A Transfinite Knuth–Bendix Order for Lambda-Free Higher-Order Terms

Heiko Becker[1], Jasmin Christian Blanchette[2,3,4(✉)], Uwe Waldmann[4], and Daniel Wand[4,5]

[1] Max-Planck-Institut für Softwaresysteme, Saarbrücken, Germany
hbecker@mpi-sws.org
[2] Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
j.c.blanchette@vu.nl
[3] Inria Nancy – Grand Est, Villers-lès-Nancy, France
jasmin.blanchette@inria.fr
[4] Max-Planck-Institut für Informatik, Saarbrücken, Germany
{jasmin.blanchette,uwe.waldmann,daniel.wand}@mpi-inf.mpg.de
[5] Technische Universität München, Munich, Germany
daniel.wand@in.tum.de

Abstract. We generalize the Knuth–Bendix order (KBO) to higher-order terms without λ-abstraction. The restriction of this new order to first-order terms coincides with the traditional KBO. The order has many useful properties, including transitivity, the subterm property, compatibility with contexts (monotonicity), stability under substitution, and well-foundedness. Transfinite weights and argument coefficients can also be supported. The order appears promising as the basis of a higher-order superposition calculus.

1 Introduction

Superposition [38] is one of the most successful proof calculi for first-order logic today, but in contrast to resolution [8,25], tableaux [4], and connections [1], it has not yet been generalized to higher-order logic (also called simple type theory). Yet, most proof assistants and many specification languages are based on some variant of higher-order logic. Tools such as HOLyHammer and Sledgehammer [12] encode higher-order constructs to bridge the gap, but their performance on higher-order problems is disappointing [44].

This motivates us to design a *graceful* generalization of superposition: a proof calculus that behaves like standard superposition on first-order problems and that smoothly scales up to arbitrary higher-order problems. The calculus should additionally be complete with respect to Henkin semantics [9, 22]. A challenge is that superposition relies on a simplification order, which is fixed in advance of the proof attempt, to prune the search space. However, no simplification order $>$ exists on higher-order terms viewed modulo β-equivalence; the cycle $\mathsf{a} =_\beta (\lambda x.\, \mathsf{a})\ (\mathsf{f}\,\mathsf{a}) > \mathsf{f}\,\mathsf{a} > \mathsf{a}$ is a counterexample. (The two $>$ steps follow from the subterm property—the requirement that proper subterms of a term are smaller than the term itself.) A solution is to give up interchangeability of β-equivalent terms, or even inclusion of β-reduction (i.e., $(\lambda x.\, s[x])\ t > s[t]$).

We start our investigations by focusing on a fragment devoid of λ-abstractions. A λ-free higher-order term is either a variable x, a symbol f, or an application $s\,t$. Application associates to the left. Functions take their arguments one at a time, in a curried style (e.g., $\mathsf{f\,a\,b}$). Compared with first-order terms, the main differences are that variables may be applied (e.g., $x\,\mathsf{a}$) and that functions may be supplied fewer arguments than they can take. Although λ-abstractions are widely perceived as the higher-order feature par excellence, they can be avoided by letting the proof calculus, and provers based on it, synthesize fresh symbols f and definitions $\mathsf{f}\,x_1\,\ldots\,x_m \approx t$ as needed, giving arbitrary names to otherwise nameless functions.

In recent work, we introduced a "graceful" λ-free higher-order recursive path order (RPO) [15]. We now contribute a corresponding Knuth–Bendix order (KBO) [29]. Leading superposition provers such as E [40], SPASS [47], and Vampire [33] implement both KBO and variants of RPO. KBO's main strength is that it tends to consider syntactically smaller terms as smaller, making it a robust option on a wide range of problems. To keep the presentation manageable, we introduce three KBO variants of increasing strength (Sect. 4): a basic KBO ($>_{\mathsf{hb}}$); a KBO with support for function symbols of weight 0 ($>_{\mathsf{hz}}$); and a KBO extended with coefficients for the arguments ($>_{\mathsf{hc}}$). They all coincide with their first-order counterparts on terms that contain only fully applied function symbols and no applied variables. For all three variants, we allow different comparison methods for comparing the arguments of different symbols (Sect. 2). In addition, we allow ordinals for the weights and argument coefficients (Sect. 3), as in the transfinite first-order KBO [36].

Our KBO variants enjoy many useful properties, including transitivity, the subterm property, stability under substitution, well-foundedness, and totality on ground terms (Sect. 5). The orders with no argument coefficients also enjoy compatibility with contexts (sometimes called monotonicity), thereby qualifying as simplification orders. Even without this property, we expect the orders to be usable in a λ-free higher-order generalization of superposition, possibly at the cost of some complications [18]. Ground totality is used in the completeness proof of superposition. The proofs of the properties were formalized using the Isabelle/HOL proof assistant (Sect. 6).

Although this is not our primary focus, the new KBO can be used to establish termination of higher-order term rewriting systems (Sect. 7). However, more research will be necessary to combine the order with the dependency pair framework, implement them in a termination prover, and evaluate them on standard term-rewriting benchmarks.

To our knowledge, KBO has not been studied before in a higher-order setting. There are, however, a considerable number of higher-order variants of RPO [16, 17, 27, 30, 31, 34] and many encodings of higher-order term rewriting systems into first-order systems [2, 21, 23, 23, 45]. The encoding approaches are more suitable to term rewriting systems than to superposition and similar proof calculi. We refer to our paper on the λ-free higher-order RPO for a discussion of such related work [15].

Conventions. We fix a set $\mathcal{V}$ of *variables* with typical elements x, y. A higher-order signature consists of a nonempty set Σ of (function) *symbols* $\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{f}, \mathsf{g}, \mathsf{h}, \ldots$. Untyped λ-free higher-order (Σ-)*terms* $s, t, u \in \mathcal{T}_\Sigma$ $(= \mathcal{T})$ are defined inductively by the grammar $s ::= x \mid \mathsf{f} \mid t\,u$. These terms are isomorphic to applicative terms [28], but we prefer the "higher-order" terminology. Symbols and variables are assigned an arity, $\mathit{arity} : \Sigma \uplus$

$\mathcal{V} \to \mathbb{N} \cup \{\infty\}$, specifying their maximum number of arguments. Infinite arities are allowed for the sake of generality. Nullary symbols are called *constants.* A term of the form $t\ u$ is called an *application*. Non-application terms $\zeta, \xi, \chi \in \Sigma \uplus \mathcal{V}$ are called *heads*. A term s can be decomposed uniquely as a head with m arguments: $s = \zeta\ s_1\ \ldots\ s_m$. We define $\mathit{hd}(s) = \zeta$, $\mathit{args}(s) = (s_1, \ldots, s_m)$, and $\mathit{arity}(s) = \mathit{arity}(\zeta) - m$.

The *size* $|s|$ of a term is the number of grammar rule applications needed to construct it. The set of *subterms* of a term consists of the term itself and, for applications $t\ u$, of the subterms of t and u. The multiset of variables occurring in a term s is written $\mathit{vars}_{\#}(s)$—e.g., $\mathit{vars}_{\#}(\mathsf{f}\ x\ y\ x) = \{x, x, y\}$. We denote by $M(a)$ the multiplicity of an element a in a multiset M and write $M \subseteq N$ to mean $\forall a.\ M(a) \leq N(a)$.

We assume throughout that the arities of all symbols and variables occurring in terms are respected—in other words, all subterms of a term have nonnegative arities. A *first-order* signature is a higher-order signature with an arity function $\mathit{arity} : \Sigma \to \mathbb{N}$. A *first-order* term is a term in which variables are nullary and heads are applied to the number of arguments specified by their respective arities. Following the view that first-order logic is a fragment of higher-order logic, we will use a curried syntax for first-order terms. Accordingly, if $\mathit{arity}(\mathsf{a}) = 0$ and $\mathit{arity}(\mathsf{f}) = 2$, then $\mathsf{f}\ \mathsf{a}\ \mathsf{a}$ is first-order, whereas f, $\mathsf{f}\ \mathsf{a}$, and $\mathsf{f}\ \mathsf{f}\ \mathsf{f}$ are only higher-order.

Our focus on untyped terms is justified by a desire to keep the definitions simple and widely applicable to a variety of type systems (monomorphic, rank-1 polymorphic, dependent types, etc.). There are straightforward ways to extend the results presented in this paper to a typed setting: Types can be simply erased, they can be encoded in the terms, or they can be used to break ties when two terms are identical except for their types. Even in an untyped setting, the *arity* function makes some of the typing information visible. In Sect. 4.3, we will introduce a mapping, called ghd, that can be used to reveal more information about the typing discipline if desired.

2 Extension Orders

KBO relies on an extension operator to recurse through tuples of arguments—typically, the lexicographic order [3, 50]. We prefer an abstract treatment, in a style reminiscent of Ferreira and Zantema [20], which besides its generality has the advantage that it emphasizes the peculiarities of our higher-order setting.

Let $A^* = \bigcup_{i=0}^{\infty} A^i$ be the set of tuples (or finite lists) of arbitrary length whose components are drawn from a set A. We write its elements as $(a_1, \ldots, a_m)$, where $m \geq 0$, or simply $\bar{a}$. The empty tuple is written $()$. Singleton tuples are identified with elements of A. The number of components of a tuple $\bar{a}$ is written $|\bar{a}|$. Given an m-tuple $\bar{a}$ and an n-tuple $\bar{b}$, we denote by $\bar{a} \cdot \bar{b}$ the $(m+n)$-tuple consisting of the concatenation of $\bar{a}$ and $\bar{b}$.

Given a function $h : A \to A$, we let $h(\bar{a})$ stand for the componentwise application of h to $\bar{a}$. Abusing notation, we sometimes use a tuple where a set is expected, ignoring the extraneous structure. Moreover, since all our functions are curried, we write $\zeta\ \bar{s}$ for a curried application $\zeta\ s_1\ \ldots\ s_m$, without risk of ambiguity.

Given a relation $>$, we write $<$ for its inverse (i.e., $a < b \Leftrightarrow b > a$) and $\geq$ for its reflexive closure (i.e., $b \geq a \Leftrightarrow b > a \vee b = a$), unless $\geq$ is defined otherwise. A (strict) partial order is a relation that is irreflexive (i.e., $a \not> a$) and transitive (i.e., $c > b \wedge$

$b > a \Longrightarrow c > a$). A (strict) total order is a partial order that satisfies totality (i.e., $b \geq a \vee a > b$). A relation $>$ is well founded if and only if there exists no infinite chain of the form $a_0 > a_1 > \cdots$.

For any relation $> \subseteq A^2$, let $\gg \subseteq (A^*)^2$ be a relation on tuples over A. For example, $\gg$ could be the lexicographic or multiset extension of $>$. We assume throughout that if $B \subseteq A$, then the extension $\gg_B$ of the restriction $>_B$ of $>$ to elements from B coincides with $\gg$ on $(B^*)^2$. Moreover, the following properties are essential for all the orders defined later, whether first- or higher-order:

X1. *Monotonicity*: $\bar{b} \gg_1 \bar{a}$ implies $\bar{b} \gg_2 \bar{a}$ if $b >_1 a$ implies $b >_2 a$ for all a, b;
X2. *Preservation of stability*: $\bar{b} \gg \bar{a}$ implies $h(\bar{b}) \gg h(\bar{a})$ if
 (1) $b > a$ implies $h(b) > h(a)$ for all a, b, and
 (2) $>$ is a partial order on the range of h;
X3. *Preservation of irreflexivity*: $\gg$ is irreflexive if $>$ is irreflexive;
X4. *Preservation of transitivity*: $\gg$ is transitive if $>$ is irreflexive and transitive;
X5. *Modularity* ("head or tail"):
 if $>$ is transitive and total, $|\bar{a}| = |\bar{b}|$, and $b \cdot \bar{b} \gg a \cdot \bar{a}$, then $b > a$ or $\bar{b} \gg \bar{a}$;
X6. *Compatibility with tuple contexts*: $a \neq b$ and $b > a$ implies $\bar{c} \cdot b \cdot \bar{d} \gg \bar{c} \cdot a \cdot \bar{d}$.

Some of the conditions in X2, X4, X5, and X6 may seem gratuitous, but they are necessary for some extension operators if the relation $>$ is arbitrary. For KBO, $>$ will always be a partial order, but we cannot assume this until we have proved it.

It may seem as though X2 is a consequence of X1, by letting $>_1$ be $>$ and $b >_2 a \Longleftrightarrow h(b) > h(a)$. However, $\bar{b} \gg_2 \bar{a}$ does not generally coincide with $h(\bar{b}) \gg h(\bar{a})$, even if $>$ satisfies X1. A counterexample follows: Let $\gg$ be the Huet–Oppen multiset extension as introduced below (Definition 6), and let $\bar{a} = \mathsf{d}$, $\bar{b} = (\mathsf{c}, \mathsf{c})$, $h(\mathsf{c}) = h(\mathsf{d}) = \mathsf{c}$, and $\mathsf{d} > \mathsf{c}$. Then $\bar{b} \gg_2 \bar{a}$ (i.e., $(\mathsf{c}, \mathsf{c}) \gg_2 \mathsf{d}$) is false, whereas $h(\bar{b}) \gg h(\bar{a})$ (i.e., $(\mathsf{c}, \mathsf{c}) \gg \mathsf{c}$) is true.

The remaining properties of $\gg$ will be required only by some of the orders or for some optional properties of $>$:

X7. *Preservation of totality*: $\gg$ is total if $>$ is total;
X8. *Compatibility with prepending*: $\bar{b} \gg \bar{a}$ implies $a \cdot \bar{b} \gg a \cdot \bar{a}$;
X9. *Compatibility with appending*: $\bar{b} \gg \bar{a}$ implies $\bar{b} \cdot a \gg \bar{a} \cdot a$;
X10. *Minimality of empty tuple*: $a \gg ()$.

