Partial Status for KBO

Akihisa Yamada[1], Keiichirou Kusakari[1], and Toshiki Sakabe[1]

1 **Graduate School of Information Science, Nagoya University, Japan**

Abstract

We propose an extension of the Knuth-Bendix order (KBO) called KBO with partial status. A standard status indicates permutation of arguments to each function symbol, but we extend them to allow some arguments to be ignored. This idea is similar to the argument filtering, but benefits of these methods are independent and hence can be combined. In addition, we introduce further refinements of KBO that become possible by partial status. Significance of the proposed method is verified through experiments.

1998 ACM Subject Classification F.4.2 Grammars and Other Rewriting Systems

Keywords and phrases term rewriting, termination, Knuth-Bendix order

1 Introduction

Reduction orders are used to prove termination of term rewrite systems (TRSs). The *Knuth-Bendix order* (KBO) [3] is a classical example of reduction orders.

The *dependency pair (DP) framework* (e.g. [2]) significantly enhances the method of reduction orders. In the DP framework, dependencies between rewrite rules are analyzed. Then each cycle of dependency is shown to be *finite* using a *reduction pair* $\langle \gtrsim, \succ \rangle$, which is typically designed from a reduction order by applying *argument filtering*. However, argument filtering is not always helpful as the following example illustrates:

▶ **Example 1.** Consider the following set of constraints:

$$\mathsf{F}(\mathsf{s}(x)) \succ \mathsf{F}(\mathsf{p}(\mathsf{s}(x))) \qquad \mathsf{p}(\mathsf{s}(x)) \gtrsim x$$

In order to satisfy the first constraint by KBO (or any other *simplification order*), the argument of p must be filtered. However, the second constraint cannot be satisfied under such an argument filtering.

In this note, we propose a reduction pair that can satisfy the above constraints by generalizing KBO *with status* [5]. Usually, a status assigns a new position to every argument of a function symbol. When defining a reduction pair, however, not every argument must be assigned a new position, but some may be ignored. We say such a status is *partial*. The difference between a partial status and an argument filter with standard (i.e. total) status may look subtle; indeed, a trivial definition of LPO with partial status should be subsumed by LPO with argument filters and total status. On the other hand, KBO benefits from partial status because of weights of ignored arguments, which would be lost if those arguments were *filtered* beforehand by an argument filter. Indeed, the constraints in Example 1 are satisfied by KBO with partial status defined in Section 2. We further introduce two refinements that become possible using partial status. Then we demonstrate the significance of our approach through experiments.

2 KBO with Partial Status

Below we define the notion of partial status and the KBO reduction pair.

▶ **Definition 2.** A *partial status function* σ is a mapping that assigns for n-ary symbol f a list $[i_1, \dots, i_{n'}]$ of distinct positions in $\{1, \dots, n\}$.

We write $\vec{s}_{\sigma(f)}$ to denote the sequence $s_{i_1}, \dots, s_{i_{n'}}$, where $\sigma(f) = [i_1, \dots, i_{n'}]$.

▶ **Definition 3** (KBO with partial status)**.** Let $\gtrsim_{\mathcal{F}}$ be a quasi-precedence, σ a partial status function and $\langle w, w_0 \rangle$ a *weight function*, i.e. $w : \mathcal{F} \to \mathbb{N}$, $w_0 > 0$ and $w(c) \geq w_0$ for every constant $c \in \mathcal{F}$. The weight $w(s)$ of a term s is defined as usual:

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

The *Knuth-Bendix order pair* $\langle \succsim_{\mathrm{KBO}}, \succ_{\mathrm{KBO}} \rangle$ is defined recursively as follows: $s \succsim(\text{resp. } \succ)_{\mathrm{KBO}}\ t$ 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 either
   a. $s = f_1(\dots f_k(t) \dots)$, $\sigma(f_1) = \dots = \sigma(f_k) = [1]$ and $t \in \mathcal{V}$ for some $k \geq (\text{resp. } >)\ 0$, or
   b. $s = f(\vec{s}_n)$, $t = g(\vec{t}_m)$ and either
      i. $f >_{\mathcal{F}} g$, or
      ii. $f \sim_{\mathcal{F}} g$ and $[\vec{s}_{\sigma(f)}] \succsim(\text{resp. } \succ)^{\mathsf{lex}}_{\mathrm{KBO}} [\vec{t}_{\sigma(g)}]$.

