EasyChair Preprint

№ 3191

A Knuth-Bendix-Like Ordering for Orienting Combinator Equations (Technical Report)

Ahmed Bhayat and Giles Reger

April 18, 2020

A Knuth-Bendix-Like Ordering for Orienting Combinator Equations (Technical Report)

Ahmed Bhayat[0000−0002−1343−5084] and Giles Reger[0000−0001−6353−952X]

University of Manchester, Manchester, UK

Abstract. We extend the graceful higher-order basic Knuth-Bendix order (KBO) of Becker et al. to an ordering that orients combinator equations left-to-right. The resultant ordering is highly suited to parameterising the first-order superposition calculus when dealing with the theory of higher-order logic, as it prevents inferences between the combinator axioms. We prove a number of desirable properties about the ordering including it having the subterm property for ground terms, being transitive and being well-founded. The ordering fails to be a reduction ordering as it lacks compatibility with certain contexts. We provide an intuition of why this need not be an obstacle when using it to parameterise superposition.

1 Introduction

There exists a wide range of methods for automated theorem proving in higher-order logic. Some provers such as AgsyHOL [16], Satallax [9] and Leo-II [4] implement dedicated higher-order proof calculi. A common approach, followed by the Leo-III prover [20], is to use a co-operative architecture with a dedicated higher-order prover working in conjunction with a first-order prover. It has long been part of theorem proving folklore that sound and complete translations from higher-order to first-order logic exist. Kerber [14] proves this result for a higher-order logic that does not assume comprehension axioms (otherwise known as applicative first-order logic). Thus, translating higher-order problems to first-order logic and running first-order provers on the translations is another method of automated higher-order theorem proving. Variations of this method are widely utilised by interactive theorem provers and their hammers such as Sledgehammer [17] and the CoqHammer [10].

Almost all translations to first-order logic translate λ-expressions using combinators. It is well known that the set of combinators S, K and I is sufficient to translate any λ-expression. For purposes of completeness, these combinators must be axiomatised: $\mathsf{S}\langle\tau_1,\tau_2,\tau_3\rangle\, x\, y\, z \;=\; x\, z\, (y\, z)$, $\mathsf{K}\langle\tau_1,\tau_2\rangle\, x\, y \;=\; x$ and $\mathsf{I}\langle\tau\rangle\, x \;=\; x$. If translating to a monomorphic logic a finite set of axioms cannot achieve completeness.

However, till now, translation based methods have proven disappointing and only achieved decent results with interactive theorem provers when the problems are first-order or nearly first-order [21]. One major reason for this is that inferences between combinator axioms can be hugely explosive. A common first-order proof calculus is superposition [18]. Consider a superposition inference from the K axiom onto the right-hand of the S axiom. The result is $\mathsf{S}\,\mathsf{K}\,y\,z = z$. There is little to restrict such inferences.

Superposition is parameterised by a simplification ordering and inferences are only carried out on the larger side of literals with respect to this ordering. Inferences are

not carried out at variables. Consider the **S**-, **K**- and **I**-axioms given above. There can clearly be no unifiers between a subterm of the left side of one axiom and the left side of another except at a variable. Thus, if a simplification ordering exists that orients the axioms left-to-right, inferences amongst the axioms would be impossible.

Currently, no such simplification ordering is known to exist and the authors suspect that no such ordering can exist. Whilst there is a large body of work on higher-order orderings, all either lack some property required for them to be simplifcation orderings or are unsuitable for orienting the combinator axioms. Jouannaud and Rubio introduced a higher-order version of the recursive path order called HORPO [13]. HORPO is compatible with β-reduction which suggests that without much difficulty it could be modified to be compatible with weak reduction. However, the ordering does not enjoy the subterm property, nor is it transitive. Likewise, is the case for orderings based on HORPO such as the computability path ordering [7] and the iterative HOIPO of Kop and Van Raamsdonk [15]. More recently, a pair of orderings for λ-free higher-order terms have been developed[2][6]. These orderings lack a specific monotonicity property, but this does not prevent their use in superposition [3]. However, neither ordering orients combinator axioms directly.

We investigate an extension of the graceful higher-order basic KBO $>_{\mathsf{hb}}$ introduced by Becker et al. [2]. Our new ordering, $>_{\mathsf{ski}}$, orients combinator equations left-to-right. Thus, if it is used to parameterise a superposition calculus, there can be no inferences among the axioms. The $>_{\mathsf{ski}}$ ordering lacks full compatibility with contexts which is normally a requirement for an ordering to parameterise superposition. In particular, the ordering is not compatible with the so-called unstable contexts. In an as yet unpublished paper, we show that this is not an obstacle to achieving completeness [5].

