Efficient Formalization of Simplification Orders

René Thiemann and Akihisa Yamada

August 11, 2022

Efficient Formalization of Simplification Orders

René Thiemann
University of Innsbruck, Austria

Akihisa Yamada
National Institute of Advanced Industrial Science and Technology, Japan

Abstract

The weighted path order (WPO) can simulate several simplification orders that are known in term rewriting. By integrating multiset comparisons into WPO, we show that also the recursive path ordering is covered. Moreover, we investigate how refinements of the classical simplification orders can efficiently be integrated: we formally prove the refinements within WPO once and then get them for free for the other simplification orders by the simulation property. Here, the most challenging part was to show that a refined version of the Knuth–Bendix order can actually be simulated by WPO. All of our proofs have been formalized in Isabelle/HOL.

2012 ACM Subject Classification Theory of computation → Logic and verification; Theory of computation → Equational logic and rewriting

Keywords and phrases formalization, Isabelle/HOL, simplification order, termination analysis

Category invited paper

1 Introduction

Automatically proving termination of *term rewrite systems* has been an active field of research for half a century. A number of *simplification orders* [2, 3] are classic methods for proving termination, and these are still integrated in several current termination tools. Classical simplification orders are *Knuth–Bendix orders (KBO)* and *lexicographic* and *recursive path orders (LPO and RPO)*. The *weighted path order (WPO)* [6] was introduced as a simplification order that unifies and extends classical ones.

When switching from theory to implementations in termination tools, limitations of the applied simplification orders become visible while studying non-successful termination proofs. Therefore several refinements of the original definitions of the orders have been developed to make them more applicable and hence more powerful. At this point the question of soundness arises, in particular whether the main properties of a simplification order are still maintained after the integration of the refinements.

To solve this problem we propose to use formal verification, i.e., one should define the orders within a proof assistant such as Coq or Isabelle and then perform the proofs within that system. The advantage is that then re-checking of proofs is quite simple, and in particular a change of a definition (e.g., triggered by some refinement) will immediately point to those parts of the proof which need an adjustment.

The price of using formal verification is its overhead in comparison to a pure proof on paper. In this work we present our approach to perform verification efficiently, namely by exploiting the property that WPO subsumes several simplification orders:

- Instead of formally verifying that KBO, LPO, RPO and WPO are simplification orders, we just prove this fact for WPO and we formally verify that KBO, LPO and RPO are instances of WPO. To this end, we slightly refine WPO itself by permitting multiset comparisons.

- We further show that several refinements of simplification orders are sound for WPO, and hence only have to integrate these refinements into one order, and automatically get the refinements for the other orders, too.

We perform our formalization using Isabelle/HOL, based on IsaFoR, the **Isa**belle **Fo**rmalization of **R**ewriting [5]. As a result of this work we were able to completely remove the formal proofs within IsaFoR that RPO is a simplification order (which entails that LPO is a simplification) order, and we could also remove several formal proofs regarding KBO.

2 Preliminaries

We assume familiarity with term rewriting [1], but briefly recall notions that are used in the following. A *term* built from *signature* $\mathcal{F}$ and set $\mathcal{V}$ of variables is either $x \in \mathcal{V}$ or of form $f(t_1, \ldots, t_n)$, where $f \in \mathcal{F}$ is n-ary and $t_1, \ldots, t_n$ are terms. A *context* C is a term with one hole, and $C[t]$ is the term where the hole is replaced by t. The *subterm relation* $\trianglerighteq$ is defined by $C[t] \trianglerighteq t$. A *substitution* is a function σ from variables to terms, and we write $t\sigma$ for the *instance* of term t in which every variable x is replaced by $\sigma(x)$.

