Unifying the Knuth-Bendix, Recursive Path and Polynomial Orders

Akihisa Yamada
Graduate School of Information Science
Nagoya University
Furo-cho, Chikusa-ku, Nagoya 4648603 Japan
ayamada@sakabe.i.is.nagoya-u.ac.jp

Keiichirou Kusakari
Graduate School of Information Science
Nagoya University
Furo-cho, Chikusa-ku, Nagoya 4648603 Japan
kusakari@is.nagoya-u.ac.jp

Toshiki Sakabe
Graduate School of Information Science
Nagoya University
Furo-cho, Chikusa-ku, Nagoya 4648603 Japan
sakabe@is.nagoya-u.ac.jp

ABSTRACT

We introduce a simplification order called the weighted path order (WPO). WPO compares weights of terms as in the Knuth-Bendix order (KBO), while WPO allows weights to be computed by an arbitrary interpretation which is weakly monotone and weakly simple. We investigate summations, polynomials and maximums for such interpretations. We show that KBO is a restricted case of WPO induced by summations, the polynomial order (POLO) is subsumed by WPO induced by polynomials, and the lexicographic path order (LPO) is a restricted case of WPO induced by maximums. By combining these interpretations, we obtain an instance of WPO that unifies KBO, LPO and POLO. We also present SMT encodings of our orders, as well as incorporating them in the dependency pair framework.

Categories and Subject Descriptors

D.2.4 [**Software Engineering**]: Software/Program Verification; F.4.2 [**Mathematical Logic and Formal Languages**]: Grammars and Other Rewriting Systems

General Terms

Languages, Theory, Verification

Keywords

Term rewriting, Simplification order, Termination

1. INTRODUCTION

Proving *termination* is one of the most important task in program verification and automated theorem proving, where a *simplification order* plays a fundamental role. A number of simplification orders have been proposed, and their efficient implementations are proposed during the recent developments of automatic termination provers (e.g. AProVE[1] and $\mathsf{T_TT_2}$[2]). Below we recall the history of simplification orders.

One of the most well-known (classes of) simplification orders is the *lexicographic path order (LPO)* of Kamin and Lévy [14], a variant of the *recursive path order (RPO)* of Dershowitz [7]. LPO is unified with RPO using *status* [22]. Recently, an efficient implementation using a SAT solver for proving termination with RPO with status has been proposed in [5].

The *Knuth-Bendix order (KBO)* [15] is the most historical simplification order. KBO has become a practical alternative in automatic termination checking since Korovin and Voronkov [18] discovered a polynomial-time algorithm for termination proof with KBO, and Zankl et al. [30] proposed an efficient implementation using SAT/SMT solvers. However, KBO is disadvantageous compared to LPO when *duplicating* rules (where a variable occurs more often in the right hand side than in the left hand side) are considered. Actually, no duplicating rule can be oriented by KBO. To overcome this disadvantage, Middeldorp and Zantema [24] proposed the *generalized KBO (GKBO)*, which extends weights over algebras that are weakly monotone and *strictly simple*: $f(\dots,a,\dots) > a$. However, the strict simplicity condition is so restrictive that GKBO does not exactly subsume the standard KBO. Ludwig and Waldmann proposed another extension of KBO called the *transfinite KBO (TKBO)* [23, 19, 28], which extends the weight function to allow linear polynomials (on ordinals). However, proving termination with TKBO involves satisfiability problem of non-linear arithmetic which is undecidable in general. Moreover, TKBO still does not encompass LPO.

The *polynomial order (POLO)* of Lankford [21] interprets each function symbol by a (strictly monotone) polynomial. Zantema [32] extended the method over algebras that are weakly monotone and *weakly* simple: $f(\dots,a,\dots) \geq a$, and suggested combining the "max" operator with polynomial in-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
PPDP '13, September 16 - 18 2013, Madrid, Spain
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2154-9/13/09 ...$15.00.
http://dx.doi.org/10.1145/2505879.2505885.

[1] `http://aprove.informatik.rwth-aachen.de/`.
[2] `http://cl-informatik.uibk.ac.at/software/ttt2/`.

terpretations (*max-polynomials* in terms of [10]). Fuhs et al. propose an efficient SAT encoding of POLO in [9], and (general version of) POLO with max in [10]. Recently, Bofill et al. [4] proposed *RPOLO*, which unifies standard POLO and RPO by choosing either *RPO-like* or *POLO-like* comparison depending on the root symbols. In RPOLO, POLO-symbols are not given precedence and considered to have minimum precedence when compared with RPO-symbols.

These orders in the literature require different correctness proofs and different implementations. In this paper, we extract the underlying essence of existing techniques, and introduce a general simplification order called the *weighted path order (WPO)* and present its implementation via SMT encoding. WPO does not only encompass the three major orders (i.e. KBO, LPO and POLO), but also significantly enhances them as we verify through examples and experiments.

In contrast to RPOLO, WPO does not divide symbols to RPO-like and POLO-like ones, but applies both style of comparison for every symbols. This goal is achieved by further generalizing GKBO. More precisely, by merging the definition of GKBO and LPO, we extend weights over algebras that are weakly monotone and *weakly* simple. The extension admits arbitrary monotone polynomial interpretations which are only weakly simple in general, and more notably admits max-polynomials. We show that the extension is significant both theoretically and practically. In particular, we investigate the following instances of WPO characterized by how weights are computed:

- WPO($\mathcal{S}um$) which uses summations for weight computation. KBO can be obtained as a restricted case of WPO($\mathcal{S}um$), where the *admissibility* constraint is enforced, and weights of constants must be greater than 0. WPO($\mathcal{S}um$) is free from these restrictions, and each extension strictly increases the power of the order.

- WPO($\mathcal{P}ol$) which uses monotone polynomial interpretations for weight computation. Obviously, POLO is subsumed by WPO($\mathcal{P}ol$). TKBO can be obtained as a restricted case of WPO($\mathcal{P}ol$), where interpretations are linear polynomials, the admissibility is enforced, and interpretations of constants are greater than 0.

- WPO($\mathcal{M}ax$) which uses maximums for weight computation. LPO can be obtained as a restricted case of WPO($\mathcal{M}ax$), where the weights of all symbols are fixed to 0. In order to keep the presentation simple, we omit status and only consider LPO. Nonetheless, it is easy to extend this result for RPO with status.

- WPO($\mathcal{MP}ol$) which combines polynomial and maximum for interpretation, and its variant WPO($\mathcal{MS}um$) whose coefficients are fixed to 1. WPO($\mathcal{MS}um$) generalizes KBO and LPO, and WPO($\mathcal{MP}ol$) moreover subsumes POLO (with max).

We present SMT encoding techniques for these instances of WPO. In particular, orientability problem of WPO($\mathcal{S}um$), WPO($\mathcal{M}ax$) and WPO($\mathcal{MS}um$) are reduced to a satisfiability problem of linear arithmetic, which is decidable.

We also show that WPO is not subsumed by RPOLO. Moreover in practice, WPO shows advantage on the problems from the Termination Problem Data Base (TPDB [27]), while (the first-order version of) RPOLO does not [4].

The remainder of this paper is organized as follows. We briefly recall the basics of term rewriting and existing simplification orders in Section 2. In Section 3, we present the definition of WPO and prove that WPO is a simplification order in an abstract setting. Several instances of WPO are defined and their relationships between existing orders are discussed in Section 4. We present SMT encoding techniques of our orders in Section 5. Then in Section 6 we integrate our orders in the DP framework [1, 12, 11]. These orders are experimented in comparison with existing orders in Section 8. We conclude in Section 9.

2. PRELIMINARIES

Term rewrite systems (TRSs) model first-order functional programs. We refer readers to e.g. [26] for details of rewriting, and only briefly recall some important notions needed in this paper. A *signature* $\mathcal{F}$ is a finite set of function symbols associated with arity. The set of n-ary symbols is denoted by $\mathcal{F}_n$. A *term* is either a variable $x \in \mathcal{V}$ or in form $f(s_1, \dots, s_n)$ where $f \in \mathcal{F}_n$ and each s_i is a term. Throughout the paper, we abbreviate a sequence $a_1, \dots, a_n$ by $\overline{a_n}$. The set of terms constructed from $\mathcal{F}$ and $\mathcal{V}$ is denoted by $\mathcal{T}(\mathcal{F}, \mathcal{V})$. The set of variables occurring in a term s is denoted by $\mathcal{V}ar(s)$, and the number of occurrences of a variable x is denoted by $|s|_x$.

A TRS is a set $\mathcal{R}$ of pairs of terms called *rewrite rules.* A rewrite rule, written $l \to r$ where $\mathcal{V}ar(l) \supseteq \mathcal{V}ar(r)$, indicates that an instance of l should be rewritten to corresponding instance of r. The *rewrite relation* $\to_\mathcal{R}$ induced by $\mathcal{R}$ is the monotonic stable closure of $\mathcal{R}$, where a relation $\sqsupset$ on terms is *monotonic* iff $s \sqsupset t$ implies $f(\dots, s, \dots) \sqsupset f(\dots, t, \dots)$, and *stable* iff $s \sqsupset t$ implies $s\sigma \sqsupset t\sigma$ for every substitution σ. A TRS $\mathcal{R}$ is *terminating* iff no infinite rewrite sequence $s_1 \to_\mathcal{R} s_2 \to_\mathcal{R} \dots$ exists.

A *reduction order* is a well-founded order which is monotonic and stable. It is easy to see that a TRS $\mathcal{R}$ is terminating iff $\mathcal{R}$ is *oriented* by a reduction order $\succ$; i.e., $\mathcal{R} \subseteq \succ$. A *simplification order* is a strict order $\succ$ on terms, which is monotonic and stable and satisfies *subterm property*: $f(\dots, s, \dots) \succ s$. For finite signatures, it is well-known that a simplification order is a reduction order [7].

In remainder of this section, we recall definitions of existing simplification orders.

2.1 Lexicographic Path Order

LPO [14] is induced by a strict order $>_\mathcal{F}$ on $\mathcal{F}$ called a *precedence.*

Definition 1. For a precedence $>_\mathcal{F}$, the *lexicographic path order* $\succ_{\mathrm{LPO}}$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ is recursively defined as follows: $s = f(\overline{s_n}) \succ_{\mathrm{LPO}} t$ iff

(a) $\exists i \in \{1, \dots, n\}.\ s_i \succeq_{\mathrm{LPO}} t$, or

(b) $t = g(\overline{t_m})$, $\forall j \in \{1, \dots, m\}.\ s \succ_{\mathrm{LPO}} t_j$ and either

  i. $f >_\mathcal{F} g$, or

  ii. $f = g$ and $[\overline{s_n}] \succ^{\mathsf{lex}}_{\mathrm{LPO}} [\overline{t_m}]$.

THEOREM 1. *$\succ_{\mathrm{LPO}}$ is a simplification order.* □

2.2 Knuth-Bendix Order

KBO [15] is induced by a precedence $>_\mathcal{F}$ and a *weight function* (w, w_0), where $w : \mathcal{F} \to \mathbb{N}$ and $w_0 \in \mathbb{N}$ s.t. $w(c) \geq w_0$

for every constant $c \in \mathcal{F}_0$. The weight $w(s)$ of a term s is defined as follows:

$$w(s) := \begin{cases} w_0 & \text{if } s \in \mathcal{V} \\ w(f) + \sum_{i=1}^{n} w(s_i) & \text{if } s = f(\overline{s_n}) \end{cases}$$

w is *admissible* for $>_\mathcal{F}$ if every unary symbol $f \in \mathcal{F}_1$ with $w(f) = 0$ is maximum w.r.t. $>_\mathcal{F}$.