Here $\succ^{\mathsf{lex}}_{\mathrm{KBO}}$ denotes the lexicographic extension of $\succ_{\mathrm{KBO}}$ modulo $\succsim_{\mathrm{KBO}}$.

The major difference to the standard KBO (e.g. [6]) is case (**2a**), where we exclude the case if $\sigma(f_i) = [\,]$ for some f_i. Because of this modification, the *admissibility* constraint of KBO can be eased as follows:

▶ **Definition 4.** A weight function w is said to be *admissible for* $\gtrsim_{\mathcal{F}}$ *and* σ iff every unary symbol f s.t. $w(f) = 0$ and $\sigma(f) = [1]$ is greatest in $\gtrsim_{\mathcal{F}}$, i.e. $f \gtrsim_{\mathcal{F}} g$ for every $g \in \mathcal{F}$.

In the remainder of this note, we always assume admissibility. Note that a unary symbol f of weight 0 need not be greatest in $\gtrsim_{\mathcal{F}}$, if $\sigma(f) = [\,]$.

▶ **Example 5.** Consider again the constraints in Example 1. Suppose w, $>_{\mathcal{F}}$ and σ satisfy $w(\mathsf{s}) > w(\mathsf{p}) = 0$, $\sigma(\mathsf{s}) = [1]$, $\sigma(\mathsf{p}) = [\,]$, and $\mathsf{s} >_{\mathcal{F}} \mathsf{p}$. Then, $\mathsf{F}(\mathsf{s}(x)) \succ_{\mathrm{KBO}} \mathsf{F}(\mathsf{p}(\mathsf{s}(x)))$ because of cases (**2b–ii**) and (**2b–i**), and $\mathsf{p}(\mathsf{s}(x)) \succsim_{\mathrm{KBO}} x$ because of case (**1**).

Note that in the above example, it also holds that $\mathsf{s}(x) \succ_{\mathrm{KBO}} \mathsf{p}(\mathsf{s}(x))$. Hence, $\succ_{\mathrm{KBO}}$ is not a simplification order anymore, or not even a reduction order. Nonetheless, we can show the following result which is sufficient for the DP framework:

▶ **Theorem 6.** *The KBO pair* $\langle \succsim_{\mathrm{KBO}}, \succ_{\mathrm{KBO}} \rangle$ *is a reduction pair.*

Due to lack of space, we only present a proof for well-foundedness of $\succ_{\mathrm{KBO}}$. We prove the following auxiliary lemma first:

▶ **Lemma 7.** *If* $\vec{s}_{\sigma(f)} \in \mathsf{SN}(\succ_{\mathrm{KBO}})$ *and* $s \succ_{\mathrm{KBO}} t$*, then* $t \in \mathsf{SN}(\succ_{\mathrm{KBO}})$.

Proof. By induction on the quadruple $\langle w(s), f, [\vec{s}_{\sigma(f)}], |t| \rangle$, which is ordered by the lexicographic composition of $>$, $>_{\mathcal{F}}$, $\succ^{\mathsf{lex}}_{\mathrm{KBO}}$ and $>$. Trivially, it is sufficient to consider $t = g(\vec{t}_m)$. Let $[j_1, \dots, j_{m'}] = \sigma(g)$.