A *reduction pair* is a pair $(\succ, \succsim)$ of two relations on terms that satisfies the following requirements: $\succ$ is well-founded, $\succsim$ and $\succ$ are compatible (i.e., $\succsim \circ \succ \circ \succsim \subseteq \succ$), both are closed under substitutions, and $\succsim$ is closed under contexts. If additionally $\succ$ is transitive, closed under contexts, and contains the strict subterm relation $\triangleright$, then $\succ$ is a *simplification order*. The *trivial reduction pair* is the one where $\succ = \emptyset$ and $\succsim$ relates all terms.

A *quasi-precedence* is a preorder $\geqslant$ on $\mathcal{F}$, such that $> := \geqslant \setminus \leqslant$ is well-founded. A *precedence* is a quasi-precedence where $\geqslant$ is the reflexive closure of $>$.

We use the following notation for common extensions of a pair of relations over terms to pairs of relations over lists of terms.

- $\succ^{\mathsf{mul}}$ and $\succsim^{\mathsf{mul}}$ are the strict- and non-strict order of the *multiset extension* of $(\succ, \succsim)$, where the lists are interpreted as multisets;
- There are two variants of the lexicographic extension: the *unbounded lexicographic extension* is defined as $[s_1, \ldots, s_i, \ldots] \succ^{\mathsf{lex}} [t_1, \ldots, t_i, \ldots]$ iff $s_i \succ t_i$ and $s_j \succsim t_j$ for all $j < i$. The *bounded lexicographic extension* is parametrized by some $b \in \mathbb{N}$, the bound, and it is defined as $[s_1, \ldots, s_n] \succ^{\mathsf{lex},b} [t_1, \ldots, t_m]$ iff $[s_1, \ldots, s_n] \succ^{\mathsf{lex}} [t_1, \ldots, t_m] \wedge (n = m \vee m \le b)$. There are similar definitions for $\succsim^{\mathsf{lex}}$ and $\succsim^{\mathsf{lex},b}$. We sometimes write $\succ^{\mathsf{lex}}$ and $\succsim^{\mathsf{lex}}$ also for the bounded lexicographic extension if the bound is clear from the context.

3 Structure of Simplification Orders

In this section we first define some quite generic relation (a simplified version of WPO) that is a template of several simplification orders, and we will then see how KBO, LPO, RPO and WPO fit into this framework. Moreover, we will also discuss refinements and their soundness.

Let $\geqslant$ be some quasi-precedence. Let $b \in \mathbb{N}$ be some bound which will be used as parameter for bounded lexicographic comparisons in the upcoming definition. Let $\tau : \mathcal{F} \to \{\mathsf{lex}, \mathsf{mul}\}$ be a status. Let *minimal* be some property of constants, such that whenever c is minimal then $f \geqslant c$ for all $f \in \mathcal{F}$. Let $(\succ, \succsim)$ be some reduction pair such that $\succ$ is transitive, $\succsim$ is a preorder, and $C[t] \succsim t$ for all terms t. We define a strict and a non-strict relation on terms ($\succ_{\mathsf{RoT}}$ and $\succsim_{\mathsf{RoT}}$) as follows: $s \succ_{\mathsf{RoT}} t$ iff

1. $s \succ t$, or
2. $s \succsim t$ and

a. $s = f(s_1, \ldots, s_n)$ and $\exists i \in \{1, \ldots, n\}.\ s_i \succsim_{\mathsf{RoT}} t$, or
b. $s = f(s_1, \ldots, s_n)$, $t = g(t_1, \ldots, t_m)$ and
  i. $\forall j \in \{1, \ldots, m\}.\ s \succ_{\mathsf{RoT}} t_j$ and
  ii. A. $f \succ g$ or
   B. $f \succsim g$ and $\tau(f) = \tau(g)$ and $[s_1, \ldots, s_n] \succ_{\mathsf{RoT}}^{\tau(f)} [t_1, \ldots, t_m]$.