Property X5, modularity, is useful to establish well-foundedness of $\gg$ from the well-foundedness of $>$. The argument is captured by Lemma 3, which builds on Lemmas 1 and 2.

Lemma 1. *For any well-founded total order $> \subseteq A^2$, let $\gg \subseteq (A^*)^2$ be a partial order that satisfies property X5. The restriction of $\gg$ to n-tuples is well founded.*

Proof. By induction on n. The base case is trivial. For the induction step, we assume that there exists an infinite descending chain of n-tuples $\bar{x}_0 \gg \bar{x}_1 \gg \cdots$ and show that this leads to a contradiction. Let $\bar{x}_i = x_i \cdot \bar{y}_i$. For each link $\bar{x}_i \gg \bar{x}_{i+1}$ in the chain, property X5 guarantees that (1) $x_i > x_{i+1}$ or (2) $\bar{y}_i \gg \bar{y}_{i+1}$. Since $>$ is well founded, there can be at most finitely many consecutive links of the first kind. Exploiting the transitivity of $\gg$, we can eliminate all such links, resulting in an infinite chain made up of links of the second kind. The existence of such a chain implies the existence of an infinite chain of $(n-1)$-tuples $\bar{y}_{i_1} \gg \bar{y}_{i_2} \gg \cdots$, contradicting the induction hypothesis. □

Lemma 2. *For any well-founded total order $> \subseteq A^2$, let $\gg \subseteq (A^*)^2$ be a partial order that satisfies property X5. The restriction of $\gg$ to tuples with at most n components is well founded.*

Proof. By induction on n. The base case is trivial. For the induction step, we assume that there exists a chain $\bar{x}_0 \gg \bar{x}_1 \gg \cdots$ involving tuples with at most n components and show that this leads to a contradiction.

We call a tuple *bad* if it belongs to an infinite descending $\gg$-chain involving tuples with at most n components. Without loss of generality, we may assume that the tuple $\bar{x}_0$ has a minimal number of components among all bad tuples and that $\bar{x}_{i+1}$ has a minimal number of components among all bad tuples $\bar{y}$ such that $\bar{x}_i \gg \bar{y}$.

If there exists no index k such that $|\bar{x}_k| = n$, we can directly invoke the induction hypothesis to finish the proof. Otherwise, we have not only $|\bar{x}_k| = n$ for some k but also $|\bar{x}_{k+1}| = n$ as well, due to the minimality of $|\bar{x}_k|$, and likewise for all indices beyond $k+1$. This means that there exists an infinite chain $\bar{x}_k \gg \bar{x}_{k+1} \gg \cdots$ involving only n-tuples, contradicting Lemma 1. □

Lemma 3 (Bounded Preservation of Well-Foundedness). *For any well-founded partial order $> \subseteq A^2$, let $\gg \subseteq (A^*)^2$ be a partial order that satisfies properties X1 and X5. The restriction of $\gg$ to tuples with at most n components is well founded.*

Proof. By Zorn's lemma, let $>'$ be a well-founded total order that extends $>$. By property X1, $\gg \subseteq \gg'$. By Lemma 2, $\gg'$ is well founded; hence, $\gg$ is well founded. □

We now define the extension operators and study their properties.

Definition 4. The *lexicographic extension* $\gg^{\mathsf{lex}}$ of the relation $>$ is defined recursively by $() \not\gg^{\mathsf{lex}} \bar{a}$, $b \cdot \bar{b} \gg^{\mathsf{lex}} ()$, and $b \cdot \bar{b} \gg^{\mathsf{lex}} a \cdot \bar{a} \iff b > a \lor b = a \land \bar{b} \gg^{\mathsf{lex}} \bar{a}$.

The reverse, or right-to-left, lexicographic extension is defined analogously. The left-to-right operator lacks property X9; a counterexample is $\bar{b} = \mathsf{c}$, $\bar{a} = ()$, and $a = \mathsf{d}$, with $\mathsf{d} > \mathsf{c}$—we then have $\mathsf{c} \gg^{\mathsf{lex}} ()$ and $(\mathsf{c}, \mathsf{d}) \not\gg^{\mathsf{lex}} \mathsf{d}$. Correspondingly, the right-to-left operator lacks X8. The other properties are straightforward to prove.

Definition 5. The *length-lexicographic extension* $\gg^{\mathsf{llex}}$ of the relation $>$ is defined by $\bar{b} \gg^{\mathsf{llex}} \bar{a} \iff |\bar{b}| > |\bar{a}| \lor |\bar{b}| = |\bar{a}| \land \bar{b} \gg^{\mathsf{lex}} \bar{a}$.

The length-lexicographic extension and its right-to-left counterpart satisfy all of the properties listed above, making them more interesting than the plain lexicographic extensions. We can also apply arbitrary permutations on same-length tuples before comparing them lexicographically; however, the resulting operators generally fail to satisfy properties X8 and X9.

Definition 6. The *multiset extension* $\gg^{\mathsf{ms}}$ of the relation $>$ is defined by $\bar{b} \gg^{\mathsf{ms}} \bar{a} \iff A \neq B \land \forall x.\ A(x) > B(x) \Longrightarrow \exists y > x.\ B(y) > A(y)$, where A and B are the multisets corresponding to $\bar{a}$ and $\bar{b}$, respectively.

The above multiset extension, due to Huet and Oppen [24], satisfies all properties except X7. Dershowitz and Manna [19] give an alternative formulation that is equivalent for partial orders $>$ but exhibits subtle differences if $>$ is an arbitrary relation. In particular, the Dershowitz–Manna order does not satisfy property X3, making it unsuitable for establishing that KBO variants are partial orders. This, in conjunction with our desire to track requirements precisely, explains the many subtle differences between this section and the corresponding section of our paper about RPO [15]. One of the main differences is that instead of property X5, the definition of RPO requires preservation of well-foundedness, which unlike X5 is not satisfied by the lexicographic extension.

Finally, we consider the componentwise extension of relations to pairs of tuples of the same length. For partial orders $>$, this order underapproximates any extension that satisfies properties X4 and X6. It also satisfies all properties except X7 and X10.

Definition 7. The *componentwise extension* $\gg^{\mathsf{cw}}$ of the relation $>$ is defined so that $(b_1,\ldots,b_n) \gg^{\mathsf{cw}} (a_1,\ldots,a_m)$ if and only if $m = n$, $b_1 \geq a_1$, …, $b_m \geq a_m$, and $b_i > a_i$ for some $i \in \{1,\ldots,m\}$.

3 Ordinals

The transfinite KBO [36] allows weights and argument coefficients to be ordinals instead of natural numbers. We restrict our attention to the ordinals below ε_0. We call these the *syntactic ordinals* **O**. They are precisely the ordinals that can be expressed in Cantor normal form, corresponding to the grammar $\alpha ::= \sum_{i=1}^{m} \omega^{\alpha_i} k_i$, where $\alpha_1 > \cdots > \alpha_m$ and $k_i \in \mathbb{N}_{>0}$ for $i \in \{1,\ldots,m\}$. We refer to the literature for the precise definition [32, 36]. The ordinals subsume the natural numbers: 0 corresponds to the $m = 0$ case of the grammar rule, and $n \in \mathbb{N}_{>0}$ is obtained by taking $m = 1$, $\alpha_1 = 0$, and $k_1 = n$. We let $\mathbf{O}_{>0} = \mathbf{O} - \{0\}$.

The traditional sum and product operations are not commutative—e.g., $1 + \omega = \omega \neq \omega + 1$. For the transfinite KBO, the Hessenberg (or natural) sum and product are used instead. These operations are commutative and coincide with the sum and product operations on polynomials over ω. Somewhat nonstandardly, we let $+$ and $\cdot$ (or juxtaposition) denote these operators. It is sometimes convenient to use subtraction on ordinals and to allow polynomials over ω in which some of the coefficients may be negative (but all of the ω exponents are always plain ordinals). We call such polynomials *signed* (*syntactic*) *ordinals* **ZO**. One way to define $\alpha > \beta$ on signed ordinals is to look at the sign of the leading coefficient of $\alpha - \beta$. Which coefficient is leading depends recursively on $>$. The relation $>$ is total for signed ordinals. Its restriction to plain ordinals is well founded.

Here is a list of properties that hold for α, β, γ ranging over signed ordinals:

1. $\alpha + \beta = \beta + \alpha$;
2. $(\alpha + \beta) + \gamma = \alpha + (\beta + \gamma)$;
3. $\alpha\beta = \beta\alpha$;
4. $(\alpha\beta)\gamma = \alpha(\beta\gamma)$;
5. $(\alpha + \beta)\gamma = \alpha\gamma + \beta\gamma$;
6. $\beta > \alpha \Leftrightarrow \beta + \gamma > \alpha + \gamma$;
7. $0 + \alpha = \alpha$;
8. $0\alpha = 0$;
9. $1\alpha = \alpha$;
10. $\beta \geq \alpha \Leftrightarrow \beta + 1 > \alpha$;
11. $\alpha\beta = 0 \Leftrightarrow \alpha = 0 \vee \beta = 0$;
12. $\beta > \alpha \wedge \gamma > 0 \Longrightarrow \gamma\beta > \gamma\alpha$;
13. $\alpha - \beta + \beta = \alpha$;
14. $\alpha - \beta + \gamma = \alpha + \gamma - \beta$;
15. $\alpha + \beta - \gamma = \alpha + (\beta - \gamma)$;
16. $\alpha - \beta - \gamma = \alpha - (\beta + \gamma)$.

Unlike the plain ordinals, the signed ordinals possess the following convenient property: For two signed ordinals α,β such that $\beta > \alpha$, there exists a signed ordinal γ (namely, $\beta - \alpha$) such that $\alpha + \gamma = \beta$.

4 Term Orders

This section presents five orders: the standard first-order KBO (Sect. 4.1), the applicative KBO (Sect. 4.2), and our three λ-free higher-order KBO variants (Sects. 4.3 to 4.5). The orders are stated with ordinal weights for generality. The occurrences of $\mathbf{O}$ and $\mathbf{O}_{>0}$ below can be consistently replaced by $\mathbb{N}$ and $\mathbb{N}_{>0}$ if desired.

For finite signatures, we can restrict the weights to be ordinals below ω^{ω^ω} without loss of generality [32]. Indeed, for proving termination of term rewriting systems that are finite and known in advance, transfinite weights are not necessary at all [49]. In the context of superposition, though, the order must be chosen in advance, before the saturation process generates the terms to be compared, and moreover their number can be unbounded; therefore, the latter result does not apply.

4.1 The Standard First-Order KBO

What we call the "standard first-order KBO" is more precisely a transfinite KBO on first-order terms with different argument comparison methods (or "statuses") but without argument coefficients. Despite the generalizations, our formulation is similar to Zantema's [50] and Baader and Nipkow's [3].

Definition 8. Let $\succ$ be a well-founded total order, or *precedence*, on Σ, let $\varepsilon \in \mathbb{N}_{>0}$, let $w : \Sigma \to \mathbf{O}$, and for any $> \subseteq \mathcal{T}^2$ and any $\mathsf{f} \in \Sigma$, let $\gg^{\mathsf{f}} \subseteq (\mathcal{T}^*)^2$ be a relation that satisfies properties X1–X6. For each constant $\mathsf{c} \in \Sigma$, assume $w(\mathsf{c}) \geq \varepsilon$. If $w(\iota) = 0$ for some unary $\iota \in \Sigma$, assume $\iota \succeq \mathsf{f}$ for all $\mathsf{f} \in \Sigma$. Let $\mathcal{W} : \mathcal{T} \to \mathbf{O}_{>0}$ be defined recursively by

$$\mathcal{W}(\mathsf{f}(s_1, \ldots, s_m)) = w(\mathsf{f}) + \textstyle\sum_{i=1}^{m} \mathcal{W}(s_i) \qquad\qquad \mathcal{W}(x) = \varepsilon$$

The induced (*standard*) *Knuth–Bendix order* $>_{\mathsf{fo}}$ on first-order Σ-terms is defined inductively so that $t >_{\mathsf{fo}} s$ if $\mathit{vars}_{\#}(t) \supseteq \mathit{vars}_{\#}(s)$ and any of these conditions is met:

F1. $\mathcal{W}(t) > \mathcal{W}(s)$;
F2. $\mathcal{W}(t) = \mathcal{W}(s)$, $t \neq x$, and $s = x$;
F3. $\mathcal{W}(t) = \mathcal{W}(s)$, $t = \mathsf{g}\,\bar{t}$, $s = \mathsf{f}\,\bar{s}$, and $\mathsf{g} \succ \mathsf{f}$;
F4. $\mathcal{W}(t) = \mathcal{W}(s)$, $t = \mathsf{f}\,\bar{t}$, $s = \mathsf{f}\,\bar{s}$, and $\bar{t} \gg^{\mathsf{f}}_{\mathsf{fo}} \bar{s}$.

The inductive definition is legitimate by the Knaster–Tarski theorem owing to the monotonicity of $\gg^{\mathsf{f}}$ (property X1).

Given two terms, KBO first compares their weights ($\mathcal{W}$), derived from the weights of their symbols (w). For terms with equal weights, KBO tries to break the tie by comparing the head symbols using a precedence $\succ$ or, if the head symbols are identical, by comparing the argument tuples using an extension operator. Variables are a special concern because their true weight is not known until they have been instantiated with a ground term. KBO assigns them the minimum weight possible for a term, ε, and ensures that there are at least as many occurrences of each variable on the greater side as on the smaller side, through the condition $\mathit{vars}_\#(t) \supseteq \mathit{vars}_\#(s)$.

Constants must have a weight of at least ε. One *special* unary symbol, denoted by ι, is allowed to have a weight of 0 if it has the maximum precedence. Rule F2 can be used to compare variables x with terms of the form $\iota^m\, x$, i.e., $\iota\,(\iota\,(\cdots(\iota\,x)\cdots))$, with $m > 0$ occurrences of ι.