- Suppose $w(s) > w(t)$. Then we have $w(s) > w(t) \geq w(t_{j_k})$ and hence $s \succ_{\mathrm{KBO}} t_{j_k}$ for every $k \in \{1, \dots, m'\}$. By the induction hypothesis on the fourth component, we obtain $t_{j_k} \in \mathsf{SN}(\succ_{\mathrm{KBO}})$. Thus for arbitrary u s.t. $t \succ_{\mathrm{KBO}} u$, the induction hypothesis on the first component yields $u \in \mathsf{SN}(\succ_{\mathrm{KBO}})$.
- Suppose $w(s) = w(t)$. First we show $t_{j_k} \in \mathsf{SN}(\succ_{\mathrm{KBO}})$ for every $k \in \{1, \dots, m'\}$. It is trivial if no such k exists, i.e. if $\sigma(g) = [\,]$. Hence suppose $\sigma(g) \neq [\,]$.
  - If $w(t) = w(t_{j_k})$, then g must be unary with $w(g) = 0$ and $\sigma(g) = [1]$. Because of the admissibility, only case (**2b–ii**) can be applied for $s \succ_{\mathrm{KBO}} g(t_1) = t$. Hence, we obtain $s_{i_1} \gtrsim_{\mathrm{KBO}} t_1$ and thus $t_1 \in \mathsf{SN}(\succ_{\mathrm{KBO}})$, since $s_{i_1} \in \mathsf{SN}(\succ_{\mathrm{KBO}})$.
  - If $w(t) > w(t_{j_k})$, then $s \succ_{\mathrm{KBO}} t_{j_k}$ by case (**1**). By the induction hypothesis on the fourth component, $t_j \in \mathsf{SN}(\succ_{\mathrm{KBO}})$.

  Now let us consider arbitrary u s.t. $t \succ_{\mathrm{KBO}} u$. Since we have either $f >_{\mathcal{F}} g$ or $f \sim_{\mathcal{F}} g$ and $[\vec{s}_{\sigma(f)}] \succ^{\mathsf{lex}}_{\mathrm{KBO}} [\vec{t}_{\sigma(g)}]$, $\langle w(s), f, [\vec{s}_{\sigma(f)}], |t|\rangle$ is greater than $\langle w(t), g, [\vec{t}_{\sigma(g)}], |u|\rangle$. Hence, the induction hypothesis yields $u \in \mathsf{SN}(\succ_{\mathrm{KBO}})$. ◀

▶ **Lemma 8.** *The relation $\succ_{\mathrm{KBO}}$ is well-founded.*

Proof. Let us show $s \in \mathsf{SN}(\succ_{\mathrm{KBO}})$ for every term s by induction on $|s|$. Suppose $s = f(\vec{s}_n) \succ_{\mathrm{KBO}} t$. By the induction hypothesis, we have $\vec{s}_n \in \mathsf{SN}(\succ_{\mathrm{KBO}})$ and thus $\vec{s}_{\sigma(f)} \in \mathsf{SN}(\succ_{\mathrm{KBO}})$. Hence by Lemma 7, we get $t \in \mathsf{SN}(\succ_{\mathrm{KBO}})$. ◀

3 Refinements

In this section, we refine $\gtrsim_{\mathrm{KBO}}$ in order to encompass the *polynomial order (POLO)* that is induced by the weight function.

▶ **Definition 9.** The *empty status function* is the partial status σ s.t. $\sigma(f) = [\,]$ for all $f \in \mathcal{F}$.

KBO induced by the quasi-precedence $\gtrsim_{\mathcal{F}} = \mathcal{F}^2$ and the empty status is quite similar to POLO induced by the interpretation $\mathcal{A}$: $f_{\mathcal{A}}(\vec{x}_n) = w(f) + \sum_{i=1}^{n} x_i$. However, the latter is slightly more powerful; the constraint $x \gtrsim \mathsf{p}(x)$ can be satisfied by POLO s.t. $\mathsf{p}_{\mathcal{A}}(x) = x$, but the weak part of KBO cannot satisfy this constraint even if $w(\mathsf{p}) = 0$.

In [6], $\gtrsim_{\mathrm{KBO}}$ is refined s.t. $x \gtrsim_{\mathrm{KBO}} c$ for a minimal constant c. In our setting, a similar refinement can be applied for non-constants:

▶ **Proposition 10.** *Let $s \in \mathcal{V}$ and $t = g(\vec{t}_m)$ s.t.*
- *$|t|_s \leq 1$ and $|t|_x = 0$ for every $x \in \mathcal{V} \setminus \{s\}$,*
- *$w(t) = w_0$,*
- *g is minimal w.r.t. $\gtrsim_{\mathcal{F}}$, and*
- *$\sigma(g) = [\,]$.*

Then for any term $s' = f(\vec{s}_n)$, $s' \gtrsim_{\mathrm{KBO}} t[s \mapsto s']$. ◀

Hence, we refine $s \gtrsim_{\mathrm{KBO}} t$ by adding the following subcase for case (**2**) of Definition 3 (note that the first two conditions above are already satisfied in case (**2**)):

c. $s \gtrsim_{\mathrm{KBO}} t$ if $s \in \mathcal{V}$ and $t = g(\vec{t}_m)$ s.t. g is minimal w.r.t. $\gtrsim_{\mathcal{F}}$ and $\sigma(g) = [\,]$.

