.48 Reduction of $\bar{1}^{k;\mathsf{p}}_{(sum,m,p)} \prec^{k;\mathsf{l}}_{n} \bar{1}^{k;\mathsf{p}}_{(sum,m',p')}$.

false.

P FAQ

This section contains frequently asked questions and answers. The concerns have been addressed in the previous sections. We collect them here just for ease of reference.

P.1 Question:

If a language with the unary function f of weight 0, how to do *depth-reduction* for literals of the forms $Lx \prec^{\mathsf{l}}_n x$ and $Lx \leqq^{\mathsf{l}}_n x$ where L is a proper non-empty selector sequence?

Answer:

If t is a boundary term or an ordinary term not containing x, then the result follows directly from Lemma 4. Suppose otherwise t must be in the form Lx where L is non-empty block of selectors. We have two cases depending on whether Lx is at the left-hand side or the right-hand side of $\star$.

As x is α_p-typed, without loss of generality we may assume $L \equiv L'\mathsf{s}_1$ where s_1 is the 1^{st} α_p-selector.

Case 1. $x \star Lx$**, for** $\star \in \{\prec^{\mathsf{w}}_n, \prec^{\mathsf{p}}_n, \prec^{\mathsf{l}}_n, \leqq^{\mathsf{w}}_n, \leqq^{\mathsf{p}}_n, \leqq^{\mathsf{l}}_n\}$**.** Then $x \star Lx$ simplifies to false. Since L is non-empty and constraints are type-complete, Lx is a proper subterm of x. For $n = 0$, $x \star Lx$ does not hold as $x \neq Lx$. For $n > 1$, $x \star Lx$ violates the subterm property of KBO [1].

Case 2. $Lx \star x$**, for** $\star \in \{\prec^{\mathsf{w}}_n, \prec^{\mathsf{p}}_n, \prec^{\mathsf{l}}_n, \leqq^{\mathsf{w}}_n, \leqq^{\mathsf{p}}_n, \leqq^{\mathsf{l}}_n\}$**.** We assume $n > 1$ as $x \neq Lx$. If $\star$ is $\prec^{\mathsf{w}}_n$ (resp. $\prec^{\mathsf{p}}_n$), Reduction F.1 (resp. F.2) puts x and Lx in separate literals, and hence the problem goes away. Similarly if $\star$ is $\leqq^{\mathsf{w}}_n$ or $\leqq^{\mathsf{p}}_n$. The last two cases are $Lx \prec^{\mathsf{l}}_n x$ and $Lx \leqq^{\mathsf{l}}_n x$, which are only possible when the language contains the unary function f of weight 0, and L must be a non-empty block of s_f (where s_f is the corresponding selector of f). (If L is not of the form $\{\mathsf{s}_f\}^+$, then $\{Lx\}^{\mathsf{w}} < x^{\mathsf{w}}$ and hence $Lx \prec^{\mathsf{l}}_n x$ simplifies to false.) We first consider $Lx \prec^{\mathsf{l}}_n x$, which is of the form

$$\mathsf{s}^i_f(x) \prec^{\mathsf{l}}_n x \tag{63}$$

where $n > 0$, s^i_f denote i time applications of s_f. First note that (63) certainly implies $\mathsf{Is}_f(\mathsf{s}^i_f(x))$. There are two subcases:

A. There is a non-atom term whose weight is x^{w} **and type is not** f**.** Let $g(t_1, \ldots, t_j)$ be such a term; i.e., $g \not\equiv f$ and $(g(t_1, \ldots, t_j))^{\mathsf{w}} = x^{\mathsf{w}}$.

1. If $\mathsf{s}^i_f(x)$ is $f^k(a)$ (where $k > 0$ and a is an atom), then x is $f^{i+k}(a)$. Since $(g(t_1, \ldots, t_j))^{\mathsf{w}} = x^{\mathsf{w}} = a^{\mathsf{w}}$, we have either $a \prec^{\mathsf{p}} g(t_1, \ldots, t_j)$ or $g(t_1, \ldots, t_j) \prec^{\mathsf{p}} a$.
   (a) $a \prec^{\mathsf{p}} g(t_1, \ldots, t_j)$. Then

$$f^k(a) \prec^{\mathsf{l}} f^k(g(t_1, \ldots, t_j)) \prec^{\mathsf{l}} \ldots \prec^{\mathsf{l}} f^k(g(f^l(t_1), \ldots, t_j)) \prec^{\mathsf{l}} \ldots \prec^{\mathsf{l}} f^{i+k}(a).$$