Definition 2. For a precedence $>_\mathcal{F}$ and a weight function (w, w_0), the *Knuth-Bendix order* on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ is recursively defined as follows: $s = f(\overline{s_n}) \succ_{\text{KBO}} t$ iff $|s|_x \geq |t|_x$ for all $x \in \mathcal{V}$ and either

1. $w(s) > w(t)$, or
2. $w(s) = w(t)$ and
   (a) $s = f^k(t)$ and $t \in \mathcal{V}$ for some $k > 0$, or
   (b) $t = g(\overline{t_m})$ and
      i. $f >_\mathcal{F} g$, or
      ii. $f = g$ and $[\overline{s_n}] \succ_{\text{KBO}}^{\text{lex}} [\overline{t_m}]$.

Here we follow [30], and the range of w is restricted to $\mathbb{N}$. According to [18], this does not decrease the power of KBO for finite TRSs. Note that we do not assume $w_0 > 0$ in the definition. This assumption, together with the admissibility is required for KBO to be a simplification order. The following result is well-known, see e.g. [2, Theorem 5.4.20] for details.

THEOREM 2. *If $w_0 > 0$ and w is admissible for $>_\mathcal{F}$, then $\succ_{\text{KBO}}$ is a simplification order.* □

The transfinite KBO (TKBO) [23, 19, 28] extends KBO by introducing a *subterm coefficient function sc*, that assigns a positive integer[3] $sc(f, i)$ to each $f \in \mathcal{F}_n$ and $i \in \{1, \ldots, n\}$. For a weight function (w, w_0) and a subterm coefficient function sc, $w(s)$ is refined as follows:

$$w(s) := \begin{cases} w_0 & \text{if } s \in \mathcal{V} \\ w(f) + \sum_{i=1}^{n} sc(f,i) \cdot w(s_i) & \text{if } s = f(\overline{s_n}) \end{cases}$$

The *variable coefficient* $\mathsf{vc}(x, s)$ of x in s is defined recursively as follows:

$$\mathsf{vc}(x, s) := \begin{cases} 1 & \text{if } x = s \\ 0 & \text{if } x \neq y \in \mathcal{V} \\ \sum_{i=1}^{n} sc(f,i) \cdot \mathsf{vc}(x, s_i) & \text{if } s = f(\overline{s_n}) \end{cases}$$

Then the order $\succ_{\text{TKBO}}$ is obtained from Definition 2 by replacing $|\cdot|_x$ by $\mathsf{vc}(x, \cdot)$ and $w(\cdot)$ by refined ones.

THEOREM 3. *[23] If $w_0 > 0$ and w is admissible for $>_\mathcal{F}$, then $\succ_{\text{TKBO}}$ is a simplification order.* □

[3]We do not use *transfinite* coefficients, since they do not add power when finite TRSs are considered [28].

2.3 Interpretation Method

We follow the abstract definition of [32]. A *well-founded $\mathcal{F}$-algebra $\mathcal{A}$* consists of a *carrier set* A, a well-founded partial order $>$ on A and an *interpretation* $f_\mathcal{A} : A^n \to A$ for each $f \in \mathcal{F}_n$. $\mathcal{A}$ is *strictly* (resp. *weakly*) *monotone* iff $a > b$ implies $f_\mathcal{A}(\ldots, a, \ldots) >$(resp. $\geq$) $f_\mathcal{A}(\ldots, b, \ldots)$, and *strictly* (resp. *weakly*) *simple* iff $f_\mathcal{A}(\ldots, a, \ldots) >$(resp. $\geq$) a for all interpretations $f_\mathcal{A}$. For $\sqsupset$ denoting $>$ or $\geq$, the relation $\sqsupset_\mathcal{A}$ on terms is defined as follows: $s \sqsupset_\mathcal{A} t$ iff $\widehat{\alpha}(s) \sqsupset \widehat{\alpha}(t)$ holds for all assignments $\alpha : \mathcal{V} \to A$ and its homomorphic extension $\widehat{\alpha}$.

THEOREM 4. *[32] A TRS $\mathcal{R}$ is terminating if $\mathcal{R}$ is oriented by $>_\mathcal{A}$ for some weakly monotone and weakly simple well-founded $\mathcal{F}$-algebra $\mathcal{A}$.* □

A *polynomial interpretation $\mathcal{P}ol$* interprets each function symbol $f \in \mathcal{F}$ as a monotone polynomial $f_{\mathcal{P}ol}$ over $\{a \in \mathbb{N} \mid a \geq w_0\}$ for some $w_0 \in \mathbb{N}$. Soundness of polynomial interpretation method can be obtained as corollary to the above theorem:

COROLLARY 1. *[21] A TRS $\mathcal{R}$ is terminating if $\mathcal{R}$ is oriented by $>_{\mathcal{P}ol}$ for some polynomial interpretation $\mathcal{P}ol$.* □

2.4 Generalized Knuth-Bendix Order

GKBO [24] uses a weakly monotone and *strictly* simple algebra for weight computation. In the following version of GKBO, we only consider the lexicographic extension.

Definition 3. For a precedence $>_\mathcal{F}$ and a well-founded $\mathcal{F}$-algebra $\mathcal{A}$, the *generalized Knuth-Bendix order* $\succ_{\text{GKBO}}$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ is recursively defined as follows: $s = f(\overline{s_n}) \succ_{\text{GKBO}} t$ iff

1. $s >_\mathcal{A} t$, or
2. $s \geq_\mathcal{A} t = g(\overline{t_m})$ and either
   i. $f >_\mathcal{F} g$, or
   ii. $f = g$ and $[\overline{s_n}] \succ_{\text{GKBO}}^{\text{lex}} [\overline{t_m}]$.

THEOREM 5. *[24] If $\mathcal{A}$ is weakly monotone and strictly simple, then $\succ_{\text{GKBO}}$ is a simplification order.* □

3. WEIGHTED PATH ORDER

Note that GKBO does not exactly subsume KBO, since the unary function of weight 0 is only weakly simple. In this section, we introduce an abstract order that further generalizes GKBO to admit algebras that are only weakly simple. The key idea is to merge the definitions of GKBO and LPO. The following notion is used to reduce recursive checks.

Definition 4. Let $\mathcal{A}$ be a well-founded algebra. The set $\mathsf{NSP}_\mathcal{A}(f)$ of *non-simple positions*[4] of f is defined as $\{i \mid f(\overline{x_n}) \not>_\mathcal{A} x_i\}$.

Definition 5. For a precedence $>_\mathcal{F}$ and a well-founded algebra $\mathcal{A}$, the *weighted path order* $\succ_{\text{WPO}(\mathcal{A})}$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ is defined as follows: $s = f(\overline{s_n}) \succ_{\text{WPO}(\mathcal{A})} t$ iff

[4]The notion of $\mathsf{NSP}_\mathcal{A}$ is for efficiency. Indeed, one can get a simpler version of Definition 5 by replacing $\mathsf{NSP}_\mathcal{A}(f)$ by the set of all argument positions of f.

1. $s >_{\mathcal{A}} t$, or
2. $s \geq_{\mathcal{A}} t$ and
   (a) $\exists i \in \mathsf{NSP}_{\mathcal{A}}(f).\ s_i \succeq_{\mathrm{WPO}(\mathcal{A})} t$, or
   (b) $t = g(\overline{t_m})$, $\forall j \in \mathsf{NSP}_{\mathcal{A}}(g).\ s \succ_{\mathrm{WPO}(\mathcal{A})} t_j$ and
      i. $f >_{\mathcal{F}} g$ or
      ii. $f = g$ and $[\overline{s_n}] \succ^{\mathsf{lex}}_{\mathrm{WPO}(\mathcal{A})} [\overline{t_m}]$.

We abbreviate $\succ_{\mathrm{WPO}(\mathcal{A})}$ by $\succ_{\mathrm{WPO}}$ when no confusion arises.

Case (1) and the condition in (2) are the same as GKBO. Case (2a) and the condition in (2b) correspond to (a) and (b) of LPO. Here we restrict $i \in \mathsf{NSP}_{\mathcal{A}}(f)$ e.g. in (2a), since $s_i \succ_{\mathrm{WPO}} t$ and $i \notin \mathsf{NSP}_{\mathcal{A}}(f)$ implies $s >_{\mathcal{A}} t$, which is considered in (1). Cases (2b–i) and (2b–ii) are common among WPO, GKBO and LPO.

In the abstract setting, we can verify that WPO is a generalization of GKBO:

THEOREM 6. *If $\mathcal{A}$ is strictly simple, then $\succ_{\mathrm{GKBO}} = \succ_{\mathrm{WPO}}$.*

PROOF. By the assumption, $\mathsf{NSP}_{\mathcal{A}}(f) = \emptyset$ for every $f \in \mathcal{F}$. Hence, case (2a) and the condition in (2b) disappear, and the definition of WPO becomes equivalent to that of GKBO. □

Now we verify that $\succ_{\mathrm{WPO}}$ is a simplification order. The proof consists of the following five lemmata: The first two show that $\succ_{\mathrm{WPO}}$ is a *rewrite relation.*

LEMMA 1. *If $\mathcal{A}$ is weakly monotone and weakly simple, then $\succ_{\mathrm{WPO}}$ is monotonic.*

PROOF. Suppose $s_i \succ_{\mathrm{WPO}} t_i$ and let us show

$$s = g(t_1, \ldots, s_i, \ldots, t_m) \succ_{\mathrm{WPO}} g(t_1, \ldots, t_i, \ldots, t_m) = t$$

Since $s_i \geq_{\mathcal{A}} t_i$, the weak monotonicity of $\mathcal{A}$ implies $s \geq_{\mathcal{A}} t$. Moreover for all $j \in \mathsf{NSP}_{\mathcal{A}}(g)$, we have $s \geq_{\mathcal{A}} t_j$ by weak simplicity of $\mathcal{A}$ and $s \succ_{\mathrm{WPO}} t_j$ by case (2a) of Definition 5. Hence, case (2b–ii) applies and we get $s \succ_{\mathrm{WPO}} t$. □

LEMMA 2. *$\succ_{\mathrm{WPO}}$ is stable.*

PROOF. Suppose $s = f(\overline{s_n}) \succ_{\mathrm{WPO}} t$ and σ is an arbitrary substitution. Let us show $s\sigma \succ_{\mathrm{WPO}} t\sigma$ by induction on $|s| + |t|$.

It is obvious if $s >_{\mathcal{A}} t$. Otherwise, we have $s \geq_{\mathcal{A}} t$ and obviously $s\sigma \geq_{\mathcal{A}} t\sigma$. The remaining cases are as follows:

- Suppose $s_i \succeq_{\mathrm{WPO}} t$ for some $i \in \mathsf{NSP}_{\mathcal{A}}(f)$. By the induction hypothesis, we get $s_i\sigma \succeq_{\mathrm{WPO}} t\sigma$. Hence, case (2a) applies for $s\sigma \succ_{\mathrm{WPO}} t\sigma$.
- Suppose $t = g(\overline{t_m})$ and $s \succ_{\mathrm{WPO}} t_j$ for all $j \in \mathsf{NSP}_{\mathcal{A}}(g)$. By the induction hypothesis, we get $s\sigma \succ_{\mathrm{WPO}} t_j\sigma$. It is obvious if $f >_{\mathcal{F}} g$. If $f = g$ and $[\overline{s_n}] \succ^{\mathsf{lex}}_{\mathrm{WPO}} [\overline{t_m}]$, then by the induction hypothesis we get

  $$[s_1\sigma, \ldots, s_n\sigma] \succ^{\mathsf{lex}}_{\mathrm{WPO}} [t_1\sigma, \ldots, t_m\sigma]$$

  Hence, case (2b–ii) applies for $s\sigma \succ_{\mathrm{WPO}} t\sigma$. □

LEMMA 3. *$\succ_{\mathrm{WPO}}$ is transitive.*

PROOF. Suppose $s \succ_{\mathrm{WPO}} t \succ_{\mathrm{WPO}} u$ and let us show $s \succ_{\mathrm{WPO}} u$ by induction on $|s| + |t| + |u|$. It is obvious if $s >_{\mathcal{A}} t$ or $t >_{\mathcal{A}} u$. Otherwise, we have $s = f(\overline{s_n}) \geq_{\mathcal{A}} t = g(\overline{t_m}) \geq_{\mathcal{A}} u$, and by transitivity of $\geq_{\mathcal{A}}$, $s \geq_{\mathcal{A}} u$. The proof proceeds to case splitting of $s \succ_{\mathrm{WPO}} t$.