▶ **Example 11.** Consider the following set of constraints:

$$\mathsf{F}(\mathsf{s}(x), y) \succ \mathsf{F}(\mathsf{p}(\mathsf{s}(x)), \mathsf{p}(y)) \qquad \mathsf{F}(x, \mathsf{s}(y)) \gtrsim \mathsf{F}(\mathsf{p}(x), \mathsf{p}(\mathsf{s}(y))) \qquad \mathsf{p}(\mathsf{s}(x)) \gtrsim x$$

Let $\sigma(\mathsf{p}) = [\,]$, $\sigma(\mathsf{F}) = [1]$, $w(\mathsf{s}) > w(\mathsf{p}) = 0$ and p be minimal w.r.t. $\gtrsim_{\mathcal{F}}$. As analogous to Example 5, the first and the third constraints are satisfied. For the second constraint, it yields $x \gtrsim \mathsf{p}(x)$, for which case (**c**) of the refined $\gtrsim_{\text{KBO}}$ applies. Note that the argument of p cannot be filtered by an argument filter, because of the third constraint. Hence the refinement of [6] does not work for this example.

We can also refine $\gtrsim_{\text{KBO}}$ when the right-hand side is a variable.

▶ **Proposition 12.** *Let $s = f(\vec{s}_n)$ and $t \in \mathsf{Var}(s)$ s.t.*
- *$\sigma(f) = [\,]$, and*
- *for any $g \in \mathcal{F}$, $f \gtrsim_{\mathcal{F}} g$ if $\sigma(g) = [\,]$ and $f >_{\mathcal{F}} g$ otherwise.*

Then for any term $t' = g(\vec{t}_m)$, $s[t \mapsto t'] \gtrsim_{\text{KBO}} t'$. ◀

Hence, we refine $s \gtrsim_{\text{KBO}} t$ by adding the following subcase for case (**2**):

d. $s \gtrsim_{\text{KBO}} t$ if $s = f(\vec{s}_n)$ and $t \in \mathcal{V}$ s.t. $\sigma(f) = [\,]$ and for any $g \in \mathcal{F}$, $f \gtrsim_{\mathcal{F}} g$ if $\sigma(g) = [\,]$ and $f >_{\mathcal{F}} g$ otherwise.

It is easy to prove the following result:

▶ **Theorem 13.** *Let $\langle w, w_0 \rangle$ be a weight function, σ the empty status function and $\gtrsim_{\mathcal{F}} = \mathcal{F}^2$. Then the refined KBO is equivalent to POLO[1] induced by the carrier set $\{n \geq w_0\}$ and the interpretation $f_{\mathcal{A}}(\vec{x}_n) := w(f) + \sum_{i=1}^{n} x_i$.* ◀

▶ **Example 14.** Consider the following set of constraints:

$$\mathsf{F}(\mathsf{g}(\mathsf{h}(x))) \succ \mathsf{F}(\mathsf{h}(\mathsf{g}(\mathsf{g}(\mathsf{h}(\mathsf{h}(x)))))) \qquad\qquad \mathsf{g}(\mathsf{h}(x)) \gtrsim x$$

Because of the second constraint, arguments of g and h cannot be filtered. Then the first constraint requires $w(\mathsf{g}) = w(\mathsf{h}) = 0$ and moreover one of the following alternatives to hold:
- $\sigma(\mathsf{g}) = \sigma(\mathsf{h}) = [\,]$ and $\mathsf{g} >_{\mathcal{F}} \mathsf{h}$: In this case, the second constraint can be satisfied only if $\gtrsim_{\text{KBO}}$ is refined by case (**d**).
- $\sigma(\mathsf{h}) = [1]$ and $\mathsf{g} >_{\mathcal{F}} \mathsf{h}$: This case is not admissible.
- $\sigma(\mathsf{g}) = [1]$, $\sigma(\mathsf{h}) = [\,]$ and $\mathsf{g} \gtrsim_{\mathcal{F}} \mathsf{h}$: In this case the second constraint cannot be satisfied.