(b) $g(t_1,\ldots,t_j) \prec^{\mathsf{p}} a$. Then

$$f^k(a) \prec^{\mathsf{l}} f^{i+k}(g(t_1,\ldots,t_j)) \prec^{\mathsf{l}} \ldots \prec^{\mathsf{l}} f^{i+k}(g(f^l(t_1),\ldots,t_j)) \prec^{\mathsf{l}} \ldots \prec^{\mathsf{l}} f^{i+k}(a).$$

2. If $\mathsf{s}^i_f(x)$ is $f^k(h(t_1,\ldots,t_j))$ (where $k > 0$ and $h \not\equiv f$), then x is $f^{k+i}(h(t_1,\ldots,t_j))$. We have

$$f^k(h(t_1,\ldots,t_j)) \prec^{\mathsf{l}} \ldots \prec^{\mathsf{l}} f^k(h(f^l(t_1),\ldots,t_j)) \prec^{\mathsf{l}} \ldots \prec^{\mathsf{l}} f^{i+k}(a)$$

In all of cases, there are infinitely many terms in between $\mathsf{s}^i_f(x)$ and x, and hence (63) simplifies to $\mathsf{Is}_f(\mathsf{s}^i_f(x))$.

B. Any term of weight x^{w} consists of only f-terms (f-type terms) or atoms. The condition says that $\mathsf{s}^i_f(x)$ must be in the form $f^k(a)$ (for some atom a and $k > 0$) and then x is in the form $f^{k+i}(a)$. Note that in this situation, $f^k(a) \leqq^{\mathsf{l}}_r f^{k+i}(a)$ where r is i times the number of trees of weight x^{w} in the signature with f excluded. For example, let $\Sigma = \langle f, a, b\rangle$, we have $f(a) \leqq^{\mathsf{l}}_4 f^3(a)$, namely

$$f(a) \leqq^{\mathsf{l}} f(b) \leqq^{\mathsf{l}} f^2(a) \leqq^{\mathsf{l}} f^2(b) \leqq^{\mathsf{l}} f^3(a).$$

It is not hard to see that (63) simplifies to

$$\mathsf{Is}_f(\mathsf{s}^i_f(x)) \;\wedge\; \mathsf{CNT}_m(x^{\mathsf{w}}),$$

where $m = \lceil n/i \rceil - 1$ and $\mathsf{CNT}_m(x^{\mathsf{w}})$ is defined in the language with f excluded.

Similarly we can simplify $Lx \leqq^{\mathsf{l}}_n x$, which has the form

$$\mathsf{s}^i_f(x) \leqq^{\mathsf{l}}_n x \tag{64}$$

where $n > 0$, s^i_f denote i time applications of s_f. As before (64) certainly implies $\mathsf{Is}_f(\mathsf{s}^i_f(x))$. There are two subcases:

A. There is a non-atom term whose weight is x^{w} and type is not f. Since there are infinitely many terms in between $\mathsf{s}^i_f(x)$ and x, (64) simplifies to false.

B. Any term of weight x^{w} consists of only f-terms (f-type terms) or atoms. It is not hard to see that (64) simplifies to

$$\mathsf{Is}_f(\mathsf{s}^i_f(x)) \;\wedge\; \mathsf{CNT}_m(x^{\mathsf{w}}) \;\wedge\; \neg\mathsf{CNT}_{m+1}(x^{\mathsf{w}}) \;\wedge\; i \mid n,$$

where $m = \lceil n/i \rceil - 1$ and $\mathsf{CNT}_m(x^{\mathsf{w}})$ is defined in the language with f excluded.

At last we note that each of preconditions can be expressed in Presburger arithmetic; the condition for **A** is

$$\bigvee_{\alpha \in C \setminus \mathcal{A} \setminus \{f\}} \mathsf{Tree}^{\alpha}(x^{\mathsf{w}}), \tag{65}$$

and the condition for **B** is just the negation of (65).

P.2 Question:

In some proofs, you do a transformation by case distinction, but these cases depend on the *values* of variables. However, you have to give syntactic transformations which are equivalences for *all* values of the variables.