The relation $s \succsim_{\mathsf{RoT}} t$ is defined in the same way, where $\succ_{\mathsf{RoT}}^{\tau(f)}$ in case **2.b.ii.B** is replaced by $\succsim_{\mathsf{RoT}}^{\tau(f)}$, and there are two additional subcases in case **2**:

c. $s = x = t$ for some $x \in \mathcal{V}$, or
d. $s = x \in \mathcal{V}$ and $t = c$ for some constant c which is minimal.

Using this generic relation on terms we can now define common instances:

- Classical LPO is obtained by
  - using the trivial reduction pair, so that **1** never applies and the condition in line **2** is always satisfied;
  - requiring a precedence so that **2.b.ii.B** is only applicable if $f = g$, i.e., only lists of the same length are compared;
  - no constant is minimal, so case **2.d** is just dropped;
  - $\tau(f) = \mathsf{lex}$ for all $f \in \mathcal{F}$.
- Classical RPO is like LPO without the requirement on τ.
- Classical KBO is similar to the setup of LPO, but
  - instead of the trivial reduction pair one defines $(\succ, \succsim)$ with the help of weight functions and the multisets of variables of terms;
  - the structure of KBO and of $(\succ_{\mathsf{RoT}}, \succsim_{\mathsf{RoT}})$ is slightly different since in KBO, condition **2.b.i** is not present and case **2.a** is also dropped; moreover in KBO there is one additional case, namely whenever $s \notin \mathcal{V}$, $x \in \mathcal{V}$ and $s \succsim x$, then both $s \succ_{\mathsf{KBO}} x$ and $s \succsim_{\mathsf{KBO}} x$.

▶ Remark. The relation $\succ_{\mathsf{RoT}}$ defined above looks like a simplified form of WPO, e.g., the status function π of WPO (for selecting arguments of each individual function symbol) has been omitted. However, the original WPO does not completely subsume $\succ_{\mathsf{RoT}}$, since the status function τ of $\succ_{\mathsf{RoT}}$ is not included in WPO and one would always compare lists of terms lexicographically in WPO.

Let us now regard two refinements of the classical simplification orders. The first refinement are quasi-precedences. When using quasi-precedences it becomes important to use the bounded version of the lexicographic extension, since otherwise one would be able to construct an infinite sequence $f_0(1) \succ_{\mathsf{RoT}} f_1(0,1) \succ_{\mathsf{RoT}} f_2(0,0,1) \succ_{\mathsf{RoT}} \ldots$ by using a quasi-precendence where $f_i \geqslant f_j$ for all i, j and $f_i > 1 > 0$ for all i. The second refinement are comparisons of the form $x \succsim c$ in case **2.d**. For LPO one requires that c is least in precedence among all symbols, in the same way as in $\succ_{\mathsf{RoT}}$. By contrast, for KBO $f \geqslant c$ is only required for those f which are constants and have weight w_0.

Note that activating both requirements – quasi-precedences and $x \succsim c$ comparisons – is sound for LPO, requires a special definition of lexicographic extensions for KBO, and is unsound for RPO.

▶ **Example 1.** Consider RPO with both refinements, i.e., $(\succ, \succsim)$ is the trivial reduction pair. Let $\geqslant = \mathcal{F} \times \mathcal{F}$ be the trivial precedence where all symbols are equivalent. Let $\tau(c) = \mathsf{lex}$ and $\tau(d) = \mathsf{mul}$ for two constants $c, d \in \mathcal{F}$. Then using case **2.d** we have $x \succsim_{\mathsf{RPO}} c$, but $d \succsim_{\mathsf{RPO}} c$ does not hold. Hence, closure under substitutions is violated.

▶ **Example 2.** Consider a KBO with precedence where all symbols are equivalent, a unary function symbol f with weight 0, and arbitrary symbols g_i with arity $i > 1$. Then $f(x) \succ_{\mathsf{KBO}} x$; however, for $f(g_i(t_1, \dots, t_i)) \succ_{\mathsf{KBO}} g_i(t_1, \dots, t_i)$ (closure under substitutions), only case **2.b.ii.B** is applicable, i.e., one needs lexicographic comparisons $[g_i(t_1, \dots, t_i)] \succ^{\mathsf{lex}}_{\mathsf{KBO}} [t_1, \dots, t_i]$ with lists of arbitrary lengths, i.e., unbounded lexicographic comparisons, which usually destroy well-foundedness in combination with unbounded arities.