The more recent literature defines KBO as a mutually recursive pair consisting of a strict order $>_{\mathsf{fo}}$ and a quasiorder $\gtrsim_{\mathsf{fo}}$ [43]. This approach yields a slight increase in precision, but that comes at the cost of substantial duplication in the proof development and appears to be largely orthogonal to the issues that interest us.

4.2 The Applicative KBO

One way to use standard first-order term orders on λ-free higher-order terms is to encode the latter using the *applicative encoding*: Make all symbols nullary and represent application by a distinguished binary symbol @. For KBO, the precedence $\succ$ and the weight w must be extended to consider @. A natural choice is to make @ the least element of $\succ$ and to assign it a weight of 0. Because @ is the only symbol that is ever applied, $\gg^{@}$ is the only relevant member of the $\gg$ family. This means that it is impossible to use the lexicographic extension for some symbols and the multiset extension for others. Moreover, the applicative encoding is incompatible with refinements such as symbols of weight 0 (Sect. 4.4) and argument coefficients (Sect. 4.5).

Definition 9. Let Σ be a higher-order signature, and let $\Sigma' = \Sigma \uplus \{@\}$ be a first-order signature in which all symbols belonging to Σ are assigned arity 0 and @ is assigned arity 2. The *applicative encoding* $[\![\ ]\!] : \mathcal{T}_\Sigma \to \mathcal{T}_{\Sigma'}$ is defined recursively by the equations $[\![\zeta]\!] = \zeta$ and $[\![s\ t]\!] = @\ [\![s]\!]\ [\![t]\!]$.

Assuming that @ has the lowest precedence and weight 0, the composition of the first-order KBO with the encoding $[\![\ ]\!]$ can be formulated directly as follows.

Definition 10. Let $\succ$ be a precedence on Σ, let $\varepsilon \in \mathbb{N}_{>0}$, let $w : \Sigma \to \mathbf{O}_{\geq\varepsilon}$, and for any $> \subseteq \mathcal{T}^2$, let $\gg \subseteq (\mathcal{T}^*)^2$ be a relation that satisfies properties X1–X6. Let $\mathcal{W} : \mathcal{T} \to \mathbf{O}_{>0}$ be defined recursively by

$$\mathcal{W}(\mathsf{f}) = w(\mathsf{f}) \qquad \mathcal{W}(x) = \varepsilon \qquad \mathcal{W}(s\ t) = \mathcal{W}(s) + \mathcal{W}(t)$$

The induced *applicative Knuth–Bendix order* $>_{\mathsf{ap}}$ on higher-order Σ-terms is defined inductively so that $t >_{\mathsf{ap}} s$ if $\mathit{vars}_\#(t) \supseteq \mathit{vars}_\#(s)$ and any of these conditions is met:

A1. $\mathcal{W}(t) > \mathcal{W}(s)$;
A2. $\mathcal{W}(t) = \mathcal{W}(s)$ and $t = \mathsf{g} \succ \mathsf{f} = s$;
A3. $\mathcal{W}(t) = \mathcal{W}(s)$, $t = \mathsf{g}$, and $s = s_1\ s_2$;
A4. $\mathcal{W}(t) = \mathcal{W}(s)$, $t = t_1\ t_2$, $s = s_1\ s_2$, and $(t_1, t_2) \gg_{\mathsf{ap}} (s_1, s_2)$.

The applicative KBO works quite differently from the standard KBO, even on first-order terms. Given $t = \mathsf{g}\ t_1\ t_2$ and $s = \mathsf{f}\ s_1\ s_2$, the order $>_{\mathsf{fo}}$ first compares the weights, then g and f, then t_1 and s_1, and finally t_2 and s_2; by contrast, $>_{\mathsf{ap}}$ compares the weights, then $\mathsf{g}\ t_1$ and $\mathsf{f}\ s_1$ (recursively starting with their weights), and finally t_2 and s_2.

Hybrid schemes have been proposed to strengthen the encoding: If a function f always occurs with at least k arguments, these can be passed directly in an uncurried style—e.g., $@\ (\mathsf{f}\ \mathsf{a}\ \mathsf{b})\ x$. However, this relies on a closed-world assumption—namely, that all terms that will ever be compared arise in the input problem. This is at odds with the need for complete higher-order proof calculi to synthesize arbitrary terms during proof search [9], in which a symbol f may be applied to fewer arguments than anywhere in the problem. A scheme by Hirokawa et al. [23] circumvents this issue but requires additional symbols and rewrite rules.

4.3 The Graceful Higher-Order Basic KBO

Our "graceful" higher-order basic KBO exhibits strong similarities with the first-order KBO. It reintroduces the symbol-indexed family of extension operators and consists of three rules B1, B2, and B3 corresponding to F1, F3, and F4. The adjective "basic" indicates that it does not allow symbols of weight 0, which complicate the picture because functions can occur unapplied in our setting. In Sect. 4.4, we will see how to support such symbols, and in Sect. 4.5, we will extend the order further with argument coefficients.

The basic KBO is parameterized by a mapping ghd from variables to nonempty sets of possible ground heads that may arise when instantiating the variables. This mapping is extended to symbols f by taking $\mathit{ghd}(\mathsf{f}) = \{\mathsf{f}\}$. The mapping is said to *respect arities* if, for all variables x, $\mathsf{f} \in \mathit{ghd}(x)$ implies $\mathit{arity}(\mathsf{f}) \geq \mathit{arity}(x)$. In particular, if $\iota \in \mathit{ghd}(\zeta)$, then $\mathit{arity}(\zeta) \leq 1$. A substitution $\sigma : \mathcal{V} \to \mathcal{T}$ *respects* the ghd mapping if for all variables x, we have $\mathit{arity}(x\sigma) \geq \mathit{arity}(x)$ and $\mathit{ghd}(\mathit{hd}(x\sigma)) \subseteq \mathit{ghd}(x)$. This mapping allows us to restrict instantiations, typically based on a typing discipline.

Convention 11. Precedences $\succ$ are extended to arbitrary heads by taking $\xi \succ \zeta \Leftrightarrow \forall \mathsf{g} \in \mathit{ghd}(\xi), \mathsf{f} \in \mathit{ghd}(\zeta).\ \mathsf{g} \succ \mathsf{f}$.

Definition 12. Let $\succ$ be a precedence on Σ following Convention 11, let $\varepsilon \in \mathbb{N}_{>0}$, let $w : \Sigma \to \mathbf{O}_{\geq\varepsilon}$, let $\mathit{ghd} : \mathcal{V} \to \mathcal{P}(\Sigma) - \{\emptyset\}$ be an arity-respecting mapping extended to symbols f by taking $\mathit{ghd}(\mathsf{f}) = \mathsf{f}$, and for any $> \subseteq \mathcal{T}^2$ and any $\mathsf{f} \in \Sigma$, let $\gg^{\mathsf{f}} \subseteq (\mathcal{T}^*)^2$ be a relation that satisfies properties X1–X6 and X8. Let $\mathcal{W} : \mathcal{T} \to \mathbf{O}_{>0}$ be defined recursively by

$$\mathcal{W}(\mathsf{f}) = w(\mathsf{f}) \qquad \mathcal{W}(x) = \varepsilon \qquad \mathcal{W}(s\ t) = \mathcal{W}(s) + \mathcal{W}(t)$$

The induced *graceful basic Knuth–Bendix order* $>_{\mathsf{hb}}$ on higher-order Σ-terms is defined inductively so that $t >_{\mathsf{hb}} s$ if $\mathit{vars}_{\#}(t) \supseteq \mathit{vars}_{\#}(s)$ and any of these conditions is met, where $t = \xi\ \bar{t}$ and $s = \zeta\ \bar{s}$:

B1. $\mathcal{W}(t) > \mathcal{W}(s)$;
B2. $\mathcal{W}(t) = \mathcal{W}(s)$ and $\xi \succ \zeta$;

B3. $\mathcal{W}(t) = \mathcal{W}(s)$, $\xi = \zeta$, and $\bar{t} \gg^{\mathsf{f}}_{\mathsf{hb}} \bar{s}$ for all symbols $\mathsf{f} \in \mathit{ghd}(\zeta)$.

The main differences with the first-order KBO $>_{\mathsf{fo}}$ is that rules B2 and B3 also apply to terms with variable heads and that symbols with weight 0 are not allowed. Property X8, compatibility with prepending, is necessary to ensure stability under substitution: If $x\,\mathsf{b} >_{\mathsf{hb}} x\,\mathsf{a}$ and $x\sigma = \mathsf{f}\,\bar{s}$, we also want $\mathsf{f}\,\bar{s}\,\mathsf{b} >_{\mathsf{hb}} \mathsf{f}\,\bar{s}\,\mathsf{a}$ to hold. Property X9, compatibility with appending, is not required by the definition, but it is necessary to ensure compatibility with a specific kind of higher-order context: If $\mathsf{f\,b} >_{\mathsf{hb}} \mathsf{f\,a}$, we often want $\mathsf{f\,b\,c} >_{\mathsf{hb}} \mathsf{f\,a\,c}$ to hold as well.

Example 13. It is instructive to contrast our new KBO with the applicative order on some examples. Let $\mathsf{h} \succ \mathsf{g} \succ \mathsf{f}$, let $w(\mathsf{f}) = w(\mathsf{g}) = \varepsilon = 1$ and $w(\mathsf{h}) = 2$, let $\gg$ be the length-lexicographic extension (which degenerates to plain lexicographic for $>_{\mathsf{ap}}$), and let $\mathit{ghd}(x) = \Sigma$ for all variables x. In all of the following cases, $>_{\mathsf{hb}}$ disagrees with $>_{\mathsf{ap}}$:

$$\mathsf{f\,f\,f\,(f\,f)} >_{\mathsf{hb}} \mathsf{f\,(f\,f\,f)\,f} \qquad \mathsf{g\,(f\,g)} >_{\mathsf{hb}} \mathsf{f\,g\,f} \qquad \mathsf{g\,(f\,(f\,f))} >_{\mathsf{hb}} \mathsf{f\,(f\,f)\,f}$$
$$\mathsf{h\,h} >_{\mathsf{hb}} \mathsf{f\,h\,f} \qquad \mathsf{h\,(f\,f)} >_{\mathsf{hb}} \mathsf{f\,(f\,f)\,f} \qquad \mathsf{g\,(f}\,x) >_{\mathsf{hb}} \mathsf{f}\,x\,\mathsf{g}$$

Rules B2 and B3 apply in a straightforward, "first-order" fashion, whereas $>_{\mathsf{ap}}$ analyses the terms one binary application at a time. For the first pair of terms, we have $\mathsf{f\,f\,f\,(f\,f)} <_{\mathsf{ap}} \mathsf{f\,(f\,f\,f)\,f}$ because $(\mathsf{f\,f\,f}, \mathsf{f\,f}) \ll^{\mathsf{lex}}_{\mathsf{ap}} (\mathsf{f\,(f\,f\,f)}, \mathsf{f})$. In the presence of variables, some terms are comparable only with $>_{\mathsf{hb}}$ or only with $>_{\mathsf{ap}}$:

$$\mathsf{g\,(g}\,x) >_{\mathsf{hb}} \mathsf{f\,g\,g} \qquad \mathsf{g\,(f}\,x) >_{\mathsf{hb}} \mathsf{f}\,x\,\mathsf{f} \qquad \mathsf{h}\,(x\,y) >_{\mathsf{hb}} \mathsf{f}\,y\,(x\,\mathsf{f})$$
$$\mathsf{f\,f}\,x >_{\mathsf{ap}} \mathsf{g\,(f\,f)} \qquad x\,x\,\mathsf{g} >_{\mathsf{ap}} \mathsf{g\,(g\,g)} \qquad \mathsf{g}\,x\,\mathsf{g} >_{\mathsf{ap}} x\,(\mathsf{g\,g})$$

To apply rule A4 on the first example, we would need $(\mathsf{g}, \mathsf{g}\,x) \gg^{\mathsf{lex}}_{\mathsf{ap}} (\mathsf{f\,g}, \mathsf{g})$, but the term g has a lighter weight than $\mathsf{f\,g}$. The last two examples in the bottom row reveal that the applicative order tends to be stronger when either side is an applied variable.

The quantification over $\mathsf{f} \in \mathit{ghd}(\zeta)$ in rule B3 can be inefficient in an implementation, when the symbols in $\mathit{ghd}(\zeta)$ disagree on which $\gg$ to use. We could generalize the definition of $>_{\mathsf{hb}}$ further to allow underapproximation, but some care would be needed to ensure transitivity. As a simple alternative, we propose instead to enrich all sets $\mathit{ghd}(\zeta)$ that disagree with a distinguished symbol for which the componentwise extension ($\gg^{\mathsf{cw}}_{\mathsf{hb}}$) is used. Since this extension operator is more restrictive than any others, whenever it is present in a set $\mathit{ghd}(\zeta)$, there is no need to compute the others.

4.4 The Graceful Higher-Order KBO

The standard first-order KBO, as introduced by Knuth and Bendix, allows symbols of arity 2 or more to have weight 0. It also allows for a special unary symbol ι of weight 0. Rule F2 makes comparisons $\iota^m\,x >_{\mathsf{fo}} x$ possible, for $m > 0$.