- Suppose $s_i \succeq_{\mathrm{WPO}} t$ for some $i \in \mathsf{NSP}_{\mathcal{A}}(f)$. By the induction hypothesis we get $s_i \succeq_{\mathrm{WPO}} u$. Hence, (2a) of Definition 5 applies for $s \succ_{\mathrm{WPO}} u$.
- Suppose $s \succ_{\mathrm{WPO}} t_j$ for all $j \in \mathsf{NSP}_{\mathcal{A}}(g)$. The proof proceeds to case splitting of the derivation of $t \succ_{\mathrm{WPO}} u$.
  - Suppose $t_j \succeq_{\mathrm{WPO}} u$ for some $j \in \mathsf{NSP}_{\mathcal{A}}(g)$. Since we already have $s \succ_{\mathrm{WPO}} t_j$, we get $s \succ_{\mathrm{WPO}} u$ by the induction hypothesis.
  - Suppose $u = h(\overline{u_l})$ and $t \succ_{\mathrm{WPO}} u_k$ for all $k \in \mathsf{NSP}_{\mathcal{A}}(h)$. By the induction hypothesis, we have $s \succ_{\mathrm{WPO}} u_k$ for all $k \in \mathsf{NSP}_{\mathcal{A}}(h)$. If either $f >_{\mathcal{F}} g$ or $g >_{\mathcal{F}} h$, then we have $s \succ_{\mathrm{WPO}} u$ by case (2b–i). Otherwise we have $f = g = h$ and $[\overline{s_n}] \succ^{\mathsf{lex}}_{\mathrm{WPO}} [\overline{t_m}] \succ^{\mathsf{lex}}_{\mathrm{WPO}} [\overline{u_l}]$. By the induction hypothesis and the transitivity preservation of lex, we get $[\overline{s_n}] \succ^{\mathsf{lex}}_{\mathrm{WPO}} [\overline{u_l}]$. Hence, case (2b–ii) applies for $s \succ_{\mathrm{WPO}} u$. □

To simplify the proof for the irreflexivity, we show the subterm property in advance.

LEMMA 4. *If $\mathcal{A}$ is weakly simple, then $\succ_{\mathrm{WPO}}$ has the subterm property.*

PROOF. Let us prove that $s = f(\overline{s_n}) \succ_{\mathrm{WPO}} s_i$. It is trivial if $s >_{\mathcal{A}} s_i$. Otherwise $i \in \mathsf{NSP}_{\mathcal{A}}(f)$ and $s \geq_{\mathcal{A}} s_i$ by the weak simplicity of $\mathcal{A}$. Since $s_i \succeq_{\mathrm{WPO}} s_i$, case (2a) applies for $s \succ_{\mathrm{WPO}} s_i$. □

LEMMA 5. *If $\mathcal{A}$ is weakly monotone and weakly simple, then $\succ_{\mathrm{WPO}}$ is irreflexive.*

PROOF. We show $s \not\succ_{\mathrm{WPO}} s$ by structural induction on s. If $s \in \mathcal{V}$, then no case in Definition 5 applies. Suppose $s = f(\overline{s_n})$. Trivially, $s \not>_{\mathcal{A}} s$ and $s \geq_{\mathcal{A}} s$ holds. Suppose case (2a) applies, i.e. there exists i s.t. $s_i \succeq_{\mathrm{WPO}} s$. Since Lemma 4 yields $s \succ_{\mathrm{WPO}} s_i$, we obtain $s_i \succ_{\mathrm{WPO}} s_i$ by the transitivity. This contradicts the induction hypothesis. Furthermore, $[\overline{s_n}] \not\succ^{\mathsf{lex}}_{\mathrm{WPO}} [\overline{s_n}]$ because of the induction hypothesis. Hence, case (2b–ii) does not apply, either. □

Now all desired properties are shown, and we arrive our main goal of this section:

THEOREM 7. *If $\mathcal{A}$ is weakly monotone and weakly simple, then $\succ_{\mathrm{WPO}}$ is a simplification order.* □

4. INSTANCES OF WPO

In this section, we introduce several instances of WPO. The first instance WPO($\mathcal{S}um$) is induced by an algebra $\mathcal{S}um$, which interprets function symbols as the summation operator $\sum$. We obtain KBO as a restricted case of WPO($\mathcal{S}um$). Then this order is generalized to WPO($\mathcal{P}ol$) which is induced by a monotone polynomial interpretation. POLO is subsumed by WPO($\mathcal{P}ol$), and TKBO is obtained as a restricted case of WPO($\mathcal{P}ol$). The third instance WPO($\mathcal{M}ax$) is induced by an algebra $\mathcal{M}ax$, which interprets function symbols as the maximum operator. We obtain LPO as a restricted case of WPO($\mathcal{M}ax$). The last instance WPO($\mathcal{MP}ol$) uses both polynomial and max interpretations. Hence, KBO, TKBO, LPO and POLO (with/without max) are all subsumed by WPO($\mathcal{MP}ol$).

4.1 WPO($\mathcal{S}um$)

We design $\succ_{\mathrm{WPO}(\mathcal{S}um)}$ from a weight function (w, w_0), such that $\succ_{\mathrm{WPO}(\mathcal{S}um)} = \succ_{\mathrm{KBO}}$ when $w_0 > 0$ and the admissibility is satisfied. First, we define an $\mathcal{F}$-algebra $\mathcal{S}um$ which plays the role of weights of KBO.

Definition 6. The $\mathcal{F}$-algebra $\mathcal{S}um$ induced by a weight function (w, w_0) consists of the carrier set $\{a \in \mathbb{N} \mid a \geq w_0\}$ and the interpretation which is defined as follows:

$$f_{\mathcal{S}um}(\overline{a_n}) = w(f) + \sum_{i=1}^{n} a_i$$

Obviously, $\mathcal{S}um$ is strictly (and hence weakly) monotone and weakly simple. We obtain the following as a corollary of Theorem 7:

COROLLARY 2. *$\succ_{\mathrm{WPO}(\mathcal{S}um)}$ is a simplification order.* □

If $w_0 > 0$ is satisfied, we also write $\mathcal{S}um^+$ for $\mathcal{S}um$.

Now let us prove that $\succ_{\mathrm{KBO}}$ is obtained as a special case of $\succ_{\mathrm{WPO}(\mathcal{S}um^+)}$. The following lemma verifies that $\mathcal{S}um$ indeed works as the weight of KBO.

LEMMA 6. *For $\sqsupset$ denoting either $\geq$ or $>$, $s \sqsupset_{\mathcal{S}um} t$ iff $|s|_x \geq |t|_x$ for all $x \in \mathcal{V}$ and $w(s) \sqsupset w(t)$.*

PROOF. The if-part is easy. For the only-if-part, suppose $s \sqsupset_{\mathcal{S}um} t$. Define the assignment α_0 which maps all variables to w_0. We have $\widehat{\alpha}_0(s) \sqsupset \widehat{\alpha}_0(t)$, that is $w(s) \sqsupset w(t)$. Furthermore, define the assignment α_x which maps x to $w(s) + w_0$ and others to w_0. We have $\widehat{\alpha}_x(s) \sqsupset \widehat{\alpha}_x(t)$, which implies $w(s) + |s|_x \cdot w(s) \sqsupset w(t) + |t|_x \cdot w(s)$. Hence, we get $|s|_x \geq |t|_x$. □

THEOREM 8. *If $w_0 > 0$ and w is admissible for $>_{\mathcal{F}}$, then $\succ_{\mathrm{WPO}(\mathcal{S}um)} = \succ_{\mathrm{KBO}}$.*

PROOF. For arbitrary terms $s = f(\overline{s_n})$ and t, we show $s \succ_{\mathrm{WPO}(\mathcal{S}um)} t$ iff $s \succ_{\mathrm{KBO}} t$ by induction on $|s| + |t|$.

- Suppose $s \succ_{\mathrm{KBO}} t$. If $w(s) > w(t)$, then we have $s >_{\mathcal{S}um} t$ by Lemma 6 and $s \succ_{\mathrm{WPO}(\mathcal{S}um)} t$ by (1) of Definition 5. Let us consider that $w(s) = w(t)$.
  - Suppose $s = f^k(t)$ and $t \in \mathcal{V}$ for some $k > 0$. Since $w(s) = w(t)$, $w(f) = 0$ and $\mathsf{NSP}_{\mathcal{S}um}(f) = \{1\}$. If $k = 1$, then we are done by case (2a). Otherwise $f^{k-1}(t) \succ_{\mathrm{KBO}} t$ by case (2a) of Definition 2. By the induction hypothesis we get $f^{k-1}(t) \succ_{\mathrm{WPO}(\mathcal{S}um)} t$, and hence case (2a) of Definition 5 applies.
  - Suppose $t = g(\overline{t_m})$ and case (2b–i) or (2b–ii) applies. For all $j \in \{1, \ldots, m\}$, we have $t \succ_{\mathrm{KBO}} t_j$ by the subterm property of $\succ_{\mathrm{KBO}}$, and we get $s \succ_{\mathrm{KBO}} t_j$ by the transitivity. By the induction hypothesis, $s \succ_{\mathrm{WPO}(\mathcal{S}um)} t_j$. Hence, the side condition in (2b) of Definition 5 is satisfied, and subcase (2b–i) or (2b–ii) applies.
- Suppose $s \succ_{\mathrm{WPO}(\mathcal{S}um)} t$. If $s >_{\mathcal{S}um} t$, then $w(s) > w(t)$ by Lemma 6 and $s \succ_{\mathrm{KBO}} t$ by (1) of Definition 2. Otherwise we get $w(s) = w(t)$ by Lemma 6.
  - Suppose $s_i \succsim_{\mathrm{WPO}(\mathcal{S}um)} t$ for some $i \in \mathsf{NSP}_{\mathcal{S}um}(f)$. By the induction hypothesis, we have $s_i \succsim_{\mathrm{KBO}} t$. The subterm property of $\succ_{\mathrm{KBO}}$ ensures $s \succ_{\mathrm{KBO}} s_i$. Hence by the transitivity, we get $s \succ_{\mathrm{KBO}} t$.
  - Suppose $t = g(\overline{t_m})$. If $f >_{\mathcal{F}} g$, then case (2b–i) of Definition 2 applies. If $f = g$ and $[\overline{s_n}] \succ^{\mathsf{lex}}_{\mathrm{WPO}(\mathcal{S}um)} [\overline{t_m}]$, then by the induction hypothesis we get $[\overline{s_n}] \succ^{\mathsf{lex}}_{\mathrm{KBO}} [\overline{t_m}]$, and hence case (2b–ii) applies.

□

Note that we do not need both admissibility and $w_0 > 0$ in Corollary 2. Let us see that the removals of these constraints are indeed advantageous. The following example illustrates that WPO($\mathcal{S}um^+$) properly enhances KBO because the admissibility is relaxed.