Answer:

Though we do the transformation at semantic level by case distinction, the corresponding syntactic transformations (that we really need) can be recovered straightforwardly as follows. Let $\varphi(t)$ be a formula in which term t occurs. Consider there are n cases in each of which under the precondition θ_i (expressible in our language), t is reduced to t_i ($1 \leq i \leq n$). Then $\varphi(t)$ can be rewritten as either one of

$$\bigwedge_i \Big(\theta_i \rightarrow \varphi(t_i)\Big), \qquad\qquad \bigvee_i \Big(\theta_i \wedge \varphi(t_i)\Big).$$

Succinctly, t can be written as either $\bigwedge_{i=1}^n \Big(\theta_i \rightarrow t_i\Big)$ or $\bigvee_{i=1}^n \Big(\theta_i \wedge t_i\Big)$, which is called a *conditional term*. For details, see pages 124-129 in

> "The Deductive Foundations of Computer Programming", by Zohar Manna and Richard Waldinger, Addison-Wesley, 1993

P.3 Question:

How do you eliminate equalities like

$$(\exists x : \mathsf{TA}) \Big[\, x = t \ \wedge\ \varphi(x) \,\Big] \tag{66}$$

where $\varphi(x)$ is an arbitrary conjunction of literals in $\mathscr{L}^{\mathbb{Z}}_{\mathsf{kb}^+}$.

Answer:

If t does not contain x, then (66) simplifies to $\varphi(t)$. Otherwise, we have three cases to consider.

Case 1. $t \equiv x$. Then $(\exists x)[x = x \wedge \varphi(x)]$ simplifies to $(\exists x)\varphi(x)$.

Case 2. $t \equiv Lx$ **where** L **is a non-empty selector block.** Then $(\exists x)(x = Lx \wedge \varphi(x))$ simplifies to false. Note that here we assume that terms like Lx are proper (see Def. 16)

Case 3. t **is a boundary term in which** $(L_i x)^{\mathsf{w}}$ **(**$i \leq n$**) occur (as parts of an integer function).** In this case (66) has the form

$$(\exists x\!:\!\mathsf{TA}) \Big[\, x = t[f((L_0 x)^{\mathsf{w}}, \ldots, (L_n x)^{\mathsf{w}}, x^{\mathsf{w}})] \ \wedge\ \varphi(x) \,\Big], \tag{67}$$

where $f((L_0 x)^{\mathsf{w}}, \ldots, (L_n x)^{\mathsf{w}}, x^{\mathsf{w}})$ denotes the outmost Presburger function containing $(L_i x)^{\mathsf{w}}$ ($i \leq n$) and x^{w}. We assume $t[f((L_0 x)^{\mathsf{w}}, \ldots, (L_n x)^{\mathsf{w}}, x^{\mathsf{w}})]$ is a well-defined boundary term (see Sec. E.15). Otherwise $t = \perp$ and hence (46) reduces to $\varphi(\perp)$. Without loss of generality, we assume $L_i x$ enumerate all selector terms containing x in (47) and they form a tree (with x being the root, $L_i x$ including all leaves). In addition, in this tree sibling nodes are coexistent, called *sibling-complete*. For example, in $\mathfrak{A}_{\mathsf{list}}$ if $\mathsf{car}(x)$ appear, we can assume that $\mathsf{cdr}(x)$ also appear. The induced tree is $\mathsf{cons}(\mathsf{car}(x), \mathsf{cdr}(x))$ where x is the root, $\mathsf{car}(x)$, $\mathsf{cdr}(x)$ are two leaves. Note that $L_i x$ also include intermediate nodes in the induced tree. In general, if we need $L'x$ to appear, we can just set

$$f((L_0x)^{\mathsf{w}}, \dots, (L_nx)^{\mathsf{w}}, (L'x)^{\mathsf{w}}, x^{\mathsf{w}}) := f((L_0x)^{\mathsf{w}}, \dots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}}) + (L'x)^{\mathsf{w}} - (L'x)^{\mathsf{w}}.$$

It is easily seen that (67) is equivalent to

$$(\exists x : \mathsf{TA}) \Big[x = t[f((L_0x)^{\mathsf{w}}, \dots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}})] \ \wedge\ \varphi(t[f((L_0x)^{\mathsf{w}}, \dots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}})]) \Big]. \tag{68}$$

Note that in (68) x only has one term occurrence, namely, as the left-hand side of

$$x = t[f((L_0x)^{\mathsf{w}}, \dots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}})]. \tag{69}$$

We claim that (68) is equivalent to

$$(\exists (L_0x)^{\mathsf{w}}, \dots (L_nx)^{\mathsf{w}}, x^{\mathsf{w}} : \mathbb{Z}) \Big[\varphi(t[f((L_0x)^{\mathsf{w}}, \dots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}})]) \wedge \Delta \Big], \tag{70}$$

where Δ is the constraint to describe the weight relations in the induced tree. More precisely, it satisfies the following conditions.