In a higher-order setting, symbols of weight 0 require special care. Functions can occur unapplied, which could give rise to terms of weight 0, violating the basic KBO assumption that all terms have at least weight $\varepsilon > 0$. Our solution is to add a penalty of δ for each missing argument to a function. Thus, even though a *symbol* f may be assigned a weight of 0, the *term* f ends up with a weight of at least $\mathit{arity}(\mathsf{f}) \cdot \delta$. These two notions of weight are distinguished formally as w and $\mathcal{W}$. For the arithmetic to work out, the δ penalty must be added for all missing arguments to all symbols and variables. Symbols and variables must then have a finite arity. For the sake of generality, we allow δ to take any value between 0 and ε, but the special symbol ι is allowed only if $\delta = \varepsilon$, so that $\mathcal{W}(\iota\ s) = \mathcal{W}(s)$. The $\delta = 0$ case coincides with the basic KBO.

Let $\mathit{mghd}(\zeta)$ denote a symbol $\mathsf{f} \in \mathit{ghd}(\zeta)$ such that $w(\mathsf{f}) + \delta \cdot \mathit{arity}(\mathsf{f})$—its weight as a term—is minimal. Clearly, $\mathit{mghd}(\mathsf{f}) = \mathsf{f}$ for all $\mathsf{f} \in \Sigma$, and $\mathit{arity}(\mathit{mghd}(\zeta)) \geq \mathit{arity}(\zeta)$ if ghd respects arities. The intuition is that any instance of the term ζ will have at least weight $w(\mathsf{f}) + \delta \cdot \mathit{arity}(\mathsf{f})$. This property is important for stability under substitution.

Definition 14. Let $\succ$ be a precedence on Σ following Convention 11, let $\varepsilon \in \mathbb{N}_{>0}$, let $\delta \in \{0,\ldots,\varepsilon\}$, let $w : \Sigma \to \mathbf{O}$, let $\mathit{ghd} : \mathcal{V} \to \mathcal{P}(\Sigma) - \{\emptyset\}$ be an arity-respecting mapping extended to symbols f by taking $\mathit{ghd}(\mathsf{f}) = \mathsf{f}$, and for any $> \subseteq \mathcal{T}^2$ and any $\mathsf{f} \in \Sigma$, let $\gg^{\mathsf{f}} \subseteq (\mathcal{T}^*)^2$ be a relation that satisfies properties X1–X6, X8, and, if $\delta = \varepsilon$, X10. For each symbol $\mathsf{f} \in \Sigma$, assume $w(\mathsf{f}) \geq \varepsilon - \delta \cdot \mathit{arity}(\mathsf{f})$. If $w(\iota) = 0$ for some unary $\iota \in \Sigma$, assume $\iota \succeq \mathsf{f}$ for all $\mathsf{f} \in \Sigma$ and $\delta = \varepsilon$. Let $\mathcal{W} : \mathcal{T} \to \mathbf{O}_{>0}$ be defined recursively by $\mathcal{W} : \mathcal{T} \to \mathbf{O}_{>0}$:

$$\mathcal{W}(\zeta) = w(\mathit{mghd}(\zeta)) + \delta \cdot \mathit{arity}(\mathit{mghd}(\zeta)) \qquad \mathcal{W}(s\ t) = \mathcal{W}(s) + \mathcal{W}(t) - \delta$$

If $\delta > 0$, assume $\mathit{arity}(\zeta) \neq \infty$ for all heads $\zeta \in \Sigma \uplus \mathcal{V}$. The induced *graceful* (*standard*) *Knuth–Bendix order* $>_{\mathsf{hz}}$ on higher-order Σ-terms is defined inductively so that $t >_{\mathsf{hz}} s$ if $\mathit{vars}_{\#}(t) \supseteq \mathit{vars}_{\#}(s)$ and any of these conditions is met, where $t = \xi\ \bar{t}$ and $s = \zeta\ \bar{s}$:

Z1. $\mathcal{W}(t) > \mathcal{W}(s)$;
Z2. $\mathcal{W}(t) = \mathcal{W}(s)$, $\bar{t} = t' \geq_{\mathsf{hz}} s$, $\xi \not\succeq \zeta$, $\xi \not\preceq \zeta$, and $\iota \in \mathit{ghd}(\xi)$;
Z3. $\mathcal{W}(t) = \mathcal{W}(s)$ and $\xi \succ \zeta$;
Z4. $\mathcal{W}(t) = \mathcal{W}(s)$, $\xi = \zeta$, and $\bar{t} \gg^{\mathsf{f}}_{\mathsf{hz}} \bar{s}$ for all symbols $\mathsf{f} \in \mathit{ghd}(\zeta)$.

The $>_{\mathsf{hz}}$ order requires minimality of the empty tuple (property X10) if $\delta = \varepsilon$. This ensures that $\iota\ s >_{\mathsf{hz}} \iota$, which is desirable to honor the subterm property. Even though $\mathcal{W}(s)$ is defined using subtraction, given an arity-respecting ghd mapping, the result is always a plain (unsigned) ordinal: Each penalty δ that is subtracted is accounted for in the weight of the head, since $\delta \cdot \mathit{arity}(\mathit{mghd}(\zeta)) \geq \delta \cdot \mathit{arity}(\zeta)$.

Rule Z2 is more complicated than its first-order counterpart F2, because it must cope with cases that cannot arise with first-order terms. The last three conditions of rule Z2 are redundant but make the calculus deterministic, in the sense that at most one rule applies to any pair of terms.

Example 15. The following examples illustrate how ι and variables that can be instantiated by ι behave with respect to $>_{\mathsf{hz}}$. Let $\mathit{arity}(\mathsf{a}) = \mathit{arity}(\mathsf{b}) = 0$, $\mathit{arity}(\mathsf{f}) = \mathit{arity}(\iota) = \mathit{arity}(x) = \mathit{arity}(y) = 1$, $\delta = \varepsilon$, $w(\mathsf{a}) = w(\mathsf{b}) = w(\mathsf{f}) = \varepsilon$, $w(\iota) = 0$, $\iota \succ \mathsf{f} \succ \mathsf{b} \succ \mathsf{a}$, and $\mathit{ghd}(x) = \mathit{ghd}(y) = \Sigma$. The following comparisons hold, where $m > 0$:

$$\begin{array}{cccc}
\iota^m\ \mathsf{f} >_{\mathsf{hz}} \mathsf{f} & \iota^m\ x >_{\mathsf{hz}} x & y^m\ \mathsf{f} >_{\mathsf{hz}} \mathsf{f} & y^m\ x >_{\mathsf{hz}} x \\
\iota^m\ (\mathsf{f}\ \mathsf{a}) >_{\mathsf{hz}} \mathsf{f}\ \mathsf{a} & \iota^m\ (x\ \mathsf{a}) >_{\mathsf{hz}} x\ \mathsf{a} & y^m\ (\mathsf{f}\ \mathsf{a}) >_{\mathsf{hz}} \mathsf{f}\ \mathsf{a} & y^m\ (x\ \mathsf{a}) >_{\mathsf{hz}} x\ \mathsf{a} \\
\iota^m\ (\mathsf{f}\ \mathsf{b}) >_{\mathsf{hz}} \mathsf{f}\ \mathsf{a} & \iota^m\ (x\ \mathsf{b}) >_{\mathsf{hz}} x\ \mathsf{a} & y^m\ (\mathsf{f}\ \mathsf{b}) >_{\mathsf{hz}} \mathsf{f}\ \mathsf{a} & y^m\ (x\ \mathsf{b}) >_{\mathsf{hz}} x\ \mathsf{a}
\end{array}$$

The first column is justified by rule Z3. The remaining columns are justified by rule Z2. The first and second rows of these columns are covered by the $t' = s$ case of rule Z2; the third row is covered by the $t' >_{\mathsf{hz}} s$ case.

4.5 The Graceful Higher-Order KBO with Argument Coefficients

The requirement that variables must occur at least as often in the greater term t than in the smaller term s—$\mathit{vars}_\#(t) \supseteq \mathit{vars}_\#(s)$—drastically restrains KBO. For example, there is no way to compare the terms $\mathsf{f}\,x\,y\,y$ and $\mathsf{g}\,x\,x\,y$, no matter which weights and precedences we assign to f and g.

The literature on transfinite KBO proposes argument (or subterm) coefficients to relax this limitation [32,36], but the idea is independent of the use of ordinals for weights; it has its origin in Otter's ad hoc term order [36, 37]. With each m-ary symbol $\mathsf{f} \in \Sigma$, we associate m positive coefficients: $\mathit{coef}_{\mathsf{f}} : \{1, \ldots, \mathit{arity}(\mathsf{f})\} \to \mathbf{O}_{>0}$. We write $\mathit{coef}(\mathsf{f}, i)$ for $\mathit{coef}_{\mathsf{f}}(i)$. When computing the weight of $\mathsf{f}\,s_1 \ldots s_m$, the weights of the arguments $s_1, \ldots, s_m$ are multiplied with $\mathit{coef}(\mathsf{f}, 1), \ldots, \mathit{coef}(\mathsf{f}, m)$, respectively. The coefficients also affect variable counts; for example, by taking 2 as the coefficient attached to g's third argument, we can make $\mathsf{g}\,x\,x\,y$ larger than $\mathsf{f}\,x\,y\,y$.

Argument coefficients are problematic for applied variables: When computing the weight of $x\,\mathsf{a}$, what coefficient should be applied to a's weight? Our solution is to delay the decision by representing the coefficient as a fixed unknown. Similarly, we represent the weight of a term variable x by an unknown. Thus, given $\mathit{arity}(x) = 1$, the weight of the term $x\,\mathsf{a}$ is a polynomial $\mathbf{w}_x + \mathbf{k}_x \mathcal{W}(\mathsf{a})$ over the unknowns $\mathbf{w}_x$ and $\mathbf{k}_x$. In general, with each variable $x \in \mathcal{V}$, we associate the unknown $\mathbf{w}_x \in \mathbf{O}_{>0}$ and the family of unknowns $\mathbf{k}_{x,i} \in \mathbf{O}_{>0}$ for $i \in \mathbb{N}_{>0}$, $i \le \mathit{arity}(x)$, corresponding to x's weight and argument coefficients, respectively. We let $\mathbf{P}$ denote the polynomials over these unknowns.

We extend w to variable heads, $w : \Sigma \uplus \mathcal{V} \to \mathbf{P}$, by letting $w(x) = \mathbf{w}_x$, and we extend coef to arbitrary terms $s \in \mathcal{T}$, $\mathit{coef}_s : \{1, \ldots, \mathit{arity}(s)\} \to \mathbf{P}$, by having

$$\mathit{coef}(x, i) = \mathbf{k}_{x,i} \qquad\qquad \mathit{coef}(s\,t, i) = \mathit{coef}(s, i+1)$$

The second equation is justified by the observation that the ith argument of the term $s\,t$ is the $(i+1)$st argument of s. Thus, the coefficient that applies to b in $\mathsf{f}\,\mathsf{a}\,\mathsf{b}$ (i.e., the first argument to $\mathsf{f}\,\mathsf{a}$, or the second argument to f) is $\mathit{coef}(\mathsf{f}\,\mathsf{a}, 1) = \mathit{coef}(\mathsf{f}, 2) = \mathbf{k}_{\mathsf{f},2}$.

An assignment A is a mapping from the unknowns to the signed ordinals. (If $\delta = 0$, we can restrict the codomain to the plain ordinals.) The operator $p|_A$ evaluates a polynomial p under an assignment A. An assignment A is *admissible* if $\mathbf{w}_x|_A \ge w(\mathit{mghd}(x))$ and $\mathbf{k}_{x,i}|_A \ge 1$ for all variables x and indices $i \in \{1, \ldots, \mathit{arity}(x)\}$. If there exists an upper bound M on the coefficients $\mathit{coef}(s, i)$, we may also require $\mathbf{k}_{x,i}|_A \le M$. The $M = 1$ case coincides with the standard KBO without argument coefficients.

Given two polynomials p, q, we have $q > p$ if and only if $q|_A > p|_A$ for all admissible assignments A. Similarly, $q \ge p$ if and only if $q|_A \ge p|_A$ for all admissible A.

Definition 16. Let $\succ$ be a precedence on Σ following Convention 11, let $\varepsilon \in \mathbb{N}_{>0}$, let $\delta \in \{0, \ldots, \varepsilon\}$, let $w : \Sigma \to \mathbf{O}$, let $\mathit{coef} : \Sigma \times \mathbb{N}_{>0} \to \mathbf{O}_{>0}$, let $\mathit{ghd} : \mathcal{V} \to \mathcal{P}(\Sigma) - \{\emptyset\}$ be an arity-respecting mapping extended to symbols f by taking $\mathit{ghd}(\mathsf{f}) = \mathsf{f}$, and for any $> \subseteq \mathcal{T}^2$

and any $\mathsf{f} \in \Sigma$, let $\gg^{\mathsf{f}} \subseteq (\mathcal{T}^*)^2$ be a relation that satisfies properties X1–X6, X8, and, if $\delta = \varepsilon$, X10. For each symbol $\mathsf{f} \in \Sigma$, assume $w(\mathsf{f}) \geq \varepsilon - \delta \cdot \mathit{arity}(\mathsf{f})$. If $w(\iota) = 0$ for some unary $\iota \in \Sigma$, assume $\iota \succeq \mathsf{f}$ for all $\mathsf{f} \in \Sigma$ and $\delta = \varepsilon$. Let $\mathcal{W} : \mathcal{T} \to \mathbf{P}$ be defined recursively by

$$\mathcal{W}(\zeta\ s_1 \ldots s_m) = w(\zeta) + \delta \cdot (\mathit{arity}(\mathit{mghd}(\zeta)) - m) + \textstyle\sum_{i=1}^{m} \mathit{coef}(\zeta, i) \cdot \mathcal{W}(s_i)$$

If $\delta > 0$, assume $\mathit{arity}(\zeta) \neq \infty$ for all heads $\zeta \in \Sigma \uplus \mathcal{V}$. The induced *graceful* (*standard*) *Knuth–Bendix order* $>_{\mathsf{hc}}$ *with argument coefficients* on higher-order Σ-terms is defined inductively so that $t >_{\mathsf{hc}} s$ if any of these conditions is met, where $t = \xi\ \bar{t}$ and $s = \zeta\ \bar{s}$:

C1. $\mathcal{W}(t) > \mathcal{W}(s)$;
C2. $\mathcal{W}(t) \geq \mathcal{W}(s)$, $\bar{t} = t' \geq_{\mathsf{hc}} s$, $\xi \not\succeq \zeta$, $\xi \not\preceq \zeta$, and $\iota \in \mathit{ghd}(\xi)$;
C3. $\mathcal{W}(t) \geq \mathcal{W}(s)$ and $\xi \succ \zeta$;
C4. $\mathcal{W}(t) \geq \mathcal{W}(s)$, $\xi = \zeta$, and $\bar{t} \gg^{\mathsf{f}}_{\mathsf{hc}} \bar{s}$ for all symbols $\mathsf{f} \in \mathit{ghd}(\zeta)$.