Example 1. Consider the following TRS $\mathcal{R}_1$:

$$\mathcal{R}_1 := \left\{ \begin{array}{l} \mathsf{f}(\mathsf{g}(x)) \to \mathsf{g}(\mathsf{f}(\mathsf{f}(x))) \\ \mathsf{f}(\mathsf{h}(x)) \to \mathsf{h}(\mathsf{h}(\mathsf{f}(x))) \end{array} \right.$$

The first rule cannot be oriented by LPO in any precedence. The second rule cannot be oriented by KBO, since it requires that $\mathsf{f} >_{\mathcal{F}} \mathsf{h}$ and $w(\mathsf{h}) = 0$ which is not admissible. On the other hand, WPO($\mathcal{S}um^+$) with precedence $\mathsf{f} >_{\mathcal{F}} \mathsf{g}$, $\mathsf{f} >_{\mathcal{F}} \mathsf{h}$ and $w(\mathsf{g}) > w(\mathsf{f}) = w(\mathsf{h}) = 0$ orients all the rules. Hence, $\mathcal{R}_1$ is orientable by WPO($\mathcal{S}um^+$), but not by KBO or LPO.

Moreover, allowing $w_0 = 0$ is also a proper enhancement.

Example 2. Consider the following TRS $\mathcal{R}_2$:

$$\mathcal{R}_2 := \left\{ \begin{array}{l} \mathsf{f}(\mathsf{a}, \mathsf{b}) \to \mathsf{f}(\mathsf{b}, \mathsf{f}(\mathsf{b}, \mathsf{a})) \\ \mathsf{f}(\mathsf{a}, \mathsf{f}(\mathsf{b}, x)) \to \mathsf{f}(x, \mathsf{f}(\mathsf{b}, \mathsf{b})) \end{array} \right.$$

The first rule cannot be oriented by KBO or WPO($\mathcal{S}um^+$), since $w(\mathsf{b}) = 0$ is required. The second rule is not orientable by LPO. On the other hand, WPO($\mathcal{S}um$) with $w(\mathsf{a}) > w(\mathsf{b}) = w(\mathsf{f}) = 0$ and $\mathsf{a} >_{\mathcal{F}} \mathsf{b}$ orients the both rules. Hence, $\mathcal{R}_2$ is orientable by WPO($\mathcal{S}um$) with $w_0 = 0$, but not by LPO, KBO, or WPO($\mathcal{S}um^+$).

4.2 WPO($\mathcal{P}ol$)

In this section, we consider generalizing WPO($\mathcal{S}um$) using monotone polynomial interpretations.

Definition 7. The $\mathcal{F}$-algebra $\mathcal{P}ol$ consists of the carrier set $\{a \in \mathbb{N} \mid a \geq w_0\}$ and monotone polynomial interpretations $f_{\mathcal{P}ol}(\overline{a_n})$ over $\mathbb{N}$.

According to [32, Proposition 4], every monotone interpretation on totally ordered set is weakly simple. Hence $\mathcal{P}ol$ is weakly simple and we obtain the following:

COROLLARY 3. *$\succ_{\mathrm{WPO}(\mathcal{P}ol)}$ is a simplification order.* □

Trivially, POLO is subsumed by WPO($\mathcal{P}ol$). More precisely, the following relation holds:

THEOREM 9. *$>_{\mathcal{P}ol} \subseteq \succ_{\mathrm{WPO}(\mathcal{P}ol)}$.* □

In the remainder of this paper, we consider $\mathcal{P}ol$ consists of linear polynomial interpretations induced by a weight function (w, w_0) and a subterm coefficient function sc, which is defined as follows:

$$f_{\mathcal{P}ol}(\overline{a_n}) := w(f) + \sum_{i=1}^{n} sc(f, i) \cdot a_i$$

Analogous to Theorem 8, we obtain the following:

THEOREM 10. *If $w_0 > 0$ and w is admissible for $>_{\mathcal{F}}$, then* $\succ_{\mathrm{WPO}(\mathcal{P}ol)} = \succ_{\mathrm{TKBO}}$. □

Moreover, we can verify that WPO($\mathcal{P}ol$) strictly enhances both POLO and TKBO.

Example 3. POLO cannot orient the first rule of $\mathcal{R}_1$:

$$l_1 = \mathsf{f}(\mathsf{g}(x)) \to \mathsf{g}(\mathsf{f}(\mathsf{f}(x))) = r_1$$

since it is not ω-terminating [31]. Suppose $\mathcal{R}_1$ is oriented by TKBO. For the first rule, we need

$$\mathsf{vc}(x, l_1) = sc(\mathsf{f}, 1) \cdot sc(\mathsf{g}, 1) \geq sc(\mathsf{g}, 1) \cdot sc(\mathsf{f}, 1)^2 = \mathsf{vc}(x, r_1)$$

Hence $sc(\mathsf{f}, 1) = 1$. Moreover,

$$\begin{aligned} w(l_1) &= w(\mathsf{f}) + w(\mathsf{g}) + sc(\mathsf{g}, 1) \cdot w_0 \\ &\geq w(\mathsf{g}) + sc(\mathsf{g}, 1) \cdot (2 \cdot w(\mathsf{f}) + w_0) = w(r_1) \end{aligned}$$

Hence $w(\mathsf{f}) = 0$. Analogously, for the second rule of $\mathcal{R}_1$:

$$l_2 = \mathsf{f}(\mathsf{h}(x)) \to \mathsf{h}(\mathsf{h}(\mathsf{f}(x))) = r_2$$

we need $sc(\mathsf{h}, 1) = 1$ and $w(\mathsf{h}) = 0$. Hence $w(l_2) = w(r_2)$. By the admissibility, $\mathsf{f} >_{\mathcal{F}} \mathsf{h}$ cannot hold and this rule cannot be oriented by TKBO.

In addition, we can verify that WPO($\mathcal{P}ol$) is not subsumed by the first-order RPOLO defined in [4]. More precisely, RPOLO does not subsume KBO.

Example 4. Let us show that the first rule of $\mathcal{R}_1$:

$$l = \mathsf{f}(\mathsf{g}(x)) \to \mathsf{g}(\mathsf{f}(\mathsf{f}(x))) = r$$

cannot be oriented by RPOLO. Note that this rule is oriented by KBO with $w(\mathsf{f}) = 0$ and $\mathsf{f} >_{\mathcal{F}} \mathsf{g}$.

- Suppose $f \in \mathcal{F}_{\mathrm{POLO}}$. Since this rule cannot be oriented by POLO, g must be in $\mathcal{F}_{\mathrm{RPO}}$. Hence we need $\mathsf{f}_{\mathcal{P}ol}(v_{\mathsf{g}(x)}) >_{C(l)} v_{\mathsf{g}(\mathsf{f}(\mathsf{f}(x)))}$. This requires either
  - $\mathsf{f}_{\mathcal{P}ol}(x) = x$ and $\mathsf{g}(x) \succ_{\mathrm{RPOLO}} \mathsf{g}(\mathsf{f}(\mathsf{f}(x)))$, or
  - $\mathsf{f}_{\mathcal{P}ol}(x) > x$ and $\mathsf{g}(x) \succeq_{\mathrm{RPOLO}} \mathsf{g}(\mathsf{f}(\mathsf{f}(x)))$.

  In either case, we obtain $\mathsf{g}(x) \succ_{\mathrm{RPOLO}} \mathsf{g}(x)$, which is a contradiction.

- Suppose $\mathsf{f} \in \mathcal{F}_{\mathrm{RPO}}$. Since this rule cannot be oriented by RPO, g must be in $\mathcal{F}_{\mathrm{POLO}}$. Hence we need either
  - $\mathsf{g}(x) \succeq_{\mathrm{RPOLO}} \mathsf{g}(\mathsf{f}(\mathsf{f}(x)))$, or
  - $\mathsf{f}(\mathsf{g}(x)) \succ_{\mathrm{RPOLO}} \mathsf{f}(\mathsf{f}(x))$.

  The first case contradicts with $\mathsf{f}(x) \succ_{\mathrm{RPOLO}} x$. The second case contradicts with the fact that $\mathsf{f}(x) \succ_{\mathrm{RPOLO}} \mathsf{g}(x)$.

We have not yet ensured if $\succ_{\mathrm{RPOLO}} \subseteq \succ_{\mathrm{WPO}(\mathcal{P}ol)}$, nor found a counterexample; we leave the task for future work. Nonetheless, we expect WPO($\mathcal{P}ol$) is beneficial since it can assign precedences for all symbols, while RPOLO forces POLO-symbols to have only minimum precedence.

4.3 WPO($\mathcal{M}ax$)

Note that $\mathcal{P}ol$ is strictly monotone. WPO also admits *weakly* monotone interpretations; a typical example is max. Let us consider an instance of WPO using max for interpretation.

Definition 8. A *subterm penalty function sp* is a mapping s.t. $sp(f, i) \in \mathbb{N}$ is defined for each $f \in \mathcal{F}_n$ and $i \in \{1, \dots, n\}$. A weight function (w, w_0) and sp induce the $\mathcal{F}$-algebra $\mathcal{M}ax$, which consists of the carrier set $\{a \in \mathbb{N} \mid a \geq w_0\}$ and interpretations given by:

$$f_{\mathcal{M}ax}(\overline{a_n}) := \max\left(w(f), \max_{i=1}^{n}\left(sp(f, i) + a_i\right)\right)$$

LEMMA 7. *$\mathcal{M}ax$ is weakly monotone and weakly simple.*

PROOF. Weak simplicity is obvious from the fact that $\max\{\dots, a, \dots\} \geq a$. For weak monotonicity, suppose $a > b$ and let us show

$$a' = f_{\mathcal{M}ax}(\overline{c_k}, a, \overline{d_l}) \geq f_{\mathcal{M}ax}(\overline{c_k}, b, \overline{d_l}) = b'$$

Let $c = f_{\mathcal{M}ax}(\overline{c_k}, 0, \overline{d_l})$. If $c \geq sp(f, k+1) + a$, then $a' = b' = c$. Otherwise, we have $a' = sp(f, k+1) + a$ and either $a' > sp(f, k+1) + b = b'$ or $a' > c = b'$. □

Note that $\mathcal{M}ax$ can be considered as the dimension-1 variant of *arctic interpretations* [16]. The weak monotonicity of $\mathcal{M}ax$ is also shown there.

COROLLARY 4. *$\succ_{\mathrm{WPO}(\mathcal{M}ax)}$ is a simplification order.* □

Now let us show that LPO can be obtained as a restricted case of WPO($\mathcal{M}ax$).

THEOREM 11. *If $w_0 = 0$, $w(f) = 0$ and $sp(f, i) = 0$ for all $f \in \mathcal{F}_n$ and $i \in \{1, \dots, n\}$, then $\succ_{\mathrm{LPO}} = \succ_{\mathrm{WPO}(\mathcal{M}ax)}$.*

PROOF. Obviously, $s >_{\mathcal{M}ax} t$ never holds. Hence, case (1) of Definition 5 can be ignored. Moreover, $s \geq_{\mathcal{M}ax} t$ is equivalent to $\mathcal{V}ar(s) \supseteq \mathcal{V}ar(t)$. One can easily verify the latter holds whenever $s \succ_{\mathrm{LPO}} t$, using the fact that $s \not\succeq_{\mathrm{LPO}} x$ for $x \notin \mathcal{V}ar(s)$. Hence, the condition of (2) can be ignored and Definition 1 and Definition 5 become equivalent. □

The following example illustrates that WPO($\mathcal{M}ax$) properly enhances LPO.

Example 5. Consider the following TRS $\mathcal{R}_3$:

$$\mathcal{R}_3 := \left\{ \begin{array}{l} \mathsf{f}(x, y) \to \mathsf{g}(x) \\ \mathsf{f}(x, \mathsf{g}(\mathsf{g}(y))) \to \mathsf{f}(\mathsf{g}(y), \mathsf{g}(y)) \end{array} \right.$$

To orient the first rule by LPO, we need $\mathsf{f} >_{\mathcal{F}} \mathsf{g}$. LPO cannot orient the second rule by this precedence, while $sp(\mathsf{g}, 1) > 0$ suffices for WPO($\mathcal{M}ax$). Since the second rule is duplicating, KBO or WPO($\mathcal{S}um$) cannot apply for $\mathcal{R}_3$.