The problem of Example 1 is easily fixed by just adding one more alternative to **2.b.ii**:

2. b. ii. C. $f \succsim g$ and $\tau(f) \neq \tau(g)$ and $m = 0$ (and $n > 0$ for $s \succ_{\mathsf{RoT}} t$)

That $(\succ_{\mathsf{RoT}}, \succsim_{\mathsf{RoT}})$ really forms a reduction pair with this fix has been formally proven. Actually, we have formalized an extended version of $\succ_{\mathsf{RoT}}$ that also includes the other features of WPO, i.e., a status function $\pi : \mathcal{F} \to \mathbb{N}^*$ and Refinements (2c) and (2d) of WPO [6, Section 4.2], and it is available in the archive of formal proofs [4]. It is the same definition as if one would take the WPO definition of [6], add multiset comparisons via a status $\tau : \mathcal{F} \to \{\mathsf{lex}, \mathsf{mul}\}$, and add case **2.b.ii.C** for symbols with different status.

▶ **Theorem 3.** $(\succ_{\mathsf{RoT}}, \succsim_{\mathsf{RoT}})$ *is a reduction pair and* $\succ_{\mathsf{RoT}}$ *is a simplification order.*

4 Simulating Classical Simplification Orders

In the previous section we have already seen that LPO and RPO are just instances of the WPO (assuming a definition of WPO that includes the status function τ). This covers quasi-precedences and the $x \succsim c$ refinement. However, such a relationship is not yet established for KBO with refinements. In particular there are three major differences:

1. minimal constants in KBO are defined differently than in WPO,
2. there is a different syntactic structure, and
3. WPO uses the bounded lexicographic extension, but KBO uses the unbounded extension.

We will address these problems and show how properties of WPO can be transferred to KBO.

1. Recall that in KBO a constant c is minimal if $f \geqslant c$ for all constants f of weight w_0, whereas in WPO $f \geqslant c$ is required for all $f \in \mathcal{F}$. We solve this problem by changing the quasi-precedence $\geqslant$ of KBO into some quasi-precedence $\geqslant'$ in a way that
   - KBO-minimal constants w.r.t. $\geqslant$ are WPO-minimal w.r.t. $\geqslant'$, and
   - $\succsim_{\mathsf{KBO}}$ and $\succ_{\mathsf{KBO}}$ are unmodified when switching from $\geqslant$ to $\geqslant'$.
2. For the syntactic differences, we prove that they do not affect the defined relations.
   - The additional case $f(C[x]) \succ_{\mathsf{KBO}} x$ of KBO can be simulated since $\succ_{\mathsf{RoT}}$ is a simplification order.
   - Assume that $f(s_1, \dots, s_n) \succ_{\mathsf{KBO}} f(t_1, \dots, t_m)$ was shown by **2.b**. Here we use some properties of KBO to conclude $f(s_1, \dots, s_n) \succ_{\mathsf{KBO}} t_j$ for all $1 \leq j \leq m$. Hence, it does not matter whether the condition in **2.b.i** – which does not occur in the original KBO definition – is added to the KBO definition.

- The definition of KBO does not contain case **2.a**. However, as in the previous step we utilize the property of KBO that the corresponding inference rule $s_i \succsim_{\mathsf{KBO}} t \longrightarrow f(s_1, \ldots, s_n) \succ_{\mathsf{KBO}} t$ is still valid for all $i \in \{1, \ldots, n\}$.

Note that for the equivalence proof we already use some properties of KBO, i.e., these must be proven before we are able to transfer properties of $\succ_{\mathsf{RoT}}$ to KBO.