The weight comparisons amount to nonlinear polynomial constraints over the unknowns, which are interpreted as universally quantified variables. Rules C2–C4 use $\geq$ instead of $=$ because $\mathcal{W}(s)$ and $\mathcal{W}(t)$ cannot always be compared precisely. For example, if $\mathcal{W}(s) = \varepsilon$ and $\mathcal{W}(t) = \mathbf{w}_y$, we might have $\mathcal{W}(t) \geq \mathcal{W}(s)$ but neither $\mathcal{W}(t) > \mathcal{W}(s)$ nor $\mathcal{W}(t) = \mathcal{W}(s)$.

Example 17. Let $\mathit{ghd}(x) = \Sigma$ for all variables x. Argument coefficients allow us to perform these comparisons: $\mathsf{g}\ x >_{\mathsf{hc}} \mathsf{f}\ x\ x$ and $\mathsf{g}\ x >_{\mathsf{hc}} \mathsf{f}\ x\ \mathsf{g}$. By taking $\delta = 0$, $\mathit{coef}(\mathsf{f}, i) = 1$ for $i \in \{1, 2\}$, $\mathit{coef}(\mathsf{g}, 1) = 3$, and $w(\mathsf{f}) = w(\mathsf{g}) = \varepsilon$, we have the constraints $\varepsilon + 3\mathbf{w}_x > \varepsilon + 2\mathbf{w}_x$ and $\varepsilon + 3\mathbf{w}_x > 2\varepsilon + \mathbf{w}_x$. Since $\mathbf{w}_x \geq \varepsilon$, we can apply rule C1 in both cases.

The nonlinear constraints are in general undecidable, but they can be underapproximated in various ways. A simple approach is to associate a fresh unknown with each monomial and systematically replace the monomials by their unknowns.

Example 18. We want to derive $z\ (y\ (\mathsf{f}\ x)) >_{\mathsf{hc}} z\ (y\ x)$ using rule C1. For $\delta = 0$, the constraint is $w(\mathsf{f}) \cdot \mathbf{k}_{z,1}\mathbf{k}_{y,1} + \mathit{coef}(\mathsf{f}, 1) \cdot w(\mathsf{f}) \cdot \mathbf{k}_{z,1}\mathbf{k}_{y,1}\mathbf{w}_z > \mathbf{k}_{z,1}\mathbf{k}_{y,1}\mathbf{w}_z$. It can be underapproximated by the linear constraint $w(\mathsf{f}) \cdot \mathbf{a} + \mathit{coef}(\mathsf{f}, 1) \cdot w(\mathsf{f}) \cdot \mathbf{b} > \mathbf{b}$, which is true given the ranges of the coefficients and unknowns involved.

5 Properties

We now state and prove the main properties of our KBO with argument coefficients, $>_{\mathsf{hc}}$. The proofs carry over easily to the two simpler orders, $>_{\mathsf{hb}}$ and $>_{\mathsf{hz}}$. Many of the proofs are inspired by Baader and Nipkow [3] and Zantema [50].

Theorem 19 (Irreflexivity). *$s \not>_{\mathsf{hc}} s$.*

Proof. By strong induction on $|s|$. Assume $s >_{\mathsf{hc}} s$ and let $s = \zeta\ \bar{s}$. Clearly, due to the irreflexivity of $\succ$, the only rule that could possibly derive $s >_{\mathsf{hc}} s$ is C4. Hence, $\bar{s} \gg^{\mathsf{f}}_{\mathsf{hc}} \bar{s}$ for some $\mathsf{f} \in \mathit{ghd}(\zeta)$. On the other hand, by the induction hypothesis $>_{\mathsf{hc}}$ is irreflexive on the arguments $\bar{s}$ of f. Since $\gg^{\mathsf{f}}$ preserves irreflexivity (property X3), we must have $\bar{s} \not\gg^{\mathsf{f}}_{\mathsf{hc}} \bar{s}$, a contradiction. $\square$

The proof of transitivity relies on several basic lemmas about $>_{\mathsf{hc}}$ and $\mathcal{W}$.

Lemma 20. *If $t >_{\mathsf{hc}} s$, then $\mathcal{W}(t) \geq \mathcal{W}(s)$.*

Proof. Immediate from the definition of $>_{\mathsf{hc}}$. □

Lemma 21. *$\mathcal{W}(s) \geq \varepsilon$.*

Proof. By strong induction on $|s|$. Let $s = \zeta\ \bar{s}$. If $\mathit{mghd}(\zeta) = \iota$, then $\delta = \varepsilon$ and $\bar{s}$ is either $()$ or a single term s'. In the first case, $\mathcal{W}(s) = \delta = \varepsilon$; in the second case, $\mathcal{W}(s) = \mathcal{W}(s')$, which is at least ε by the induction hypothesis. Finally, if $\mathit{mghd}(\zeta) \neq \iota$, we have $w(\mathit{mghd}(\zeta)) \geq \varepsilon$, and each argument in $\bar{s}$ additionally contributes at least $\varepsilon - \delta \geq 0$ to the weight of s. □

Lemma 22. *The following properties hold for $i \in \{1,2\}$:*

(1) $\mathcal{W}(s_1\ s_2) \geq \mathcal{W}(s_i)$; (2) $\mathcal{W}(s_1\ s_2)|_A = \mathcal{W}(s_i)|_A \Longrightarrow \delta = \varepsilon$.

Proof. The properties follow from the definition of $\mathcal{W}$, the requirement that argument coefficients are nonzero (i.e., ≥ 1), and Lemma 21. □

Lemma 23. *If $\delta = \varepsilon$, then $() \not\gg^{\mathsf{f}}_{\mathsf{hc}} s$.*

Proof. Assume $() \gg^{\mathsf{f}}_{\mathsf{hc}} s$. By minimality of the empty tuple (property X10), we also have $s \gg^{\mathsf{f}}_{\mathsf{hc}} ()$. By preservation of transitivity of $\gg^{\mathsf{f}}_{\mathsf{hc}}$ (property X4) together with irreflexivity of $>_{\mathsf{hc}}$ (Theorem 19) and transitivity of $>_{\mathsf{hc}}$ on the set $\{s\}$ (a triviality), we get $() \gg^{\mathsf{f}}_{\mathsf{hc}} ()$. Yet, by preservation of irreflexivity (property X3) together with irreflexivity of $>_{\mathsf{hc}}$, we have $() \not\gg^{\mathsf{f}}_{\mathsf{hc}} ()$, a contradiction. □

Lemma 24. *$s\ t >_{\mathsf{hc}} t$.*

Proof. By strong induction on $|t|$. First, we have $\mathcal{W}(s\ t) \geq \mathcal{W}(t)$ by Lemma 22(1), as required to apply rule C2 or C3. If $\mathcal{W}(s\ t) > \mathcal{W}(t)$, we derive $s\ t >_{\mathsf{hc}} t$ by rule C1. Otherwise, there must exist an assignment A such that $\mathcal{W}(s\ t)|_A = \mathcal{W}(t)|_A$. By Lemmas 21 and 22(2), this can happen only if $\mathcal{W}(s)|_A = \delta = \varepsilon$, which in turns means that $\iota \in \mathit{ghd}(\mathit{hd}(s))$. Since ι is the maximal symbol for $\succ$, either $\mathit{hd}(s) = \mathit{hd}(t)$, $\mathit{hd}(s) \succ \mathit{hd}(t)$, or the two heads are incomparable. The last two possibilities are easily handled by appealing to rule C2 or C3. If $\mathit{hd}(s) = \mathit{hd}(t) = \zeta$, then t must be of the form ζ or $\zeta\ t'$, with $\iota \in \mathit{ghd}(\zeta)$. In the $t = \zeta$ case, we have $\zeta \gg^{\mathsf{f}}_{\mathsf{hc}} ()$ for all $\mathsf{f} \in \Sigma$ by minimality of the empty tuple (property X10). In the $t = \zeta\ t'$ case, we have $t >_{\mathsf{hc}} t'$ by the induction hypothesis and hence $t \gg^{\mathsf{f}}_{\mathsf{hc}} t'$ for any $\mathsf{f} \in \Sigma$ by compatibility with tuple contexts (property X6) together with irreflexibility (Theorem 19). In both cases, $\zeta\ t >_{\mathsf{hc}} \zeta\ t'$ by rule C4. □

Theorem 25 (Transitivity). *If $u >_{\mathsf{hc}} t$ and $t >_{\mathsf{hc}} s$, then $u >_{\mathsf{hc}} s$.*

Proof. By well-founded induction on the multiset $\{|s|,|t|,|u|\}$ with respect to the multiset extension of $>$ on $\mathbb{N}$. Let $u = \chi\ \bar{u}$, $t = \xi\ \bar{t}$, and $s = \zeta\ \bar{s}$. By Lemma 20, we have $\mathcal{W}(u) \geq \mathcal{W}(t) \geq \mathcal{W}(s)$.

If either $u >_{\mathsf{hc}} t$ or $t >_{\mathsf{hc}} s$ was derived by rule C1, we get $u >_{\mathsf{hc}} s$ by rule C1.

If $u >_{\mathsf{hc}} t$ was derived by rule C2, u must be of the form $\chi\ u'$, with $\iota \in \mathit{ghd}(\chi)$ and $u' \geq_{\mathsf{hc}} t$. Since $t >_{\mathsf{hc}} s$ by hypothesis, $u' >_{\mathsf{hc}} s$ follows either immediately (if $u' = t$) or by the induction hypothesis (if $u' >_{\mathsf{hc}} t$). We also have $u >_{\mathsf{hc}} u'$ by Lemma 24. Then:

- If $t >_{\mathsf{hc}} s$ was derived by rule C2, t must be of the form $\xi\ t'$ with $t' \geq_{\mathsf{hc}} s$. We also have $t >_{\mathsf{hc}} t'$ by Lemma 24. Recall that $u' \geq_{\mathsf{hc}} t$. We then proceed by distinguishing three subcases, based on the respective sizes of the terms we want to compare:
  - If $|u'| < |t|$, we invoke the induction hypothesis on $u >_{\mathsf{hc}} u'$ and $u' >_{\mathsf{hc}} s$ to get the desired result, $u >_{\mathsf{hc}} s$.
  - If $|t'| < |s|$, we invoke the induction hypothesis first on $u >_{\mathsf{hc}} t$ and $t >_{\mathsf{hc}} t'$ to derive $u >_{\mathsf{hc}} t'$ and then on $u >_{\mathsf{hc}} t'$ and $t' >_{\mathsf{hc}} s$ to get $u >_{\mathsf{hc}} s$.
  - Otherwise, we have $|u| > |u'| \geq |t| > |t'| \geq |s|$. We further distinguish four sub-subcases. If $\chi \succ \zeta$, we apply rule C3. If $\zeta \succ \chi$, we get a contradiction with $\iota \in \mathit{ghd}(\chi)$. If ζ and χ are incomparable, we apply rule C2. The (sub-sub)case where $\chi = \zeta$ remains. Because of arity constraints on C2, s is either ζ or of the form $\zeta\ s'$. If $s = \zeta$, we have $\zeta\ u' >_{\mathsf{hc}} \zeta$ by minimality of the empty tuple (property X10). Otherwise, $s = \zeta\ s'$. We apply the induction hypothesis twice to derive $u' >_{\mathsf{hc}} t'$ (via t) and $t' >_{\mathsf{hc}} s'$ (via s, using Lemma 24 to derive $s >_{\mathsf{hc}} s'$). A third application yields $u' >_{\mathsf{hc}} s'$ and hence $u' \gg^{\mathsf{f}}_{\mathsf{hc}} s'$ for all $\mathsf{f} \in \Sigma$ by compatibility with tuple contexts (property X6) together with irreflexibility (Theorem 19). Finally, we get $u = \zeta\ u' >_{\mathsf{hc}} \zeta\ s' = s$ by rule C4.
- If $t >_{\mathsf{hc}} s$ was derived by rule C3, we have $\xi \succ \zeta$. Since ξ and χ are incomparable and $\succ$ is transitive, it cannot be that $\zeta \succeq \chi$. The remaining options are that $\chi \succ \zeta$ and that ζ and χ are incomparable. Depending on the case, we apply rule C3 or C2.
- If $t >_{\mathsf{hc}} s$ was derived by rule C4, we have that $\zeta = \xi$ is incomparable with χ. We get $u >_{\mathsf{hc}} s$ by rule C2 using $u' >_{\mathsf{hc}} s$.

If $u >_{\mathsf{hc}} t$ was derived by rule C3, we have $\chi \succ \xi$. Then:

- If $t >_{\mathsf{hc}} s$ was derived by rule C2, we have $\iota \in \mathit{ghd}(\xi)$. But since $\chi \succ \xi$, necessarily $\iota \notin \mathit{ghd}(\chi)$, contradicting one of the conditions on rule C2.
- If $t >_{\mathsf{hc}} s$ was derived by rule C3 or C4, we derive $u >_{\mathsf{hc}} s$ by rule C3, possibly exploiting the transitivity of $\succ$.