However, WPO($\mathcal{M}ax$) does not cover WPO($\mathcal{S}um$), or not even KBO. In the next section, we consider unifying WPO($\mathcal{S}um$) and WPO($\mathcal{M}ax$) to cover both KBO and LPO.

4.4 WPO($\mathcal{MP}ol$) and WPO($\mathcal{MS}um$)

Let us consider unifying WPO($\mathcal{M}ax$) and WPO($\mathcal{P}ol$). The goal is achieved by an approach that resembles the *status* which unifies LPO into RPO. We introduce the *weight status* to choose a polynomial or max for each function symbol.

Definition 9. A *weight status function* is a mapping ws which maps each function symbol f to either symbol pol or max. The $\mathcal{F}$-algebra $\mathcal{MP}ol$ consists of the carrier set $\{a \in \mathbb{N} \mid a \geq w_0\}$ and the interpretation which is defined as follows:

$$f_{\mathcal{MP}ol}(\overline{a_n}) := \begin{cases} w(f) + \sum_{i=1}^{n} sc(f,i) \cdot a_i & \text{if } ws(f) = \mathsf{pol} \\ \max\left(w(f), \max_{i=1}^{n}\left(sp(f,i) + a_i\right)\right) & \text{if } ws(f) = \mathsf{max} \end{cases}$$

We denote $\mathcal{MP}ol$ by $\mathcal{MS}um$ if coefficients are fixed to 1. Trivially, WPO($\mathcal{MS}um$) encompasses both WPO($\mathcal{S}um$) and WPO($\mathcal{M}ax$). Hence, we obtain the following more influential result:

THEOREM 12. WPO($\mathcal{MS}um$) *encompasses both LPO and KBO.* □

The following example illustrates that WPO($\mathcal{MS}um$) is strictly stronger than the union of WPO($\mathcal{S}um$) and WPO($\mathcal{M}ax$).

Example 6. Consider the following TRS $\mathcal{R}_4$:

$$\mathcal{R}_4 := \begin{cases} \mathsf{f}(\mathsf{f}(x,y),z) \to \mathsf{f}(x,\mathsf{f}(y,z)) \\ \mathsf{g}(\mathsf{f}(\mathsf{a},x),\mathsf{b}) \to \mathsf{g}(\mathsf{f}(x,\mathsf{b}),x) \end{cases}$$

If $ws(\mathsf{f}) = \mathsf{max}$, then the first rule requires $sp(\mathsf{f},2) = 0$. Under this restriction the second rule cannot be oriented. If $ws(\mathsf{f}) = \mathsf{pol}$, then the first rule is always oriented. On the other hand, the duplicating variable x in the second rule requires $ws(\mathsf{g}) = \mathsf{max}$. Hence, $\mathcal{R}_4$ is orientable by WPO($\mathcal{MS}um$) only if $ws(\mathsf{f}) = \mathsf{pol}$ and $ws(\mathsf{g}) = \mathsf{max}$.

Let us close this section with an example that suggests WPO($\mathcal{MS}um$) may advance the state-of-the-art of automated termination proving.

Example 7. The most powerful termination provers including AProVE 2013 and $\mathsf{T_TT_2}$ 1.11 fail to prove termination of the following TRS $\mathcal{R}_5$:

$$\mathcal{R}_5 := \begin{cases} \mathsf{f}(\mathsf{g}(\mathsf{g}(x,\mathsf{a}),\mathsf{g}(\mathsf{b},y))) \to \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{h}(x,x),\mathsf{b}),\mathsf{g}(y,\mathsf{a}))) \\ \mathsf{g}(x,y) \to x \\ \mathsf{h}(x,\mathsf{h}(y,z)) \to y \end{cases}$$

Let us show that WPO($\mathcal{MS}um$) with $ws(\mathsf{g}) = \mathsf{pol}$, $ws(\mathsf{h}) = \mathsf{max}$, $w(\mathsf{a}) > w(\mathsf{b})$ and $w(\mathsf{h}) = sp(\mathsf{h},1) = sp(\mathsf{h},2) = 0$ orients all the rules. For the first rule, applying case (2b–ii) twice it yields orienting $\mathsf{g}(x,\mathsf{a}) \succ_{\mathrm{WPO}(\mathcal{MS}um)} \mathsf{g}(\mathsf{h}(x,x),\mathsf{b})$ where case (1) applies. The other rules are trivially oriented.

5. SMT ENCODING OF WPO

In the preceding sections, we have concentrated on theoretical aspects. In this section, we consider how to implement the instances of WPO using SMT solvers. We extend the corresponding approach for KBO [30] to WPO. In particular, WPO($\mathcal{S}um$), WPO($\mathcal{M}ax$) and WPO($\mathcal{MS}um$) are reduced to SMT problems of linear arithmetic, and as a consequence, decidability is ensured for orientability problems of these orders.

An *expression* e is built from (non-negative integer) variables, constants and the binary symbols $\cdot$ and $+$ denoting multiplication and addition, resp. A *formula* is built from atoms of the form $e_1 > e_2$ and $e_1 \geq e_2$, and the binary symbols $\wedge$, $\vee$ and $\Rightarrow$ denoting conjunction, disjunction and implication, resp. The precedence of these symbols are in the order we listed above.

5.1 The Common Structure

To optimize the presentation, we first present an encoding of the common structure of WPO independent from the choice of $\mathcal{A}$. Hence, we assume encodings for $>_{\mathcal{A}}$, $\geq_{\mathcal{A}}$ and $\mathsf{NSP}_{\mathcal{A}}(f)$ are given. We introduce an integer variable p_f for each $f \in \mathcal{F}$, which denotes the position of f in the precedence.

Definition 10. The encoding of $s \succ_{\mathrm{WPO}(\mathcal{A})} t$ is defined as follows:

$$[\![s \succ_{\mathrm{WPO}(\mathcal{A})} t]\!] := \begin{cases} \mathtt{false} & \text{if } s \in \mathcal{V} \text{ or } t \in \mathcal{V} \setminus \mathcal{V}ar(s) \\ \mathtt{true} & \text{if } s \notin \mathcal{V} \text{ and } t \in \mathcal{V}ar(s) \\ \phi_1 & \text{if } s = f(\overline{s_n}) \text{ and } t = g(\overline{t_m}) \end{cases}$$

where

$$\phi_1 := [\![s >_{\mathcal{A}} t]\!] \vee \left([\![s \geq_{\mathcal{A}} t]\!] \wedge \phi_2\right)$$

$$\phi_2 := \bigvee_{i \in [\![\mathsf{NSP}_{\mathcal{A}}(f)]\!]} [\![s_i \succeq_{\mathrm{WPO}(\mathcal{A})} t]\!] \;\vee\; \bigwedge_{j \in [\![\mathsf{NSP}_{\mathcal{A}}(g)]\!]} [\![s \succ_{\mathrm{WPO}(\mathcal{A})} t_j]\!] \;\wedge\; \phi_3$$

$$\phi_3 := \begin{cases} \mathsf{p}_f > \mathsf{p}_g & \text{if } f \neq g \\ [\![s_k \succ_{\mathrm{WPO}(\mathcal{A})} t_k]\!] & \text{if } f = g \end{cases}$$

where k denotes the least $i \leq n$ s.t. $s_i \neq t_i$.

THEOREM 13. *Let encodings for $\mathcal{A}$ are given s.t.*

1. *For $\sqsupset$ denoting $\geq$ or $>$, $[\![s \sqsupset_{\mathcal{A}} t]\!]$ iff $s \sqsupset_{\mathcal{A}} t$, and*
2. *$i \in [\![\mathsf{NSP}_{\mathcal{A}}(f)]\!]$ if $i \in \mathsf{NSP}_{\mathcal{A}}(f)$.*

Then, $[\![s \succ_{\mathrm{WPO}(\mathcal{A})} t]\!]$ iff $s \succ_{\mathrm{WPO}(\mathcal{A})} t$. □

Note that an encoding of $\mathsf{NSP}_{\mathcal{A}}(f)$ can be an *overestimation*; it suffices if $i \in [\![\mathsf{NSP}_{\mathcal{A}}(f)]\!]$ holds whenever $i \in \mathsf{NSP}_{\mathcal{A}}(f)$. Using this fact, one may reduce the size of encoding by reducing $i \in [\![\mathsf{NSP}_{\mathcal{A}}(f)]\!]$ only if it is statically known to be `false`.

In the following sections, we give encodings depending on the choice of $\mathcal{A}$ for each instance of WPO.

5.2 Encoding WPO($\mathcal{P}ol$) and WPO($\mathcal{S}um$)

In this section, we consider encoding linear polynomials $\mathcal{P}ol$. The encodings for $\mathcal{S}um$ is obtained by fixing coefficients to 1. We introduce non-negative integer variables w_0, w_f and $\mathsf{sc}_{f,i}$, each representing w_0, $w(f)$ and $sc(f,i)$ for every $f \in \mathcal{F}_n$ and $i \in \{1,\ldots,n\}$. The weight of a term s and the variable coefficient of x in s are encoded as follows:

$$\mathsf{w}(s) := \begin{cases} \mathsf{w}_0 & \text{if } s \in \mathcal{V} \\ \mathsf{w}_f + \sum_{i=1}^{n} \mathsf{sc}_{f,i} \cdot \mathsf{w}(s_i) & \text{if } s = f(\overline{s_n}) \end{cases}$$

$$\mathsf{vc}(x,s) := \begin{cases} 1 & \text{if } x = s \\ 0 & \text{if } x \neq s \in \mathcal{V} \\ \sum_{i=1}^{n} \mathsf{sc}_{f,i} \cdot \mathsf{vc}(x,s_i) & \text{if } s = f(\overline{s_n}) \end{cases}$$

In order to ensure w_0 to be the lower bound, we introduce the following constraint:

$$\mathsf{WMIN} := \bigwedge_{c \in \mathcal{F}_0} \mathsf{w}_c \geq \mathsf{w}_0$$

To ensure the subterm coefficients to be greater than 1, we introduce the following constraint:

$$\mathsf{COEF} := \bigwedge_{f \in \mathcal{F}_n} \bigwedge_{i=1}^{n} \mathsf{sc}_{f,i} \geq 1$$

For $\sqsupset$ denoting either $>$ or $\geq$, $\sqsupset_{\mathcal{P}ol}$ is encoded as follows:

$$[\![s \sqsupset_{\mathcal{P}ol} t]\!] := \mathsf{w}(s) \sqsupset \mathsf{w}(t) \wedge \bigwedge_{x \in \mathcal{V}ar(t)} \mathsf{vc}(x,s) \geq \mathsf{vc}(x,t)$$

For estimating $\mathsf{NSP}_{\mathcal{P}ol}$, let $f \in \mathcal{F}_n$ and $i \in \{1, \dots, n\}$. The formula $i \in [\![\mathsf{NSP}_{\mathcal{P}ol}(f)]\!]$ is defined as follows:

$$i \in [\![\mathsf{NSP}_{\mathcal{P}ol}(f)]\!] := \begin{cases} \texttt{true} & \text{if } w_0 = 0 \text{ or } n = 1 \\ \texttt{false} & \text{otherwise} \end{cases}$$

THEOREM 14. *A TRS $\mathcal{R}$ is orientable by* WPO($\mathcal{P}ol$) *if the following formula is satisfiable:*

$$\mathsf{WMIN} \wedge \mathsf{COEF} \wedge \bigwedge_{l \to r \in \mathcal{R}} [\![l \succ_{\mathrm{WPO}(\mathcal{P}ol)} r]\!] \qquad \square$$

By fixing the subterm coefficients to 1, we obtain an encoding of WPO($\mathcal{S}um$). Let us write $[\![l \succ_{\mathrm{WPO}(\mathcal{S}um)} r]\!]$ to denote the formula obtained from $[\![l \succ_{\mathrm{WPO}(\mathcal{P}ol)} r]\!]$ by replacing every occurrence of $\mathsf{sc}_{f,i}$ by 1.