3. One cannot replace the unbounded lexicographic extension by a bounded one if function symbols of unbounded arity are considered. However, whenever terms s and t are compared, only finitely many symbols appear in s and t, and thus there is the maximum arity b among them. For these terms there is no difference in whether b-bounded or unbounded lexicographic extension is used.

We arrive at the following result.

▶ **Theorem 4.** *Let a KBO with quasi-precedence $\geqslant$ and some bound b be given. Then a reduction pair (encoding the weight-function) and quasi-precedence $\geqslant'$ can be constructed as parameters to $\succ_{\mathsf{RoT}}$ and $\succsim_{\mathsf{RoT}}$ (or to WPO), such that $(s \succ_{\mathsf{KBO}} t) \longleftrightarrow (s \succ_{\mathsf{RoT}} t)$ and $(s \succsim_{\mathsf{KBO}} t) \longleftrightarrow (s \succsim_{\mathsf{RoT}} t)$ for all terms s, t whose function symbols have arity below b.*

▶ **Corollary 5.** *For every KBO over a finite signature there exists an equivalent WPO.*

▶ **Corollary 6.** *KBO is transitive, closed under substitutions and well-founded.*

Proof. Consider the set of terms $\{s, t, u, s\sigma, t\sigma\}$, and define b as the maximum arity that occurs within these terms. From Theorem 3 we conclude $s \succ_{\mathsf{RoT}} t \succ_{\mathsf{RoT}} u \longrightarrow s \succ_{\mathsf{RoT}} u$ and $s \succ_{\mathsf{RoT}} t \longrightarrow s\sigma \succ_{\mathsf{RoT}} t\sigma$. By Theorem 4 and the choice of b, transitivity and closure under substitutions of KBO are proved.

For well-foundedness of KBO, consider an infinite sequence $t_1 \succ_{\mathsf{KBO}} t_2 \succ_{\mathsf{KBO}} \ldots$. Define b' as the weight of t_1. Hence b' is larger than the weight of all terms in the sequence. Since the weight is an upper bound for the arities, b' is also larger than the arities of all t_i. Thus, by Theorem 4 we know $t_1 \succ_{\mathsf{RoT}} t_2 \succ_{\mathsf{RoT}} \ldots$ in contradiction to Theorem 3. ◀

As future work it remains to be clarified whether the addition of multiset comparisons to WPO will improve the power of automated termination tools.

References

1 Franz Baader and Tobias Nipkow. *Term rewriting and all that.* Cambridge University Press, 1998. `doi:10.1017/CBO9781139172752`.

2 Nachum Dershowitz. Orderings for term-rewriting systems. *Theor. Comput. Sci.*, 17:279–301, 1982. `doi:10.1016/0304-3975(82)90026-3`.

3 Donald E. Knuth and Peter Bendix. Simple word problems in universal algebras. In *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, New York, 1970. `doi:10.1016/B978-0-08-012975-4.50028-X`.

4 Christian Sternagel, René Thiemann, and Akihisa Yamada. A formalization of weighted path orders and recursive path orders. *Archive of Formal Proofs*, 2021. `https://isa-afp.org/entries/Weighted_Path_Order.html`, Formal proof development.

5 René Thiemann and Christian Sternagel. Certification of termination proofs using CeTA. In *Theorem Proving in Higher Order Logics, 22nd International Conference, TPHOLs 2009, Munich, Germany, August 17-20, 2009. Proceedings*, volume 5674 of *Lecture Notes in Computer Science*, pages 452–468. Springer, 2009. `doi:10.1007/978-3-642-03359-9_31`.

6 Akihisa Yamada, Keiichirou Kusakari, and Toshiki Sakabe. A unified ordering for termination proving. *Sci. Comput. Program.*, 111:110–134, 2015. `doi:10.1016/j.scico.2014.07.009`.