If $u >_{\mathsf{hc}} t$ was derived by rule C4, we have $\chi = \xi$ and $\bar{u} \gg^{\mathsf{f}}_{\mathsf{hc}} \bar{t}$. Then:

- If $t >_{\mathsf{hc}} s$ was derived by rule C2, t is of the form $\xi\ t'$ with $\mathit{arity}(\xi) = 1$ and $t' \geq_{\mathsf{hc}} s$. Hence, u is either ξ or of the form $\xi\ u'$. In the first case, we have $\xi >_{\mathsf{hc}} \xi\ t'$, but this is impossible by Lemma 23. In the second case, we have $u' >_{\mathsf{hc}} t'$ (i.e., $\bar{u} \gg^{\mathsf{f}}_{\mathsf{hc}} \bar{t}$). Since $t' \geq_{\mathsf{hc}} s$, we get $u' >_{\mathsf{hc}} s$ either immediately (if $t' = s$) or by the induction hypothesis (if $t' >_{\mathsf{hc}} s$). Finally, we apply C2 to derive $u >_{\mathsf{hc}} s$.
- If $t >_{\mathsf{hc}} s$ was derived by rule C3, we get $u >_{\mathsf{hc}} s$ by rule C3.
- If $t >_{\mathsf{hc}} s$ was derived by rule C4, we apply C4 to derive $u >_{\mathsf{hc}} s$. This relies on the preservation by $\gg^{\mathsf{f}}_{\mathsf{hc}}$ of transitivity (property X4) on the set consisting of the argument tuples of s, t, u. Transitivity of $>_{\mathsf{hc}}$ on these tuples follows from the induction hypothesis. □

By Theorems 19 and 25, $>_{\mathsf{hc}}$ is a partial order. In the remaining proofs, we will often leave applications of these theorems (and of antisymmetry) implicit.

Lemma 26. *$s\ t >_{\mathsf{hc}} s$.*

Proof. If $\mathcal{W}(s\,t) > \mathcal{W}(s)$, the desired result can be derived using C1. Otherwise, we have $\mathcal{W}(s\,t) \geq \mathcal{W}(s)$ and $\delta = \varepsilon$ by Lemma 22. The desired result follows from rule C4, compatibility with prepending (property X8), and minimality of the empty tuple (property X10). □

Theorem 27 (Subterm Property). *If s is a proper subterm of t, then $t >_{\mathsf{hc}} s$.*

Proof. By structural induction on t, exploiting Lemmas 24 and 26 and transitivity. □

The first-order KBO satisfies compatibility with Σ-operations. A slightly more general property holds for $>_{\mathsf{hc}}$:

Theorem 28 (Compatibility with Functions). *If $t' >_{\mathsf{hc}} t$, then $s\,t'\,\bar{u} >_{\mathsf{hc}} s\,t\,\bar{u}$.*

Proof. By induction on the length of $\bar{u}$. The base case, $\bar{u} = ()$, follows from rule C4, Lemma 20, compatibility of $\gg^{\mathsf{f}}$ with tuple contexts (property X6), and irreflexivity of $>_{\mathsf{hc}}$. In the step case, $\bar{u} = \bar{u}' \cdot u$, we have $\mathcal{W}(s\,t'\,\bar{u}') \geq \mathcal{W}(s\,t\,\bar{u}')$ from the induction hypothesis together with Lemma 20. Hence $\mathcal{W}(s\,t'\,\bar{u}) \geq \mathcal{W}(s\,t\,\bar{u})$ by the definition of $\mathcal{W}$. Thus, we can apply rule C4, again exploiting compatibility of $\gg^{\mathsf{f}}$ with contexts. □

To build arbitrary higher-order contexts, we also need compatibility with arguments. This property can be used to rewrite subterms such as $\mathsf{f\,a}$ in $\mathsf{f\,a\,b}$ using a rewrite rule $\mathsf{f}\,x \to t_x$. The property holds unconditionally for $>_{\mathsf{hb}}$ and $>_{\mathsf{hz}}$ but not for $>_{\mathsf{hc}}$: $s' >_{\mathsf{hc}} s$ does not imply $s'\,t >_{\mathsf{hc}} s\,t$, because the occurrence of t may weigh more as an argument to s than to s'. By restricting the coefficients of s and s', we get a weaker property:

Theorem 29 (Compatibility with Arguments). *Assume that $\gg^{\mathsf{f}}$ is compatible with appending (property X9) for every symbol $\mathsf{f} \in \Sigma$. If $s' >_{\mathsf{hc}} s$ and $\mathit{coef}(s', 1) \geq \mathit{coef}(s, 1)$, then $s'\,t >_{\mathsf{hc}} s\,t$.*

Proof. If $s' >_{\mathsf{hc}} s$ was derived by rule C1, by exploiting $\mathit{coef}(s', 1) \geq \mathit{coef}(s, 1)$ and the definition of $\mathcal{W}$, we can apply rule C1 to get the desired result. Otherwise, we have $\mathcal{W}(s') \geq \mathcal{W}(s)$ by Lemma 20 and hence $\mathcal{W}(s'\,t) \geq \mathcal{W}(s\,t)$, a prerequisite for applying rules C2–C4. Due to the implicit assumption that $\mathit{coef}(s', 1)$ is defined, and hence that s' is not fully applied, $s' >_{\mathsf{hc}} s$ cannot have been derived by rule C2. If $s' >_{\mathsf{hc}} s$ was derived by rule C3, we get the desired result by rule C3. If $s' >_{\mathsf{hc}} s$ was derived by rule C4, we get the result by rule C4 together with property X9. □

The next theorem, stability under substitution, depends on a substitution lemma connecting term substitutions and polynomial unknown assignments.

Definition 30. The *composition* $A \circ \sigma$ of a substitution σ and an assignment A is defined by

$$(A \circ \sigma)(\mathbf{w}_x) = \mathcal{W}(x\sigma)|_A - \delta \cdot \mathit{arity}(\mathit{mghd}(x)) \qquad (A \circ \sigma)(\mathbf{k}_{x,i}) = \mathit{coef}(x\sigma, i)\big|_A$$

Lemma 31 (Substitution). *Let σ be a substitution that respects the mapping ghd. Then $\mathcal{W}(s\sigma)|_A = \mathcal{W}(s)|_{A \circ \sigma}$.*

Proof. By strong induction on $|s|$. Let $s = \zeta\ s_1 \ldots s_m$. If ζ is a symbol f, we have

$$\begin{aligned} \mathcal{W}(s\sigma)\big|_A &= w(\mathsf{f}) + \delta \cdot (\mathit{arity}(\mathsf{f}) - m) + \textstyle\sum_{i=1}^m \mathit{coef}(\mathsf{f}, i) \cdot \mathcal{W}(s_i\sigma)\big|_A \\ &\overset{\text{IH}}{=} w(\mathsf{f}) + \delta \cdot (\mathit{arity}(\mathsf{f}) - m) + \textstyle\sum_{i=1}^m \mathit{coef}(\mathsf{f}, i) \cdot \mathcal{W}(s_i)\big|_{A\circ\sigma} \\ &= \mathcal{W}(s)\big|_{A\circ\sigma} \end{aligned}$$

where 'IH' indicates an application of the induction hypothesis. Otherwise, ζ is a variable x. Let $x\sigma = \xi\ t_1 \ldots t_n$. Note that $s\sigma = \xi\ t_1 \ldots t_n\ (s_1\sigma) \ldots (s_m\sigma)$ and $\mathit{coef}(x, i)\big|_A = \mathit{coef}(\xi, n+i)\big|_{A\circ\sigma}$ for all indices i. Then:

$$\begin{aligned} \mathcal{W}(s\sigma)\big|_A &= w(\xi)\big|_A + \delta \cdot (\mathit{arity}(\mathit{mghd}(\xi)) - n - m) + \textstyle\sum_{j=1}^n \mathit{coef}(\xi, j) \cdot \mathcal{W}(t_j)\big|_A \\ &\quad + \textstyle\sum_{i=1}^m \mathit{coef}(\xi, n+i) \cdot \mathcal{W}(s_i\sigma)\big|_A \\ &\overset{\text{IH}}{=} w(\xi)\big|_A + \delta \cdot (\mathit{arity}(\mathit{mghd}(\xi)) - n - m) + \textstyle\sum_{j=1}^n \mathit{coef}(\xi, j) \cdot \mathcal{W}(t_j)\big|_A \\ &\quad + \textstyle\sum_{i=1}^m \mathit{coef}(x, i) \cdot \mathcal{W}(s_i)\big|_{A\circ\sigma} \\ &= w(x)\big|_{A\circ\sigma} + \delta \cdot (\mathit{arity}(\mathit{mghd}(\xi)) - m) + \textstyle\sum_{i=1}^m \mathit{coef}(x, i) \cdot \mathcal{W}(s_i)\big|_{A\circ\sigma} \\ &= \mathcal{W}(s)\big|_{A\circ\sigma} \end{aligned}$$

□

Theorem 32 (Stability under Substitution). *If $t >_{\mathsf{hc}} s$, then $t\sigma >_{\mathsf{hc}} s\sigma$ for any substitution σ that respects the mapping ghd.*

Proof. By well-founded induction on the multiset $\{|s|, |t|\}$ with respect to the multiset extension of $>$ on $\mathbb{N}$.

If $t >_{\mathsf{hc}} s$ was derived by rule C1, $\mathcal{W}(t) > \mathcal{W}(s)$. Hence, $\mathcal{W}(t)\big|_{A\circ\sigma} > \mathcal{W}(s)\big|_{A\circ\sigma}$, and by the substitution lemma (Lemma 31), we get $\mathcal{W}(t\sigma) > \mathcal{W}(s\sigma)$. The desired result, $t\sigma >_{\mathsf{hc}} s\sigma$, follows by rule C1.

If $t >_{\mathsf{hc}} s$ was derived by rule C2, then t must be of the form $\xi\ t'$ with $t' \geq s$. If $t' = s$, we get $t\sigma >_{\mathsf{hc}} s\sigma$ by the subterm property (Theorem 27). Otherwise, we have $t' >_{\mathsf{hc}} s$ and hence $t'\sigma >_{\mathsf{hc}} s\sigma$ by the induction hypothesis. Moreover, $t\sigma >_{\mathsf{hc}} t'\sigma$ by the subterm property. By transitivity, $t\sigma >_{\mathsf{hc}} s\sigma$.

If $t >_{\mathsf{hc}} s$ was derived by rule C3, we have $\mathcal{W}(t) \geq \mathcal{W}(s)$ and $\mathit{hd}(t) \succ \mathit{hd}(s)$. We get $\mathcal{W}(t\sigma) \geq \mathcal{W}(s\sigma)$ by the substitution lemma and $\mathit{hd}(t\sigma) \succ \mathit{hd}(s\sigma)$ by the definition of $\succ$. The desired result follows by rule C3.

If $t >_{\mathsf{hc}} s$ was derived by rule C4, we have $\mathcal{W}(t) \geq \mathcal{W}(s)$, $\mathit{hd}(t) = \mathit{hd}(s) = \zeta$, and $\mathit{args}(t) \gg^{\mathsf{f}}_{\mathsf{hc}} \mathit{args}(s)$ for all $\mathsf{f} \in \mathit{ghd}(\zeta)$. Since σ respects ghd, we have the inclusion $\mathit{ghd}(\mathit{hd}(s\sigma)) \subseteq \mathit{ghd}(\zeta)$. We apply preservation of stability of $\gg^{\mathsf{f}}_{\mathsf{hc}}$ (property X2) to derive $\mathit{args}(t)\sigma \gg^{\mathsf{f}}_{\mathsf{hc}} \mathit{args}(s)\sigma$ for all $\mathsf{f} \in \mathit{ghd}(\mathit{hd}(s\sigma)) \subseteq \mathit{ghd}(\zeta)$. This step requires that $t' > s'$ implies $t'\sigma > s'\sigma$ for all $s', t' \in \mathit{args}(s) \cup \mathit{args}(t)$, which follows from the induction hypothesis. From $\mathit{args}(t)\sigma \gg^{\mathsf{f}}_{\mathsf{hc}} \mathit{args}(s)\sigma$, we get

$$\mathit{args}(t\sigma) = \mathit{args}(\zeta)\sigma \cdot \mathit{args}(t)\sigma \gg^{\mathsf{f}}_{\mathsf{hc}} \mathit{args}(\zeta)\sigma \cdot \mathit{args}(s)\sigma = \mathit{args}(s\sigma)$$

by compatibility with prepending (property X8). Finally, we apply rule C4. □

The use of signed ordinals is crucial for Definition 30 and Lemma 31. Consider the signature $\Sigma = \{\mathsf{f}, \mathsf{g}\}$ where $\mathit{arity}(\mathsf{f}) = 3$, $\mathit{arity}(\mathsf{g}) = 0$, $w(\mathsf{f}) = 1$, and $w(\mathsf{g}) = \omega$.

Assume $\delta = \varepsilon = 1$. Let $x \in \mathcal{V}$ be an arbitrary variable such that $\mathit{ghd}(x) = \Sigma$; clearly, $\mathit{mghd}(x) = \mathsf{f}$. Let A be an assignment such that $A(x) = w(\mathit{mghd}(x)) = w(\mathsf{f}) = 1$, and let σ be a substitution that maps x to g. A negative coefficient arises when we compose σ with A: $(A \circ \sigma)(x) = \mathcal{W}(\mathsf{g}) - \delta \cdot \mathit{arity}(\mathsf{f}) = w(\mathsf{g}) + \delta \cdot \mathit{arity}(\mathsf{g}) - \delta \cdot \mathit{arity}(\mathsf{f}) = \omega - 3$. However, if we fix $\delta = 0$, we can use plain ordinals throughout.