A complete superposition calculus for HOL already exists [3]. This calculus has the λ-calculus rather than combinatory logic as its underlying logic. It also employs higher-order unification. There appear to be two potential benefits to using a slightly modified first-order superposition calculus parameterised by our new ordering $>_{\mathsf{ski}}$ over lambda superposition as developed in [3].

- A superposition calculus parameterised by $>_{\mathsf{ski}}$ is far closer to standard first-order superposition than lambda superposition. Unification is first-order and there is no need to deal with binders and bound variables. This allows the re-use of the well-studied data-structures and algorithms used in first-order superposition [11][19].
- As discussed further in the conclusion (Section 7), the $>_{\mathsf{ski}}$ ordering allows the comparison of a larger class of non-ground terms than the ordering used in [3]. This results in fewer superposition inferences.

In Section 2, we provide the necessary preliminaries and then move on to the main contributions of this paper which are:

- Two approaches extending the $>_{\mathsf{hb}}$ ordering by first comparing terms by the length of the longest weak reduction from them. The approaches differ in the manner they compare non-ground terms. A useful trait for an ordering that parameterises superposition is to be able to compare a large class of non-ground terms since this reduces the number of inferences carried out. The most powerful method of defining a non-ground ordering $\succ$ is to *semantically lift* a ground ordering, i.e., to

define $t \succ s$ to hold iff $t\theta \succ s\theta$ for all grounding substitutions θ. Such an ordering in non-computable and both our methods attempt to approximate it (Section 3).

- A set of proofs that the introduced $>_{\mathsf{ski}}$ ordering enjoys the necessary properties required for its use within the superposition calculus (Section 4) and a set of examples demonstrating how the ordering applies to certain terms (Section 5).
- An extension of the previous idea to λ-*terms* (those containing λ-abstractions) with β-reduction replacing weak reduction (Section 6).

2 Preliminaries

Syntax of types and terms: we work in a polymorphic applicative first-order logic. Let $\mathcal{V}_{\mathsf{ty}}$ be a set of type variables and Σ_{ty} be a set of type constructors with fixed arities. It is assumed that a binary type constructor $\rightarrow$ is present in Σ_{ty} which is written infix. The set of types is defined:

$$\textbf{Polymorphic Types} \quad \tau ::= \kappa(\overline{\tau_n}) \mid \alpha \mid \tau \rightarrow \tau \quad \text{where } \alpha \in \mathcal{V}_{\mathsf{ty}} \text{ and } \kappa \in \Sigma_{\mathsf{ty}}$$

The notation $\overline{t_n}$ is used to denote a tuple or list of types or terms depending on the context. A type declaration is of the form $\Pi\, \overline{\alpha}\, .\sigma$ where σ is a type and all type variables in σ appear in $\overline{\alpha}$. Let Σ be a set of typed function symbols and $\mathcal{V}$ a set of variables with associated types. It is assumed that Σ contains the following function symbols, known as *basic combinators*:

$$\begin{array}{ll} \mathsf{S} : \Pi\alpha\tau\gamma.(\alpha \rightarrow \tau \rightarrow \gamma) \rightarrow (\alpha \rightarrow \tau) \rightarrow \alpha \rightarrow \gamma & \mathsf{I} : \Pi\alpha.\alpha \rightarrow \alpha \\ \mathsf{C} : \Pi\alpha\tau\gamma.(\alpha \rightarrow \tau \rightarrow \gamma) \rightarrow \tau \rightarrow \alpha \rightarrow \gamma & \mathsf{K} : \Pi\alpha\gamma.\alpha \rightarrow \gamma \rightarrow \alpha \\ \mathsf{B} : \Pi\alpha\tau\gamma.(\alpha \rightarrow \gamma) \rightarrow (\tau \rightarrow \alpha) \rightarrow \tau \rightarrow \gamma & \end{array}$$

The set of terms over Σ and $\mathcal{V}$ is defined below. In what follows, type subscripts, and at times even type arguments, are omitted.

$$\textbf{Terms} \quad \mathcal{T} ::= x \mid \mathsf{f}\langle \overline{\tau_n} \rangle \mid t_{1\tau_1 \rightarrow \tau_2} t_{2\tau_1}$$
$$\text{where } x \in \mathcal{V},\, t_1, t_2 \in \mathcal{T},\, \mathsf{f} \in \Sigma,\, \mathsf{f} : \Pi\, \overline{\alpha_n}\, .\sigma \text{ and } \overline{\tau_n} \text{ are types}$$