COROLLARY 5. *A TRS $\mathcal{R}$ is orientable by* WPO($\mathcal{S}um$) *if the following formula is satisfiable:*

$$\mathsf{WMIN} \wedge \bigwedge_{l \to r \in \mathcal{R}} [\![l \succ_{\mathrm{WPO}(\mathcal{S}um)} r]\!] \qquad \square$$

5.3 Encoding WPO($\mathcal{M}ax$)

In order to encode $\mathcal{M}ax$, we introduce a non-negative integer variable $\mathsf{sp}_{f,i}$ denoting $sp(f,i)$ for each $f \in \mathcal{F}_n$ and $i \in \{1, \dots, n\}$. The encoding of $\mathsf{NSP}_{\mathcal{M}ax}$ is given as follows:

$$i \in [\![\mathsf{NSP}_{\mathcal{M}ax}(f)]\!] := \mathsf{sp}_{f,i} = 0$$

Now we consider encoding the constraint $s >_{\mathcal{M}ax} t$ into both quantified and quantifier-free formulas. Unfortunately, we are aware of no SMT solver which supports a built-in max operator. A straightforward encoding would involve

$$\mathsf{w}(s) := \begin{cases} s & \text{if } s \in \mathcal{V} \\ v & \text{if } s = f(\overline{s_n}) \end{cases}$$

where v is a fresh integer variable with the following constraint ϕ added into the context:

$$\phi := v \geq \mathsf{w}_f \wedge \bigwedge_{i=1}^{n} v \geq \mathsf{w}(s_i) \wedge \Big(v = \mathsf{w}_f \vee \bigvee_{i=1}^{n} v = \mathsf{w}(s_i) \Big)$$

Then the constraint $s \sqsupset_{\mathcal{M}ax} t$ can be encoded as follows:

$$[\![s \sqsupset_{\mathcal{M}ax} t]\!] := \forall \overline{x_n}, \overline{v_m}.\ \phi_1 \wedge \ldots \wedge \phi_m \Rightarrow \mathsf{w}(s) \sqsupset \mathsf{w}(t)$$

where $\sqsupset$ denotes either $>$ or $\geq$, $\{x_1, \dots, x_k\} = \mathcal{V}ar(s) \cup \mathcal{V}ar(t)$, and each (ϕ_j, v_j) is the pair of the constraint and the fresh variable introduced during the encoding.

Although quantified linear integer arithmetic is known to be decidable, the SMT solvers we have tested could not solve the problems generated by the above straightforward encoding efficiently, if not at all. Fuhs et al. [10] proposes a sound elimination of quantifiers by introducing new template polynomials. Here we propose another encoding that is sound and complete for linear polynomials.

Definition 11. A *generalized weight* [17] is a pair (n, N) where $n \in \mathbb{N}$ and N is a finite multiset[5] over $\mathcal{V}$. We use the following abbreviations:

$$(n, N) + (m, M) := (n + m, N \uplus M)$$
$$n \cdot (m, M) := (n \cdot m, n \cdot M)$$

where $n \cdot M$ denotes the multiset that maps x to $n \cdot M(x)$ for every $x \in \mathcal{V}$. We encode a generalized weight as a pair of an expression and a mapping N from $\mathcal{V}$ to expressions s.t. the *domain* $\mathsf{dom}(N) := \{x \mid N(x) \neq 0\}$ of N is finite. Notations for generalized weights are naturally extended for encoded ones. The relation $\supseteq$ on multisets is encoded as follows:

$$N \supseteq M := \bigwedge_{x \in \mathsf{dom}(M)} N(x) \geq M(x)$$

A generalized weight (n, N) represents the expression $n + \sum_{x \in N} x$. Now we consider removing max.

Definition 12. The *expanded weight* $\overline{\mathsf{w}}(s)$ of a term s induced by sp is a set of generalized weights, which is defined as follows:

$$\overline{\mathsf{w}}(s) := \begin{cases} \{(\mathsf{w}_0, \{s\})\} & \text{if } s \in \mathcal{V} \\ \{(\mathsf{w}_f, \emptyset)\} \cup T & \text{if } s = f(\overline{s_n}) \end{cases}$$

where T denotes $\{\mathsf{sp}_{f,i} + p \mid p \in \overline{\mathsf{w}}(s_i),\ 1 \leq i \leq n\}$.

The expanded weight $\overline{\mathsf{w}}(s) = \{p_1, \dots, p_n\}$ represents the expression $\max_{i=1}^{n} e_i$, where each generalized weight p_i represents the expression e_i.

Using expanded weights, we can encode $\sqsupset_{\mathcal{M}ax}$ for $\sqsupset$ denoting $>$ and $\geq$, in a way similar to the *max set ordering* presented in [3]:

$$[\![s \sqsupset_{\mathcal{M}ax} t]\!] := \bigwedge_{(m,M) \in \overline{\mathsf{w}}(t)} \bigvee_{(n,N) \in \overline{\mathsf{w}}(s)} (n \sqsupset m \wedge N \supseteq M)$$

THEOREM 15. *A TRS $\mathcal{R}$ is orientable by* WPO($\mathcal{M}ax$) *if the following formula is satisfiable:*

$$\mathsf{WMIN} \wedge \bigwedge_{l \to r \in \mathcal{R}} [\![l \succ_{\mathrm{WPO}(\mathcal{M}ax)} r]\!] \qquad \square$$

5.4 Encoding WPO($\mathcal{MP}ol$) and WPO($\mathcal{MS}um$)

In this section, we consider encoding linear polynomials with max into SMT formulas. First we extend Definition 12 for weight status.

Definition 13. For a weight status ws, the *expanded weight* $\overline{\mathsf{w}}^{ws}(s)$ of a term s is the set of generalized weight, which is recursively defined as follows:

$$\overline{\mathsf{w}}^{ws}(s) := \begin{cases} \{(\mathsf{w}_0, \{s\})\} & \text{if } s \in \mathcal{V} \\ S & \text{if } s = f(\overline{s_n}),\ ws(f) = \mathsf{pol} \\ T & \text{if } s = f(\overline{s_n}),\ ws(f) = \mathsf{max} \end{cases}$$

where

$$S = \Big\{ \mathsf{w}_f + \sum_{i=1}^{n} \mathsf{sc}_{f,i} \cdot p_i \mid p_1 \in \overline{\mathsf{w}}^{ws}(s_1), \dots, p_n \in \overline{\mathsf{w}}^{ws}(s_n) \Big\}$$
$$T = \{\mathsf{w}_f\} \cup \{\mathsf{sp}_{f,i} + \mathsf{sc}_{f,i} \cdot p \mid p \in \overline{\mathsf{w}}^{ws}(s_i),\ 1 \leq i \leq n\}$$

[5]In the encoding for $\mathcal{M}ax$, N need not contain more than one variable. This generality is reserved for encoding of $\mathcal{MP}ol$.

Definition 14. For $\sqsupset$ denoting $>$ and $\geq$, the encoding of $\sqsupset_{\mathcal{MP}ol}$ is given as:

$$[\![s \sqsupset_{\mathcal{MP}ol} t]\!] := \bigwedge_{(m,M)\in\overline{w}^{ws}(t)} \bigvee_{(n,N)\in\overline{w}^{ws}(s)} (n \sqsupset m \wedge N \supseteq M)$$

THEOREM 16. *A TRS $\mathcal{R}$ is orientable by* WPO($\mathcal{MP}ol$) *if the following formula is satisfiable:*

$$\mathsf{COEF} \wedge \mathsf{WMIN} \wedge \bigwedge_{l\to r\in\mathcal{R}} [\![l \succ_{\mathrm{WPO}(\mathcal{MP}ol)} r]\!] \qquad \square$$

Analogous to Corollary 5, let us denote $[\![l \succ_{\mathrm{WPO}(\mathcal{MS}um)} r]\!]$ the formula obtained from $[\![l \succ_{\mathrm{WPO}(\mathcal{MP}ol)} r]\!]$ by replacing every occurrence of $\mathsf{sc}_{f,i}$ by 1.

COROLLARY 6. *A TRS $\mathcal{R}$ is orientable by* WPO($\mathcal{MS}um$) *if the following formula is satisfiable:*

$$\mathsf{WMIN} \wedge \bigwedge_{l\to r\in\mathcal{R}} [\![l \succ_{\mathrm{WPO}(\mathcal{MS}um)} r]\!] \qquad \square$$

6. DEPENDENCY PAIRS

The *dependency pair (DP) framework* [1, 12, 11] significantly enhances classical method of simplification orders by analyzing dependencies between rewrite rules. We briefly recall the essential notions for DP framework: Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$. The *root symbol* of a term $s = f(\overline{s_n})$ is f and denoted by $\mathsf{root}(s)$. The set of *defined symbols* w.r.t. $\mathcal{R}$ is defined as $\mathcal{D} := \{\mathsf{root}(l) \mid l \to r \in \mathcal{R}\}$. For each $f \in \mathcal{D}$, the signature $\mathcal{F}$ is extended by a fresh *marked symbol* $f^\sharp$ whose arity is the same as f. For $s = f(\overline{s_n})$ with $f \in \mathcal{D}$, the term $f^\sharp(\overline{s_n})$ is denoted by $s^\sharp$. The set of *dependency pairs* for $\mathcal{R}$ is defined as $\mathsf{DP}(\mathcal{R}) := \{l^\sharp \to t^\sharp \mid l \to r \in \mathcal{R}, t \text{ is a subterm of } r, \mathsf{root}(t) \in \mathcal{D}\}$. A *DP problem* is a pair $\langle\mathcal{P},\mathcal{R}\rangle$ of a TRS $\mathcal{R}$ and a set $\mathcal{P}$ of dependency pairs for $\mathcal{R}$. A DP problem $\langle\mathcal{P},\mathcal{R}\rangle$ is *finite* iff $\to_{\mathcal{P}} \cdot \to^*_{\mathcal{R}}$ is well-founded, where $\mathcal{P}$ is viewed as a TRS. The main result of the DP framework is the following:

THEOREM 17. *A TRS $\mathcal{R}$ is terminating if the DP problem $\langle\mathsf{DP}(\mathcal{R}),\mathcal{R}\rangle$ is finite.* $\square$

Finiteness of DP problems are proved by applying *DP processors*: A sound *DP processor* inputs a DP problem and outputs a set of (hopefully simpler) DP problems s.t. the input problem is finite if all the output problems are finite.

Following is a typical technique to design a DP processor from a simplification order: An *argument filter* [1, 20] π maps each $f \in \mathcal{F}_n$ to either a position $i \in \{1,\dots,n\}$ or a list $[\overline{i_m}]$ of positions s.t. $1 \leq i_1 < \cdots < i_m \leq n$. The signature $\mathcal{F}^\pi$ consists of every $f \in \mathcal{F}$ s.t. $\pi(f) = [\overline{i_m}]$, whose arity is m in $\mathcal{F}^\pi$. An argument filter π induces a mapping $\pi : \mathcal{T}(\mathcal{F},\mathcal{V}) \to \mathcal{T}(\mathcal{F}^\pi,\mathcal{V})$ as follows:

$$\pi(s) := \begin{cases} s & \text{if } s \in \mathcal{V} \\ \pi(s_i) & \text{if } s = f(\overline{s_n}),\ \pi(f) = i \\ f(\pi(s_{i_1}),\dots,\pi(s_{i_m})) & \text{if } s = f(\overline{s_n}),\ \pi(f) = [\overline{i_m}] \end{cases}$$

For an argument filter π and a reduction order $\succ$ on $\mathcal{T}(\mathcal{F}^\pi,\mathcal{V})$, the relations $\succsim^\pi$ and $\succ^\pi$ of relations on $\mathcal{T}(\mathcal{F},\mathcal{V})$ are defined as follows: $s \succsim^\pi t$ iff $\pi(s) \succeq \pi(t)$, and $s \succ^\pi t$ iff $\pi(s) \succ \pi(t)$.