Theorem 33 (Ground Totality). *Assume $\gg^{\mathsf{f}}$ preserves totality (property X7) for every symbol $\mathsf{f} \in \Sigma$, and let s, t be ground terms. Then either $t \geq_{\mathsf{hc}} s$ or $t <_{\mathsf{hc}} s$.*

Proof. By strong induction on $|s| + |t|$. Let $t = \mathsf{g}\,\bar{t}$ and $s = \mathsf{f}\,\bar{s}$. If $\mathcal{W}(s) \neq \mathcal{W}(t)$, then either $\mathcal{W}(t) > \mathcal{W}(s)$ or $\mathcal{W}(t) < \mathcal{W}(s)$, since the weights of ground terms contain no polynomial unknowns. Hence, we have $t >_{\mathsf{hc}} s$ or $t <_{\mathsf{hc}} s$ by rule C1. Otherwise, $\mathcal{W}(s) = \mathcal{W}(t)$. If $\mathsf{f} \neq \mathsf{g}$, then either $\mathsf{g} \succ \mathsf{f}$ or $\mathsf{g} \prec \mathsf{f}$, and we have $t >_{\mathsf{hc}} s$ or $t <_{\mathsf{hc}} s$ by rule C3. Otherwise, $\mathsf{g} = \mathsf{f}$. By preservation of totality (property X7), we have either $\bar{t} \gg^{\mathsf{f}}_{\mathsf{hc}} \bar{s}$, $\bar{t} \ll^{\mathsf{f}}_{\mathsf{hc}} \bar{s}$, or $\bar{s} = \bar{t}$. In the first two cases, we have $t >_{\mathsf{hc}} s$ or $t <_{\mathsf{hc}} s$ by rule C4. In the third case, we have $s = t$. □

Lemma 34. *Let $\mathsf{f}\,\bar{s}$ be a ground term. Then $|\bar{s}| \leq \mathit{sumcoefs}(\mathcal{W}(\mathsf{f}\,\bar{s}))$, where sumcoefs is defined by $\mathit{sumcoefs}(\sum_{i=1}^{m} \omega^{\alpha_i} k_i) = \sum_{i=1}^{m} k_i$ for all $m \in \mathbb{N}$, $\alpha_i \in \mathbf{O}$, and $k_i \in \mathbb{N}_{>0}$.*

Proof. First, we observe that $\mathcal{W}(s)|_A$ is a plain ordinal for any term s, as a consequence of the definition of $\mathcal{W}$ and Lemma 21. Since each argument from $\bar{s}$ contributes at least $\varepsilon \geq 1$ to the weight of $\mathcal{W}(\zeta\,\bar{s})$, they must contribute at least 1 to one of the coefficients of the plain ordinal $\mathcal{W}(\zeta\,\bar{s})|_A$. □

Theorem 35 (Well-foundedness). *There exists no infinite descending chain $s_0 >_{\mathsf{hc}} s_1 >_{\mathsf{hc}} \cdots$.*

Proof. We assume that there exists a chain $s_0 >_{\mathsf{hc}} s_1 >_{\mathsf{hc}} \cdots$ and show that this leads to a contradiction. If the chain contains nonground terms, we can instantiate all variables by arbitrary terms respecting ghd and exploit stability under substitution (Theorem 32). Thus, we may assume without loss of generality that the terms $s_0, s_1, \ldots$ are ground.

We call a ground term *bad* if it belongs to an infinite descending $>_{\mathsf{hc}}$-chain. Without loss of generality, we may assume that s_0 has minimal size among all bad terms and that s_{i+1} has minimal size among all bad terms t such that $s_i >_{\mathsf{hc}} t$.

For each index i, the term s_i must be of the form $\mathsf{f}\,u_1 \ldots u_n$ for some symbol f and ground terms $u_1, \ldots, u_n$. Let $U_i = \{u_1, \ldots, u_n\}$. Now let $U = \bigcup_{i=0}^{\infty} U_i$. All terms belonging to U are good: If a term from U_0 were bad, this would contradict the minimality of s_0; and if a term $u \in U_{i+1}$ were bad, then we would have $s_{i+1} >_{\mathsf{hc}} u$ by the subterm property (Theorem 27) and $s_i >_{\mathsf{hc}} u$ by transitivity, contradicting the minimality of s_{i+1}.

Next, we analyze which rules can be used to justify each link $s_i >_{\mathsf{hc}} s_{i+1}$ in the chain. Since all terms s_i are ground and $\succ$ is total on symbols, rule C2 is inapplicable. This leaves C1, C3, and C4. Since the weight of a ground term is clearly a plain ordinal, each transition either keeps the weight unchanged or strictly decreases it. Since $>$ is well founded on ordinals, the rule C1 is applicable only a finite number of times in the chain. Hence, there must exist an index k such that $s_i >_{\mathsf{hc}} s_{i+1}$ is derived using C3 or C4 for all $i \geq k$, and all these terms share the same weight w. Moreover, because $\succ$ is well founded and C4 preserves the head symbol, rule C3 can be applied only a finite number of times in the chain. Hence, there must exist an index $l \geq k$ such that $s_i >_{\mathsf{hc}} s_{i+1}$ is derived using C4 for all $i \geq l$. Consequently, all terms s_i for $i \geq l$ share the same head symbol f.

The last step of the proof requires us to bound the number of arguments to f. The obvious candidate, $\mathit{arity}(\mathsf{f})$, is not an option because it can be ∞. Instead, we appeal to Lemma 34, which gives us $\mathit{sumcoefs}(w)$ as a bound on $|\bar{u}_i|$ for $i \geq l$. Since rule C4 is used consistently from index l, we have an infinite $\gg^{\mathsf{f}}_{\mathsf{hc}}$-chain: $\bar{u}_l \gg^{\mathsf{f}}_{\mathsf{hc}} \bar{u}_{l+1} \gg^{\mathsf{f}}_{\mathsf{hc}} \bar{u}_{l+2} \gg^{\mathsf{f}}_{\mathsf{hc}} \cdots$. But since U contains only good terms and comprises all terms occurring in some argument tuple $\bar{u}_i$, $>_{\mathsf{hc}}$ is well founded on U. By bounded preservation of well-foundedness (Lemma 3), $\gg^{\mathsf{f}}_{\mathsf{hc}}$ is well founded. This contradicts the existence of the above $\gg^{\mathsf{f}}_{\mathsf{hc}}$-chain. ☐

Theorem 36 (Coincidence with First-Order KBO). *Let $>_{\mathsf{hz}}$ and $>_{\mathsf{fo}}$ be orders induced by the same precedence $\succ$ and extension operator family $\gg^{\mathsf{f}}$ satisfying properties X8–X10 and $\delta = \varepsilon$. Then $>_{\mathsf{hz}}$ and $>_{\mathsf{fo}}$ coincide on first-order terms.*

Proof. This is obvious from Definitions 8 and 14. ☐

Moreover, instances of $>_{\mathsf{hc}}$ coincides with the first-order KBO with argument coefficients on first-order terms as described in the literature.

6 Formalization

The definitions and the proofs presented in this paper have been fully formalized in Isabelle/HOL [39] and are part of the *Archive of Formal Proofs* [6]. The formal development relies on no custom axioms; at most local assumptions such as "$\succ$ is a well-founded total order on Σ" are made. The development focuses on two KBO variants: the transfinite $>_{\mathsf{hc}}$ with argument coefficients and the restriction of $>_{\mathsf{hz}}$ to natural number weights. The use of Isabelle, including its model finder Nitpick [13] and a portfolio of automatic theorem provers [12], was invaluable for designing the orders, proving their properties, and carrying out various experiments.

The basic infrastructure for λ-free higher-order terms and extension orders is shared with our formalization of the λ-free higher-order RPO [14]. Beyond standard Isabelle libraries, the formal proof development also required polynomials and ordinals. For the polynomials, we used Sternagel and Thiemann's *Archive of Formal Proofs* entry [41]. For the ordinals, we developed our own library, with help from Mathias Fleury and Dmitriy Traytel [10]. Syntactic ordinals are isomorphic to the hereditarily finite multisets, which can be defined easily using Isabelle's (co)datatype definitional package [11]:

datatype *hmultiset* $=$ HMSet $(\mathit{hmultiset\ multiset})$

The above command introduces a type *hmultiset* generated freely by the constructor $\mathsf{HMSet} : \mathit{hmultiset\ multiset} \to \mathit{hmultiset}$, where *multiset* is Isabelle's unary (postfix) type constructor of finite multisets. A syntactic ordinal $\sum_{i=1}^{m} \omega^{\alpha_i} k_i$ is represented by the multiset consisting of k_1 copies of α_1, k_2 copies of α_2, …, k_m copies of α_m. Accordingly:

$$0 = \mathsf{HMSet}\ \{\} \quad 1 = \mathsf{HMSet}\ \{0\} \quad 5 = \mathsf{HMSet}\ \{0,0,0,0,0\} \quad 2\omega = \mathsf{HMSet}\ \{1,1\}$$

Signed syntactic ordinals are defined as finite signed multisets of *hmultiset* values. Signed (or hybrid) multisets generalize standard multisets by allowing negative multiplicities [5].

The main discrepancy between this report and the formalization concerns the basic formalism: set theory versus higher-order logic (simple type theory) with polymorphism and type classes. The types of simple type theory can often be used to model the sets of set theory—for example, the polymorphic type α *list* is often an appropriate approximation of A^*. But this approach is too coarse for the delicate bootstrapping necessary for properties such as irreflexivity and transitivity, which require (simply typed) sets, leading to somewhat more convoluted statements. As an example among many, preservation of transitivity (property X4) must be relativized to sets A in Isabelle's simple type theory:

assume
$\mathsf{finite}\,A \Longrightarrow zs \in \mathsf{lists}\,A \Longrightarrow ys \in \mathsf{lists}\,A \Longrightarrow xs \in \mathsf{lists}\,A \Longrightarrow$
$(\forall x \in A.\ x \not> x) \Longrightarrow (\forall z \in A.\ \forall y \in A.\ \forall x \in A.\ z > y \longrightarrow y > x \longrightarrow z > x) \Longrightarrow$
$zs \gg ys \Longrightarrow ys \gg xs \Longrightarrow zs \gg xs$

7 Examples

Notwithstanding our focus on superposition, we can use $>_{\mathsf{hc}}$ or its special cases $>_{\mathsf{hb}}$ and $>_{\mathsf{hz}}$ to show the termination of λ-free higher-order term rewriting systems or, equivalently, applicative term rewriting systems [28]. To establish termination of a term rewriting system, it suffices to show that all of its rewrite rules $t \to s$ can be oriented as $t > s$ by a single *reduction order*: a well-founded partial order that is compatible with contexts and stable under substitutions. If the order additionally enjoys the subterm property, it is called a *simplification order*. Under the proviso that ghd honestly captures the set of ground heads that may arise when instantiating the variables, the order $>_{\mathsf{hz}}$ is a simplification order. By contrast, $>_{\mathsf{hc}}$ is not even a reduction order since it lacks compatibility with arguments. Nonetheless, the conditional Theorem 29 is sufficient if the outermost heads are fully applied or if their pending argument coefficients are known and suitable [15, Sect. 5].

In the examples below, unless specified otherwise, $\delta = 0$, $\varepsilon = 1$, $w(\mathsf{f}) = 1$, and $\gg^{\mathsf{f}}$ is the length-lexicographic order, for all symbols f.

Example 37. Consider the following system [15, Example 23], where f is a variable:

$$\mathsf{insert}\ (f\ n)\ (\mathsf{image}\ f\ A) \xrightarrow{1} \mathsf{image}\ f\ (\mathsf{insert}\ n\ A) \qquad \mathsf{square}\ n \xrightarrow{2} \mathsf{times}\ n\ n$$

Rule 1 captures a set-theoretic property: $\{f(n)\} \cup f[A] = f[\{n\} \cup A]$, where $f[A]$ denotes the image of set A under function f. We can prove this system terminating using $>_{\mathsf{hc}}$: By letting $w(\mathsf{square}) = 2$ and $\mathit{coef}(\mathsf{square}, 1) = 2$, both rules can be oriented by C1. Rule 2 is beyond the reach of the orders $>_{\mathsf{ap}}$, $>_{\mathsf{hb}}$, and $>_{\mathsf{hz}}$, because there are too many occurrences of n on the right-hand side. The system is also beyond the scope of the uncurrying approach of Hirokawa et al. [23], because of the applied variable f on the left-hand side of rule 1.

Example 38. The following system specifies map functions on ML-style option and list types, each equipped with two constructors:

$$\mathsf{omap}\ f\ \mathsf{None} \xrightarrow{1} \mathsf{None} \qquad \mathsf{omap}\ f\ (\mathsf{Some}\ n) \xrightarrow{2} \mathsf{Some}\ (f\ n)$$
$$\mathsf{map}\ f\ \mathsf{Nil} \xrightarrow{3} \mathsf{Nil} \qquad \mathsf{map}\ f\ (\mathsf{Cons}\ m\ ms) \xrightarrow{4} \mathsf{Cons}\ (f\ m)\ (\mathsf{map}\ f\ ms)$$

Rules 1–3 are easy to orient using C1, but rule 4 is beyond the reach of all KBO variants. To compensate for the two occurrences of the variable f on the right-hand side, we would need a coefficient of at least 2 on map's first argument, but the coefficient would also make the recursive call map f heavier on the right-hand side.

The limitation affecting the map function in Example 38 prevents us from using KBO to prove termination of most of the term rewriting systems we used to demonstrate our RPO [15]. Moreover, the above examples are easy for modern first-order termination provers, which use uncurrying techniques [23, 42] to transform applicative rewrite systems into functional systems that can be analyzed by standard techniques. This is somewhat to be expected: Even with transfinite weights and argument coefficients, KBO tends to consider syntactically smaller terms smaller. However, for superposition, this limitation might be a strength. The calculus's inferences and simplifications rely on the term order to produce smaller and smaller terms (and literals and clauses). Using KBO, the terms will typically be syntactically smaller as well. This is desirable, because many algorithms and data structures do not scale well in term size.