The type of the term $\mathsf{f}\langle \overline{\tau_n} \rangle$ is $\sigma\{\, \overline{\alpha_n} \rightarrow \overline{\tau_n} \,\}$. Following [2], terms of the form $t_1\, t_2$ are called applications. Non-application terms are called heads. A term can uniquely be decomposed into a head and n arguments. Let $t = \zeta\ \overline{t'_n}$. Then $head(t) = \zeta$ where ζ could be a variable or constant applied to possibly zero type arguments. The symbol $\boldsymbol{\mathcal{C}}_{\mathsf{any}}$ denotes a member of $\{\mathsf{S}, \mathsf{C}, \mathsf{B}, \mathsf{K}, \mathsf{I}\}$, whilst $\boldsymbol{\mathcal{C}}_{\mathsf{3}}$ denotes a member of $\{\mathsf{S}, \mathsf{C}, \mathsf{B}\}$. These symbols are only used when the combinator is assumed to have a full complement of arguments. Thus, in $\boldsymbol{\mathcal{C}}_{\mathsf{3}}\ \overline{t_n}$, $n \geq 3$ is assumed. The symbols $x, y, z \ldots$ are reserved for variables, $\mathsf{c}, \mathsf{d}, \mathsf{f} \ldots$ for non-combinator constants and ζ, ξ range over arbitrary symbols and, by an abuse of notation, at times even terms. A term is *ground* if it contains no variables and *term ground* if it contains no term variables.

Positions over terms: for a term t, if $t \in \mathcal{V}$ or $t = \mathsf{f}\langle \overline{\tau} \rangle$, then $pos(t) = \{\epsilon\}$ (type arguments have no position). If $t = t_1 t_2$ then $pos(t) = \{\epsilon\} \cup \{i.p \mid 1 \leq i \leq 2,\ p \in pos(t_i)\}$. Subterms at positions of the form $p.1$ are called *prefix* subterms and subterms

at positions of the form $p.2$ are known as *first-order* subterms. A position p is *strictly above* a position p' (denoted $p < p'$) if $\exists p''.p'' \neq \epsilon \ \wedge \ p' = p.p''$. Positions p and p' are *incomparable* (denoted $p \parallel p'$) if neither $p < p'$ nor $p' < p$, nor $p = p'$. By $|t|$, the number of symbols occurring in t is denoted. By $vars_{\#}(t)$ the multiset of variables in t is denoted. The expression $A \subseteq B$ means that either A is a subset of B or A is a submultiset of B depending on whether A and B are sets or multisets.

Stable subterms: we define a subset of first-order subterms called *stable* subterms. Let $\mathsf{LPP}(t, p)$ (LPP stands for Longest Proper Prefix) be a partial function that takes a term t and a position p and returns the longest proper prefix p' of p such that $head(t|_{p'})$ is not a partially applied combinator if such a position exists. For a position $p \in pos(t)$, p is a stable position in t if $\mathsf{LPP}(t, p)$ is not defined or $head(t|_{\mathsf{LPP}(t,p)})$ is not a combinator. A *stable subterm* is a subterm occurring at a stable position and is denoted $t\langle\langle u \rangle\rangle_p$. We call $t\langle\langle\rangle\rangle_p$ a *stable context* and drop the position where it is not relevant. For example, the subterm a is not stable in $\mathsf{f}\ (\mathbf{S}\ \mathsf{a}\ \mathsf{b}\ \mathsf{c})$, $\mathbf{S}\ (\mathbf{S}\ \mathsf{a})\ \mathsf{b}\ \mathsf{c}$ (in both cases, $head(t|_{\mathsf{LPP}(t,p)}) = \mathbf{S}$) and $\mathsf{a}\ \mathsf{c}$ (a is not a first-order subterm), but is in $\mathsf{g}\ \mathsf{a}\ \mathsf{b}$ and $\mathsf{f}\ (\mathbf{S}\ \mathsf{a})\ \mathsf{b}$. A subterm that is not stable is known as an *unstable* subterm.

The notation $t[u]_p$ denotes an arbitrary subterm u of t that occurs at position p and may be unstable. The notation $t[u_1, \ldots, u_n]$ (or $t[\overline{u_n}]$) denotes the term t containing n *non-overlapping* subterms u_1 to u_n. By $u[]_n$, we refer to a context with n non-overlapping holes. Whilst this resembles the notation for a term at position n, ambiguity is avoided by never using n to denote a position or p to denote a natural number.