1. If L_ix is of type α_p and has immediate children $\mathsf{s}_1^{\alpha_p} L_ix, \dots, \mathsf{s}_k^{\alpha_p} L_ix$, then Δ includes

   $$(L_ix)^{\mathsf{w}} = \Big(\sum_{j=1}^{k} (\mathsf{s}_j^{\alpha_p} L_ix)^{\mathsf{w}} \Big) + \mathsf{W}(\alpha_p).$$

   In particular, it includes

   $$x^{\mathsf{w}} = \sum_{i \in I \subseteq \{0..n\}} (L_ix)^{\mathsf{w}} + c.$$

   where I is the index of all leaf nodes and c is the sum of the weight of intermediate nodes of the tree with root x.
2. For $i \in I$, Δ includes $\mathsf{Tree}((L_ix)^{\mathsf{w}})$; if L_ix is α-typed, Δ includes $\mathsf{Tree}^{\alpha}((L_ix)^{\mathsf{w}})$. These assertions constrains each leaf to have a well-defined weight.
3. Δ also includes constraints that require that $L_0x, \dots, L_nx, x$ form a boundary term. For example, suppose $t \equiv \mathsf{0}^{\mathsf{w}}_{(f((L_0x)^{\mathsf{w}}, \dots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}}))}$. Then each node L_ix must be the smallest (w.r.t. $<^{\mathsf{kb}}$) of weight $(L_ix)^{\mathsf{w}}$. If L_ix is of type α_p, then there exists no p' such that $p < p' \leq |\Sigma|$ and $\mathsf{Tree}^{\alpha_{p'}}((L_ix)^{\mathsf{w}})$ holds. This constraint is formally expressed by

   $$\bigwedge_{p < p' \leq |\Sigma|} \neg \mathsf{Tree}^{\alpha_{p'}}((L_ix)^{\mathsf{w}}).$$

A similar constraint is added for x. Besides that, if $L_i x$ has k siblings, namely, in the order from left to right, $L_{i_1}x, \ldots, L_{i_k}x$ (where $L_{i_j} \equiv \mathsf{s}_j^{\alpha_p}$ for $1 \le j \le k$), we require that $\langle L_{i_1}x, \ldots, L_{i_k}\rangle$ be the smallest k-tuple (w.r.t. $\prec^{k;\mathsf{kb}}$) of weight $n = \sum_{j=1}^{k}(L_{i_j}x)^{\mathsf{w}}$. This constraint is formally expressed by

$$\bigwedge_{j=1}^{k}(L_{i_j}x)^{\mathsf{w}} = \mathsf{SCW}_j^k(n),$$

where $\mathsf{SCW}_j^k(n)$ returns the weight of the j^{th} component of the smallest k-tuple (w.r.t. $\prec^{k;\mathsf{kb}}$) of weight n (see Sec. E.11). Similarly we obtain constraints for

$$t \equiv 0^{\mathsf{p}}_{(f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},x^{\mathsf{w}}),\_)}, \quad t \equiv 1^{\mathsf{w}}_{(f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},x^{\mathsf{w}}))}, \quad t \equiv 1^{\mathsf{p}}_{(f((L_0x)^{\mathsf{w}},\ldots,(L_nx)^{\mathsf{w}},x^{\mathsf{w}}),\_)}.$$

Notice that here $(L_ix)^{\mathsf{w}}$, x^{w} serve as purely syntactical variables. A variable renaming can bring (70) to

$$(\exists z_0, \ldots z_n, z\!:\!\mathbb{Z})\Big[\,\varphi'(t[f(z0, \ldots, z_n, z)]) \;\wedge\; \varDelta(z_0, \ldots, z_n, z)\,\Big]. \tag{71}$$

We show the equivalence between (68) and (70). First it is obvious that (68) implies (70) because $\varDelta$ describes weight constraints on the boundary term tree rooted at x. To show that (70) implies (68), it suffices to show there exists x such that (69) holds. Again because of $\varDelta$, $(L_0x)^{\mathsf{w}}, \ldots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}}$ satisfies weight constraints of a boundary term (tree) and $f((L_0x)^{\mathsf{w}}, \ldots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}})$ makes t well-defined. Therefore $(L_0x)^{\mathsf{w}}, \ldots, (L_nx)^{\mathsf{w}}, x^{\mathsf{w}}$ can be "realized" by a TA-term x which is a boundary term. Intuitively, when a term x occurs inside the weight function, its term properties is deprived.

We like to point out two things in this elimination by substitution. First the substitution from (67) to (68) may put a boundary term inside a selector term. In fact, it is only the substitution that makes a boundary term appear inside a selector. However, Lemma 2 eliminates this superficial complication. Second, the substitution obviously does not increase the number of open order literals.