Hence the set of constraints can be satisfied by KBO with partial status only if it is refined by case (**d**). Note that POLO (and LPO) cannot satisfy the set of constraints, since the first rule is not simply terminating and neither g nor h may have 0-coefficient.

4 Experiments and Future Work

We implemented our method via an SMT encoding that extends [9]. 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. We also implemented *usable rules* w.r.t. argument filters following the encoding proposed in [1].

The experiments[2] 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`. The test set of termination problems are the 1463 TRSs from the TRS Standard category of TPDB 8.0.6[3] and 1315 from the SRS Standard category. Timeout is set to 60 seconds.

[1] Note that $c_{\mathcal{A}} \geq w_0 > 0$ is required for every constant c.
[2] Detailed results are available at `http://www.sakabe.i.is.nagoya-u.ac.jp/~ayamada/WST2013/`.
[3] The Termination Problems Data Base. `http://termination-portal.org/wiki/TPDB`.

Table 1 Experiments

Method	Solo			Combination			Combination (SRS)		
	yes	T.O.	time	yes	T.O.	time	yes	T.O.	time
KBO	439	3	1079.23	565	5	1008.69	117	467	28654.39
Def. 3	463	4	1255.77	583	5	1036.46	120	468	28780.43
Prop. 10	464	4	1305.19	584	5	1044.68	120	468	28853.21
Prop. 12	464	4	1275.01	584	5	1036.92	121	474	29356.51
Prop. 10+12	465	4	1323.83	585	5	1049.84	121	474	29380.88

In Table 1, the 'Method' field indicates the reduction pair processor used. 'KBO' row is the standard KBO with (total) status and 'Def. 3' is the KBO with partial status. 'Prop. 10' and 'Prop. 12' applies the refinement of Proposition 10 and Proposition 12, resp. All the methods are with quasi-precedences, argument filters and usable rules.

'Solo' field only applies the reduction pair processor indicated by the 'Method' field. Partial status gives measurable increase in the number of successes (indicated by 'yes' field), though the efficiency is affected ('time' field). Each refinement of Section 3 gains one success with probably acceptable increase in runtime.

'Combination' field applies several reduction pair processors first: It applies the linear POLO (with/without max) with coefficient at most 1, and then LPO with quasi-precedence and status. In this situation partial status becomes more attractive; the increase in number of successes remains measurable, while increase of runtime gets dramatically smaller. For 'Combination (SRS)' field, it applies linear POLO before the indicated processor.

Despite the benefit observed in our experimental implementation, all the TPDB examples our tool proved terminating are also proved terminating by existing termination tools such as AProVE. Our next task is to apply partial status to other extensions of KBO (e.g. [4, 7, 8]) to further increase the number of successful termination proofs.

References

1. 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.
2. J. Giesel, R. Thiemann, P. Schneider-Kamp, and S. Falke. Mechanizing and improving dependency pairs. *Journal of Automated Reasoning*, 37(3):155–203, 2006.
3. D.E. Knuth and P. Bendix. Simple word problems in universal algebras. In *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, New York, 1970.
4. 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.
5. J. Steinbach. Extensions and comparison of simplification orders. In *Proc. RTA '89*, volume 355 of *LNCS*, pages 434–448, 1989.
6. C. Sternagel and R. Thiemann. Formalizing Knuth-Bendix orders and Knuth-Bendix completion. In *Proc. RTA '13*, volume 21 of *LIPIcs*, pages 287–302, 2013.
7. 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.
8. A. Yamada, K. Kusakari, and T. Sakabe. Unifying the Knuth-Bendix, recursive path and polynomial orders. In *Proc. PPDP '13*, 2013. to appear.
9. H. Zankl, N. Hirokawa, and A. Middeldorp. KBO orientability. *Journal of Automated Reasoning*, 43(2):173–201, 2009.