THEOREM 18. *Let $\succ$ be a reduction order and π an argument filter s.t. $\mathcal{P} \cup \mathcal{R} \subseteq \succsim^\pi$ and $\mathcal{P}' \subseteq \succ^\pi$. Then the DP processor that maps $\langle\mathcal{P},\mathcal{R}\rangle$ to $\{\langle\mathcal{P}\setminus\mathcal{P}',\mathcal{R}\rangle\}$ is sound.* $\square$

The effect of an argument filtering is especially apparent for KBO; it relaxes the variable condition. Consider a dependency pair $p := \mathsf{f}^\sharp(\mathsf{s}(x), y) \to \mathsf{f}^\sharp(x, x)$. Without argument filtering, p cannot be oriented by KBO because x is duplicating. On the other hand, after applying an argument filter s.t. $\pi(\mathsf{f}^\sharp) = [1]$, any instance of KBO obviously orients $\mathsf{f}^\sharp(\mathsf{s}(x)) \succ_{\mathrm{KBO}} \mathsf{f}^\sharp(x)$.

We refer [11] for a summary of other DP processors which simplify or decompose DP problems.

6.1 Encoding WPO with Argument Filters

We follow [6] and [29] for encoding of an argument filter π. For every $f \in \mathcal{F}_n$, we introduce the following boolean variables: af_f which is assigned true iff $\pi(f)$ is a list, and $\mathsf{af}_{f,i}$ which is assigned true iff $\pi(f)$ is either i or a list containing i. The following constraint is introduced:

$$\mathsf{AF} := \bigwedge_{f\in\mathcal{F}_n} \left(\mathsf{af}_f \vee \sum_{i=1}^{n} \mathsf{af}_{f,i} = 1\right)$$

Definition 15. The *equality modulo* π is defined by: $s \sim^\pi t$ iff $\pi(s) = \pi(t)$, and encoded as follows:

$$[\![s \sim^\pi t]\!] := \begin{cases} \mathtt{true} & \text{if } s = t \\ \phi_1 & \text{if } s = f(\overline{s_n}),\ t = f(\overline{t_n}) \\ \phi_2 \vee \phi_3 & \text{otherwise} \end{cases}$$

where

$$\phi_1 := \bigwedge_{i=1}^{n} \left(\mathsf{af}_{f,i} \Rightarrow [\![s_i \sim^\pi t_i]\!]\right)$$

$$\phi_2 := \begin{cases} \mathtt{false} & \text{if } s \in \mathcal{V} \\ \neg\mathsf{af}_f \wedge \bigwedge_{i=1}^{n} \left(\mathsf{af}_{f,i} \Rightarrow [\![s_i \sim^\pi t]\!]\right) & \text{if } s = f(\overline{s_n}) \end{cases}$$

$$\phi_3 := \begin{cases} \mathtt{false} & \text{if } t \in \mathcal{V} \\ \neg\mathsf{af}_g \wedge \bigwedge_{j=1}^{m} \left(\mathsf{af}_{g,j} \Rightarrow [\![s \sim^\pi t_j]\!]\right) & \text{if } t = g(\overline{t_m}) \end{cases}$$

In order to incorporate an argument filter π into a weight function, [30] refines the encoding of weights to take π into account. Here we present an approach that does not change the encodings of weights but modifies the constraints on a weight function and subterm coefficients.

Definition 16. The formulas COEF^π, WMIN^π and COLL are defined as follows:

$$\mathsf{COEF}^\pi := \bigwedge_{f\in\mathcal{F}_n} \bigwedge_{i=1}^{n} \left(\mathsf{af}_{f,i} \Leftrightarrow \mathsf{sc}_{f,i} > 0\right)$$

$$\mathsf{WMIN}^\pi := \bigwedge_{f\in\mathcal{F}_n} \left(\neg\mathsf{af}_f \ \vee\ \mathsf{w}_f \geq \mathsf{w}_0 \ \vee\ \bigvee_{i=1}^{n} \mathsf{af}_{f,i}\right)$$

$$\mathsf{COLL} := \bigwedge_{f\in\mathcal{F}_n} \left(\mathsf{af}_f \vee \left(\mathsf{w}_f = 0 \ \wedge\ \bigwedge_{i=1}^{n} \mathsf{sc}_{f,i} \leq 1\right)\right)$$

COEF^π ensures that the weight of an argument is ignored iff the argument is filtered out by π. WMIN^π ensures w_0 to be the minimal weight. Together with AF and COEF^π, COLL ensures that $w(f(\overline{s_n})) = w(s_i)$ if $\pi(f) = i$ for every $f \in \mathcal{F}_n$.

Finally, we encode WPO modulo π:

Definition 17. The encoding of $s \succ^{\pi}_{\mathrm{WPO}(\mathcal{A})} t$ is defined as follows:

$$[\![s \succ^{\pi}_{\mathrm{WPO}(\mathcal{A})} t]\!] := \begin{cases} \texttt{false} & \text{if } s \in \mathcal{V} \\ [\![s >_{\mathcal{A}} t]\!] \vee \big([\![s \geq_{\mathcal{A}} t]\!] \wedge (\psi_1 \vee \psi_2) \big) & \text{if } s = f(\overline{s_n}) \end{cases}$$

where

$$\psi_1 := \bigvee_{i=1}^{n} \Big(\mathsf{af}_{f,i} \wedge \Big([\![s_i \succ^{\pi}_{\mathrm{WPO}(\mathcal{A})} t]\!] \vee \big(\mathsf{af}_f \wedge [\![s_i \sim^{\pi} t]\!] \big) \Big) \Big)$$

$$\psi_2 := \begin{cases} \texttt{false} & \text{if } t \in \mathcal{V} \\ \bigwedge_{j=1}^{m} \Big(\mathsf{af}_{g,j} \Rightarrow [\![s \succ^{\pi}_{\mathrm{WPO}(\mathcal{A})} t_j]\!] \Big) \wedge \psi_3 & \text{if } t = g(\overline{t_m}) \end{cases}$$

$$\psi_3 := \begin{cases} \mathsf{af}_g \Rightarrow \mathsf{af}_f \wedge \mathsf{p}_f > \mathsf{p}_g & \text{if } f \neq g \\ [\![[\overline{s_n}] \succ^{\mathsf{lex},\pi}_{\mathrm{WPO}(\mathcal{A})} [\overline{t_m}]]\!] & \text{if } f = g \end{cases}$$

where $\succ^{\mathsf{lex},\pi}_{\mathrm{WPO}(\mathcal{A})}$ is the lexicographic extension of $\succ^{\pi}_{\mathrm{WPO}(\mathcal{A})}$ modulo $\sim^{\pi}$ and takes π into account.

THEOREM 19. *If the following formula is satisfiable:*

$$\mathsf{AF} \wedge \mathsf{COEF}^{\pi} \wedge \mathsf{WMIN}^{\pi} \wedge \mathsf{COLL} \wedge \bigwedge_{l \to r \in \mathcal{R} \cup \mathcal{P}} [\![l \succsim^{\pi}_{\mathrm{WPO}(\mathcal{MP}ol)} r]\!] \wedge \bigvee_{l \to r \in \mathcal{P}'} [\![l \succ^{\pi}_{\mathrm{WPO}(\mathcal{MP}ol)} r]\!]$$

then the DP processor that maps $\langle \mathcal{P}, \mathcal{R} \rangle$ to $\{ \langle \mathcal{P} \setminus \mathcal{P}', \mathcal{R} \rangle \}$ is sound. □

7. IMPLEMENTATION

We implemented the encodings presented in Sections 5 and 6.1. In addition, *statuses* and *quasi precedences* are also implemented. For the DP framework, we implemented a simple estimation of *dependency graphs*, and *strongly connected components* are sequentially processed in order of size where smaller ones are precedent. Moreover, *usable rules* w.r.t. argument filters are also implemented by following the encoding proposed in [6].

For comparison, KBO, LPO, POLO (with/without max) and TKBO are implemented in the same manner. Some optimizations are performed during the encoding: formulas like $\texttt{false} \wedge \phi$ are reduced in advance to avoid generating meaningless formulas, and temporary variables are inserted to avoid multiple occurrences of an expression or a formula. For TKBO, POLO and WPO using $\mathcal{P}ol$ and $\mathcal{MP}ol$, we choose 3 for upper bounds of weights and coefficients.

7.1 Fixing w_0

Moreover, we simplify the formula by fixing w_0. For KBO, Winkler et al. [28] shows that w_0 can be fixed to arbitrary $k > 0$ e.g. 1 without loosing the power of the order. Applying their technique, it can be shown that for WPO($\mathcal{S}um$), w_0 can be fixed to 0. On the contrary to KBO, however, w_0 cannot be fixed to $k > 0$ since transforming a weight function $(w, 0)$ into (w^k, k) may assign negative weights to some symbols.

7.2 Fixing Weight Status

For POLO and WPO using algebras $\mathcal{MS}um$ and $\mathcal{MP}ol$, it may not be practical to consider all possible weight statuses, since it leads to exponential growth in encoded formula. Hence, we introduce a heuristic for fixing ws. In case of WPO($\mathcal{MS}um$), ws should at least satisfy the following condition for all $l \to r \in \mathcal{R}$:

$$\bigwedge_{(m,M) \in \overline{\mathrm{w}}^{ws}(r)} \bigvee_{(n,N) \in \overline{\mathrm{w}}^{ws}(l)} N \supseteq M$$

since otherwise the formula $\bigwedge_{l \to r \in \mathcal{R}} [\![l \succ_{\mathrm{WPO}(\mathcal{MS}um)} r]\!]$ is trivially unsatisfiable. Hence in the implementation, we consider $\mathcal{MS}um$ and $\mathcal{MP}ol$ are induced by the weight status function which minimizes the number of f with $ws(f) = \mathsf{max}$, while satisfying the above condition.

8. EXPERIMENTS

The experiments are run on a server equipped with two quad-core Intel Xeon W5590 processors running at a clock rate of 3.33GHz and 48GB of main memory, though only one thread of SMT solver runs at once. As the SMT solver, we choose z3 4.3.1[6]. The test set of termination problems are the 1463 TRSs from the TRS Standard category of TPDB 8.0.6 [27], and timeout is set to 60s. Details of the experiments are available at `http://www.sakabe.i.is.nagoya-u.ac.jp/~ayamada/PPDP2013/`.

8.1 Results for Orientability

First we evaluated WPO by directly orienting TRSs (Theorems 15, 14, 16 and Corollaries 5, 6). The results are listed in 'Orientability' field of Table 1. The test set is split into two groups; non-duplicating ones (consist of 439 TRSs) and duplicating ones (consist of 1024 TRSs). In the table, 'yes' column indicates the number of successful termination proofs, 'T.O.' indicates the number of timeouts, and 'time' is the total time.

We can see that WPO($\mathcal{MS}um$) is significantly stronger than TKBO and the sequential application of KBO and LPO ('KBO+LPO' row). In addition, WPO($\mathcal{S}um^+$) and WPO($\mathcal{MS}um^+$) are reasonably efficient enhancements to KBO. This efficiency is due to the fact that the estimation of $\mathsf{NSP}_{\mathcal{S}um^+}$ significantly reduces formulas generated for recursive checks.