Moreover, for superposition, the goal is not to orient a given set of equations in a particular way, but rather to obtain either $t > s$ or $t < s$ for a high percentage of terms s, t arising during proof search, quickly. The first-order KBO can be implemented so that it takes linear time to compute in the size of the terms [35]. The same techniques are easy to generalize to our KBO variants, if we use the approach discussed at the end of Sect. 4.3 to compare the arguments of variable heads.

8 Conclusion

Compared with RPO, generalizing KBO to λ-free higher-order terms was fairly straightforward: Since weights are the primary criterion of comparison, the subterm property also holds in a higher-order setting, where f a is a subterm of f a b. On the other hand, allowing symbols with weight 0 raised issues that are specific to KBO. The arithmetic needed to make this work led to the introduction of signed ordinals, which are interesting in their own right.

When designing the KBO variants $>_{\mathsf{hb}}$, $>_{\mathsf{hz}}$, and $>_{\mathsf{hc}}$ and the RPO variants that preceded them [15], we aimed at full coincidence with the first-order case. Our goal is to gradually transform existing first-order automatic provers into higher-order provers. By carefully generalizing the proof calculi and data structures, we aim at designing provers that behave exactly like first-order provers on first-order problems, perform mostly like first-order provers on higher-order problems that are mostly first-order, and scale up to arbitrary higher-order problems.

An open question is, *What is the best way to cope with λ-abstraction in a superposition prover?* The Leo-III prover [48] relies on the computability path order [16] to reduce the search space; however, the order lacks many of the properties needed for completeness. With its stratified architecture, Otter-λ [7] is closer to what we are aiming at, but it is limited to second-order logic and offers no completeness guarantees.

A simple approach to λ-abstractions is to encode them using SK combinators [46]. This puts a heavy burden on the superposition machinery (and is a reason why HOLy-Hammer and Sledgehammer are so weak on higher-order problems). We could alleviate some of this burden by making the prover aware of the combinators, implementing higher-order unification and other algorithms specialized for higher-order reasoning in terms of them. A more appealing approach may be to employ a lazy variant of λ-lifting [26], whereby fresh symbols f and definitions $\mathsf{f}\,\bar{x} \approx t$ are introduced during proof search. Argument coefficients could be used to orient the definition as desired. For example, $\lambda x.\ x+x+x$ could be mapped to a symbol g with an argument coefficient of 3 and a sufficiently large weight to ensure that $\mathsf{g}\ x \approx x+x+x$ is oriented from left to right. However, it is not even clear that a left-to-right orientation is suitable here. Since superposition provers generally work better on syntactically small terms, it might be preferable to fold the definition of g whenever possible rather than unfold it.

Acknowledgment. We are grateful to Stephan Merz, Tobias Nipkow, and Christoph Weidenbach for making this research possible; to Mathias Fleury and Dmitriy Traytel for helping us formalize the syntactic ordinals; to Andrei Popescu and Christian Sternagel for advice with extending a partial well-founded order to a total one in the mechanized proof of Lemma 3; to Andrei Voronkov for the enlightening discussion about KBO at the IJCAR 2016 banquet; and to Alexander Bentkamp, Carsten Fuhs, Mark Summerfield, and the anonymous reviewers for suggesting many textual improvements.

Blanchette has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 713999, Matryoshka). Wand is supported by the Deutsche Forschungsgemeinschaft (DFG) grant Hardening the Hammer (NI 491/14-1).

References

[1] P. B. Andrews and E. L. Cohen. Theorem proving in type theory. In R. Reddy, editor, *International Joint Conference on Artificial Intelligence (IJCAI-77)*, page 566. William Kaufmann, 1977.

[2] T. Aoto and T. Yamada. Termination of simply typed term rewriting by translation and labelling. In R. Nieuwenhuis, editor, *Rewriting Techniques and Applications (RTA 2003)*, volume 2706 of *LNCS*, pages 380–394. Springer, 2003.

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

[4] J. Backes and C. E. Brown. Analytic tableaux for higher-order logic with choice. *J. Autom. Reasoning*, 47(4):451–479, 2011.

[5] J.-P. Banâtre, P. Fradet, and Y. Radenac. Generalised multisets for chemical programming. *Math. Struct. Comput. Sci.*, 16(4):557–580, 2006.

[6] H. Becker, J. C. Blanchette, U. Waldmann, and D. Wand. Formalization of Knuth–Bendix orders for lambda-free higher-order terms. *Archive of Formal Proofs*, 2016. `https://isa-afp.org/entries/Lambda_Free_KBOs.shtml`, Formal proof development.

[7] M. Beeson. Lambda Logic. In D. A. Basin and M. Rusinowitch, editors, *International Joint Conference on Automated Reasoning (IJCAR 2004)*, volume 3097 of *LNCS*, pages 460–474. Springer, 2004.

[8] C. Benzmüller and M. Kohlhase. Extensional higher-order resolution. In C. Kirchner and H. Kirchner, editors, *Conference on Automated Deduction (CADE-15)*, volume 1421 of *LNCS*, pages 56–71. Springer, 1998.

[9] C. Benzmüller and D. Miller. Automation of higher-order logic. In J. H. Siekmann, editor, *Computational Logic*, volume 9 of *Handbook of the History of Logic*, pages 215–254. Elsevier, 2014.

[10] J. C. Blanchette, M. Fleury, and D. Traytel. Formalization of nested multisets, hereditary multisets, and syntactic ordinals. *Archive of Formal Proofs*, 2016. `https://isa-afp.org/entries/Nested_Multisets_Ordinals.shtml`, Formal proof development.

[11] J. C. Blanchette, J. Hölzl, A. Lochbihler, L. Panny, A. Popescu, and D. Traytel. Truly modular (co)datatypes for Isabelle/HOL. In G. Klein and R. Gamboa, editors, *Interactive Theorem Proving (ITP 2014)*, LNCS. Springer, 2014.

[12] J. C. Blanchette, C. Kaliszyk, L. C. Paulson, and J. Urban. Hammering towards QED. *J. Formalized Reasoning*, 9(1):101–148, 2016.

[13] J. C. Blanchette and T. Nipkow. Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In M. Kaufmann and L. C. Paulson, editors, *Interactive Theorem Proving (ITP 2010)*, volume 6172 of *LNCS*, pages 131–146. Springer, 2010.

[14] J. C. Blanchette, U. Waldmann, and D. Wand. Formalization of recursive path orders for lambda-free higher-order terms. *Archive of Formal Proofs*, 2016. `https://isa-afp.org/entries/Lambda_Free_RPOs.shtml`, Formal proof development.

[15] J. C. Blanchette, U. Waldmann, and D. Wand. A lambda-free higher-order recursive path order. In J. Esparza and A. S. Murawski, editors, *Foundations of Software Science and Computation Structures (FoSSaCS 2017)*, volume 10203 of *LNCS*, pages 461–479. Springer, 2017.

[16] F. Blanqui, J.-P. Jouannaud, and A. Rubio. The computability path ordering. *Log. Meth. Comput. Sci.*, 11(4), 2015.

[17] M. Bofill, C. Borralleras, E. Rodríguez-Carbonell, and A. Rubio. The recursive path and polynomial ordering for first-order and higher-order terms. *J. Log. Comput.*, 23(1):263–305, 2013.

[18] M. Bofill and A. Rubio. Paramodulation with non-monotonic orderings and simplification. *J. Autom. Reasoning*, 50(1):51–98, 2013.

[19] N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Commun. ACM*, 22(8):465–476, 1979.

[20] M. C. F. Ferreira and H. Zantema. Well-foundedness of term orderings. In N. Dershowitz and N. Lindenstrauss, editors, *Conditional and Typed Rewriting Systems (CTRS-94)*, volume 968 of *LNCS*, pages 106–123. Springer, 1994.

[21] J. Giesl, R. Thiemann, and P. Schneider-Kamp. Proving and disproving termination of higher-order functions. In B. Gramlich, editor, *Frontiers of Combining Systems (FroCoS 2005)*, volume 3717 of *LNCS*, pages 216–231. Springer, 2005.

[22] L. Henkin. Completeness in the theory of types. *J. Symb. Log.*, 15(2):81–91, 1950.

[23] N. Hirokawa, A. Middeldorp, and H. Zankl. Uncurrying for termination and complexity. *J. Autom. Reasoning*, 50(3):279–315, 2013.

[24] G. Huet and D. C. Oppen. Equations and rewrite rules: A survey. In R. V. Book, editor, *Formal Language Theory: Perspectives and Open Problems*, pages 349–405. Academic Press, 1980.

[25] G. P. Huet. A mechanization of type theory. In N. J. Nilsson, editor, *International Joint Conference on Artificial Intelligence (IJCAI-73)*, pages 139–146. William Kaufmann, 1973.

[26] R. J. M. Hughes. Super-combinators: A new implementation method for applicative languages. In *ACM Symposium on LISP and Functional Programming (LFP '82)*, pages 1–10. ACM Press, 1982.

[27] J.-P. Jouannaud and A. Rubio. Polymorphic higher-order recursive path orderings. *J. ACM*, 54(1):2:1–2:48, 2007.

[28] R. Kennaway, J. W. Klop, M. R. Sleep, and F. de Vries. Comparing curried and uncurried rewriting. *J. Symbolic Computation*, 21(1):15–39, 1996.

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

[30] C. Kop. *Higher Order Termination*. Ph.D. thesis, Vrije Universiteit Amsterdam, 2012.

[31] C. Kop and F. van Raamsdonk. A higher-order iterative path ordering. In I. Cervesato, H. Veith, and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2008)*, volume 5330 of *LNCS*, pages 697–711. Springer, 2008.

[32] L. Kovács, G. Moser, and A. Voronkov. On transfinite Knuth-Bendix orders. In N. Bjørner and V. Sofronie-Stokkermans, editors, *Conference on Automated Deduction (CADE-23)*, volume 6803 of *LNCS*, pages 384–399. Springer, 2011.

[33] L. Kovács and A. Voronkov. First-order theorem proving and Vampire. In N. Sharygina and H. Veith, editors, *Computer Aided Verification (CAV 2013)*, volume 8044 of *LNCS*, pages 1–35. Springer, 2013.

[34] M. Lifantsev and L. Bachmair. An LPO-based termination ordering for higher-order terms without λ-abstraction. In J. Grundy and M. C. Newey, editors, *Theorem Proving in Higher Order Logics (TPHOLs '98)*, volume 1479 of *LNCS*, pages 277–293. Springer, 1998.

[35] B. Löchner. Things to know when implementing KBO. *J. Autom. Reasoning*, 36(4):289–310, 2006.

[36] M. Ludwig and U. Waldmann. An extension of the Knuth-Bendix ordering with LPO-like properties. In N. Dershowitz and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence (LPAR 2007)*, volume 4790 of *LNCS*, pages 348–362. Springer, 2007.

[37] W. McCune. OTTER 3.3 reference manual. Technical Report 263, 2003.

[38] R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In J. A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, pages 371–443. Elsevier and MIT Press, 2001.

[39] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

[40] S. Schulz. System description: E 1.8. In K. L. McMillan, A. Middeldorp, and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-19)*, volume 8312 of *LNCS*, pages 735–743. Springer, 2013.

[41] C. Sternagel and R. Thiemann. Executable multivariate polynomials. *Archive of Formal Proofs*, 2010. `https://isa-afp.org/entries/Polynomials.shtml`, Formal proof development.

[42] C. Sternagel and R. Thiemann. Generalized and formalized uncurrying. In C. Tinelli and V. Sofronie-Stokkermans, editors, *Frontiers of Combining Systems (FroCoS 2011)*, volume 6989 of *LNCS*, pages 243–258. Springer, 2011.

[43] C. Sternagel and R. Thiemann. Formalizing Knuth-Bendix orders and Knuth-Bendix completion. In F. van Raamsdonk, editor, *Rewriting Techniques and Applications (RTA 2013)*, volume 21 of *LIPIcs*, pages 287–302. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2013.

[44] N. Sultana, J. C. Blanchette, and L. C. Paulson. LEO-II and Satallax on the Sledgehammer test bench. *J. Applied Logic*, 11(1):91–102, 2013.

[45] Y. Toyama. Termination of S-expression rewriting systems: Lexicographic path ordering for higher-order terms. In V. van Oostrom, editor, *Rewriting Techniques and Applications (RTA 2004)*, volume 3091 of *LNCS*, pages 40–54. Springer, 2004.

[46] D. A. Turner. A new implementation technique for applicative languages. *Software: Practice and Experience*, 9(1):31–49, 1979.

[47] C. Weidenbach, D. Dimova, A. Fietzke, R. Kumar, M. Suda, and P. Wischnewski. SPASS version 3.5. In R. A. Schmidt, editor, *Conference on Automated Deduction (CADE-22)*, volume 5663 of *LNCS*, pages 140–145. Springer, 2009.

[48] M. Wisniewski, A. Steen, K. Kern, and C. Benzmüller. Effective normalization techniques for HOL. In N. Olivetti and A. Tiwari, editors, *International Joint Conference on Automated Reasoning (IJCAR 2016)*, volume 9706 of *LNCS*, pages 362–370. Springer, 2016.

[49] H. Zankl, S. Winkler, and A. Middeldorp. Beyond polynomials and Peano arithmetic—Automation of elementary and ordinal interpretations. *J. Symb. Comput.*, 69:129–158, 2015.

[50] H. Zantema. Termination. In M. Bezem, J. W. Klop, and R. de Vrijer, editors, *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*, pages 181–259. Cambridge University Press, 2003.