8.2 Results for Dependency Pairs

Second we evaluated WPO in the DP framework (Theorem 19). 'As DP processors' field in Table 1 compares the power of orders when used as reduction pair processors.

On the contrary to the direct orientability experiment, WPO($\mathcal{S}um$) simply outperforms KBO and WPO($\mathcal{S}um^+$) without loosing efficiency in this setting. This is because the encodings of KBO and WPO($\mathcal{S}um^+$) with argument filter require LPO-like recursive comparison in order to incorporate with the case a term *collapses* to its argument when an argument filtering is considered. In addition, KBO needs extra constraints that correspond to the admissibility.

Theorems 8, 10 and 11 ensures that WPO subsumes KBO, TKBO and LPO even in the DP framework. On the other hand, Theorem 9 does *not* imply that WPO subsumes POLO in the DP framework. This is because the weak part of POLO, i.e. $\geq_{\mathcal{A}}$ is not subsumed by the weak part of WPO($\mathcal{A}$), i.e. $\succsim^{\pi}_{\mathrm{WPO}(\mathcal{A})}$. Nonetheless, WPO remains stronger than POLO when the number of successes is considered.

[6] `http://z3.codeplex.com/`

Table 1: Experimental Results

order	algebra	Orientability						As DP processors		
		439 non-dup. TRSs			1024 dup. TRSs			1463 TRSs		
		yes	T.O.	time	yes	T.O.	time	yes	T.O.	time
POLO	$\mathcal{S}um$	41	0	4.82	–	–	–	455	0	114.30
POLO	$\mathcal{MS}um$	41	0	4.41	19	0	28.93	469	1	301.36
KBO		102	0	5.33	–	–	–	439	3	1083.92
LPO		90	0	31.85	90	0	39.03	444	3	726.29
KBO+LPO		121	0	35.35	90	0	47.71	493	4	1327.73
WPO	$\mathcal{S}um^+$	126	0	6.09	–	–	–	447	3	1104.32
WPO	$\mathcal{S}um$	135	0	46.38	–	–	–	457	2	981.05
WPO	$\mathcal{M}ax$	109	0	56.97	125	0	53.56	490	4	1196.66
WPO	$\mathcal{MS}um^+$	126	0	6.10	132	0	79.23	517	7	1720.98
WPO	$\mathcal{MS}um$	135	0	50.50	138	0	68.95	525	6	1285.41
POLO	$\mathcal{P}ol$	104	3	200.71	21	11	1074.35	486	28	3438.89
POLO	$\mathcal{MP}ol$	104	3	200.92	39	8	615.71	484	19	2365.89
TKBO		125	3	225.46	27	12	1458.79	455	84	11435.84
WPO	$\mathcal{P}ol$	149	3	286.01	29	11	1535.78	471	50	6513.37
WPO	$\mathcal{MP}ol$	149	3	286.18	138	9	1035.11	527	31	4792.18

Table 2: In Combination

strategy	yes	T.O.	time
existing	565	5	1012.86
with WPO	569	5	1082.44

8.3 Combination with Existing Orders

One of the benefits of DP framework is that DP processors can be combined. Table 2 estimates the impact of our contribution when it is combined with existing orders. The strategy indicated by 'existing' sequentially applies POLO($\mathcal{S}um$), POLO($\mathcal{MS}um$), LPO and KBO in this order. The strategy 'with WPO' applies WPO($\mathcal{MS}um$) instead of KBO. In this setting, WPO adds only four successful termination proof in the TPDB problems. However, two of them (`Transformed_CSR_04/Ex3_2_Luc97_Z.trs` and `Ex5_7_Luc97_Z.trs`) are not proved by AProVE 2013, TT$_2$ 1.11 and tools that participated in the full-run of the *termination competition 2011*.

9. CONCLUSION AND FUTURE WORK

We have introduced the weighted path order that encompasses KBO, LPO, TKBO and POLO. As instances of WPO, we presented several orders: WPO($\mathcal{S}um$) subsumes KBO, WPO($\mathcal{P}ol$) subsumes POLO and TKBO, WPO($\mathcal{M}ax$) subsumes LPO, WPO($\mathcal{MS}um$) unifies KBO and LPO, and WPO($\mathcal{MP}ol$) unifies all of them.

We also presented SMT encoding techniques for these orders. The orientability problems of WPO($\mathcal{S}um$), WPO($\mathcal{M}ax$) and WPO($\mathcal{MS}um$) are decidable, since they are reduced to satisfiability problems of linear integer arithmetic which is known to be decidable. We verified the usefulness of our orders by experiments both by directly orienting TRSs and in combination with the DP framework.

In order to keep the presentation simple, we did not present WPO with (multiset) *status*. Nonetheless, it is easy to define WPO with status and verify that WPO($\mathcal{S}um$) with status encompasses KBO with status [25], and WPO($\mathcal{M}ax$) with status encompasses RPO.

Even as a DP processor, WPO subsumes KBO, TKBO and LPO. On the other hand, POLO becomes incomparable to WPO. We leave it for future work to overcome this problem.

To combine WPO($\mathcal{S}um$) and WPO($\mathcal{M}ax$), we considered a straightforward method using 'weight statuses', and moreover heuristically fixed the weight status. It should be interesting to search for other possible weight statuses, or to find more sophisticated combination of max-polynomials such as $f_{\mathcal{A}}(x, y, z) = x + \max(y, z)$.

Note that RPOLO has strength in its higher order version [4]. It might be interesting to apply their techniques to extend WPO for higher order case.

Finally, let us point another direction of extending WPO: unifying with the *matrix interpretation method* [13, 8]. This goal is apparently challenging, since a matrix interpretation is not weakly simple in general. To illustrate this, consider the following TRS from [8]:

$$\mathcal{R}_6 := \{\ \mathsf{f}(\mathsf{f}(x)) \to \mathsf{f}(\mathsf{g}(\mathsf{f}(x)))$$

$\mathcal{R}_6$ is shown terminating by a matrix interpretation $\mathcal{A}$ s.t.

$$\mathsf{f}_{\mathcal{A}}(\vec{x}) = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \cdot \vec{x} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad \mathsf{g}_{\mathcal{A}}(\vec{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \cdot \vec{x}$$

However, $\mathcal{A}$ is not weakly simple. For example,

$$\mathsf{g}_{\mathcal{A}}(\begin{pmatrix} 0 \\ 1 \end{pmatrix}) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \not\geq \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Hence to unify the matrix interpretation with WPO, we have to further relax the weak simplicity constraint on interpretations.

Acknowledgments. We thank Sarah Winkler and Aart Middeldorp for discussions and the anonymous reviewers of previous versions of this paper for helpful comments. This work was supported by JSPS KAKENHI #24500012.

10. REFERENCES

[1] T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236(1-2):133–178, 2000.

[2] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

[3] A. M. Ben-Amram and M. Codish. A SAT-based approach to size change termination with global ranking functions. In *Proc. TACAS '08*, volume 4963 of *LNCS*, pages 218–232, 2008.

[4] M. Bofill, C. Borralleras, E. Rodríguez-Carbonell, and A. Rubio. The recursive path and polynomial ordering for first-order and higher-order terms. *Journal of Logic and Computation*, 23(1):263–305, 2012.

[5] M. Codish, J. Giesl, P. Schneider-Kamp, and R. Thiemann. SAT solving for termination proofs with recursive path orders and dependency pairs. *Journal of Automated Reasoning*, 49:53–93, 2012.

[6] M. Codish, P. Schneider-Kamp, V. Lagoon, R. Thiemann, and J. Giesl. SAT solving for argument filterings. In *Proc. LPAR '06*, volume 4246 of *LNCS*, pages 30–44, 2006.

[7] N. Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17(3):279–301, 1982.

[8] J. Endrullis, J. Waldmann, and H. Zantema. Matrix interpretations for proving termination of term rewriting. *Journal of Automated Reasoning*, 40:195–220, 2008.

[9] C. Fuhs, J. Giesl, A. Middeldorp, P. Schneider-Kamp, R. Thiemann, and H. Zankl. SAT solving for termination analysis with polynomial interpretations. In *Proc. SAT '07*, volume 4501 of *LNCS*, pages 340–354, 2007.

[10] C. Fuhs, J. Giesl, A. Middeldorp, P. Schneider-Kamp, R. Thiemann, and H. Zankl. Maximal termination. In *Proc. RTA '08*, volume 5117 of *LNCS*, pages 110–125, 2008.

[11] J. Giesl, R. Thiemann, P. Schneider-Kamp, and S. Falke. Mechanizing and improving dependency pairs. *Journal of Automated Reasoning*, 37(3):155–203, 2006.

[12] N. Hirokawa and A. Middeldorp. Automating the dependency pair method. *Information and Computation*, 199(1,2):172–199, 2005.

[13] D. Hofbauer and J. Waldmann. Termination of string rewriting with matrix interpretations. In *Proc. RTA '06*, volume 4098 of *LNCS*, pages 328–342, 2006.

[14] S. Kamin and J.-J. Lévy. Two generalizations of the recursive path ordering, 1980. Unpublished note.

[15] D. Knuth and P. Bendix. Simple word problems in universal algebras. In *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

[16] A. Koprowski and J. Waldmann. max/plus tree automata for termination of term rewriting. *Acta Cybernetica*, 19(2):357–392, 2009.

[17] K. Korovin and A. Voronkov. An AC-compatible Knuth-Bendix order. In *Proc. CADE '03*, volume 2741 of *LNAI*, pages 47–59, 2003.

[18] K. Korovin and A. Voronkov. Orienting rewrite rules with the Knuth-Bendix order. *Information and Computation*, 183(2):165–186, 2003.

[19] L. Kovács, G. Moser, and A. Voronkov. On transfinite Knuth-Bendix orders. In *Proc. CADE '11*, volume 6803 of *LNAI*, pages 384–399, 2011.

[20] K. Kusakari, M. Nakamura, and Y. Toyama. Argument filtering transformation. In *Proc. PPDP '99*, volume 1702 of *LNCS*, pages 47–61, 1999.

[21] D. Lankford. On proving term rewrite systems are Noetherian. Technical Report MTP-3, Louisiana Technical University, 1979.

[22] P. Lescanne. Computer experiments with the REVE term rewriting system generator. In *Proc. POPL '83*, pages 99–108. ACM, 1983.

[23] M. Ludwig and U. Waldmann. An extension of the Knuth-Bendix ordering with LPO-like properties. In *Proc. LPAR '07*, volume 4790 of *LNAI*, pages 348–362, 2007.

[24] A. Middeldorp and H. Zantema. Simple termination of rewrite systems. *Theoretical Computer Science*, 175(1):127–158, 1997.

[25] J. Steinbach. Extensions and comparison of simplification orders. In *Proc. RTA '89*, volume 355 of *LNCS*, pages 434–448, 1989.

[26] TeReSe. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[27] TPDB. The termination problem data base. `http://termination-portal.org/wiki/TPDB`.

[28] S. Winkler, H. Zankl, and A. Middeldorp. Ordinals and Knuth-Bendix orders. In *Proc. LPAR '11*, volume 7180 of *LNCS Advanced Research in Computing and Software Science*, pages 420–434, 2012.

[29] H. Zankl, N. Hirokawa, and A. Middeldorp. Constraints for argument filterings. In *Proc. SOFSEM '07*, volume 4362 of *LNCS*, pages 579–590, 2007.

[30] H. Zankl, N. Hirokawa, and A. Middeldorp. KBO orientability. *Journal of Automated Reasoning*, 43(2):173–201, 2009.

[31] H. Zantema. Termination of term rewriting: interpretation and type elimination. *Journal of Symbolic Computation*, 17(1):23–50, 1994.

[32] H. Zantema. The termination hierarchy for term rewriting. *Applicable Algebra in Engineering, Communication and Computing*, 12:3–19, 2001.