Weak reduction: each combinator is defined by its characteristic equation; $\mathbf{S}\,x\,y\,z = x\,z\,(y\,z)$, $\mathbf{C}\,x\,y\,z = x\,z\,y$, $\mathbf{B}\,x\,y\,z = x\,(y\,z)$, $\mathbf{K}\,x\,y = x$ and $\mathbf{I}\,x = x$. A term t *weak-reduces* to a term t' in one step (denoted $t \longrightarrow_w t'$) if $t = u[s]_p$ and there exists a combinator axiom $l = r$ and substitution σ such that $l\sigma = s$ and $t' = u[r\sigma]_p$. The term $l\sigma$ in t is called a *weak redex* or just redex. By $\longrightarrow_w^*$, the reflexive transitive closure of $\longrightarrow_w$ is denoted. If term t weak-reduces to term t' in n steps, we write $t \longrightarrow_w^n t'$. Further, if there exists a weak-reduction path from a term t of length n, we say that $t \in n_w$. Weak-reduction is terminating and confluent as proved in [12]. By $(t)\downarrow^w$, we denote the term formed from t by contracting its leftmost redex.

The length of the longest weak reduction from a term t is denoted $\|t\|$. This measure is one of the crucial features of the ordering investigated in this paper.

2.1 A Maximal Weak-reduction Strategy

To show that the measure $\|\|$ is computable we provide a maximal weak-reduction strategy and prove its maximality. The strategy is used in a number of proofs later in the paper. It is in a sense equivalent to Barendregt's 'perpetual strategy' in the λ-calculus [1]. Our proof of its maximality follows the style of Van Raamsdonk et al. [22] in their proof of the maximality of a particular β-reduction strategy. We begin by proving the fundamental lemma of maximality for combinatory terms.

Lemma 1 (Fundamental Lemma of Maximality). $\|\mathcal{C}_{\mathsf{any}}\ \overline{t_n}\| = \|(\mathcal{C}_{\mathsf{any}}\ \overline{t_n})\downarrow^w\| + 1 + isK(\mathcal{C}_{\mathsf{any}}) \times \|t_2\|$ *where* $isK(\mathcal{C}_{\mathsf{any}}) = 1$ *if* $\mathcal{C}_{\mathsf{any}} = \mathbf{K}$ *and is* 0 *otherwise. The lemma holds for* $n \geq 3$ *if* $\mathcal{C}_{\mathsf{any}} \in \{\mathbf{S}, \mathbf{C}, \mathbf{B}\}$, $n \geq 2$ *if* $\mathcal{C}_{\mathsf{any}} = \mathbf{K}$ *and* $n \geq 1$ *otherwise.*

Proof. Assume that $\mathcal{C}_{\mathsf{any}} = \mathsf{K}$. Then any maximal reduction from $\mathsf{K}\ \overline{t_n}$ is of the form:

$$\begin{array}{rl} \mathsf{K}\, t_1\, t_2 \ldots t_n \longrightarrow_w^m & \mathsf{K}\, t_1'\, t_2' \ldots t_n' \\ \longrightarrow_w & t_1'\, t_3' \ldots t_n' \\ \longrightarrow_w^{m'} & s \end{array}$$

where $\|s\| = 0$, $t_1 \longrightarrow_w^{m_1} t_1' \ldots t_n \longrightarrow_w^{m_n} t_n'$, $\|t_2\| = m_2$ and $m = m_1 + \cdots + m_n$. Thus, $\|\mathsf{K}\ \overline{t_n}\| = \sum_{i=1}^{n} m_i + 1 + m'$. There is another method of reducing $\mathsf{K}\ \overline{t_n}$ to s:

$$\begin{array}{rl} \mathsf{K}\, t_1\, t_2 \ldots t_n \longrightarrow_w^{m_2} & \mathsf{K}\, t_1\, t_2' \ldots t_n \\ \longrightarrow_w & t_1\, t_3 \ldots t_n \\ \longrightarrow_w^{m-m_2} & t_1'\, t_3' \ldots t_n' \\ \longrightarrow_w^{m'} & s \end{array}$$

As the length of this reduction is the same as the previous reduction, it must be a maximal reduction as well. Therefore we have that:

$$\begin{array}{rl} \|\mathsf{K}\, t_1\, t_2 \ldots t_n\| = & m + m' + 1 \\ = & (m - m_2 + m') + m_2 + 1 \\ = & \|t_1\, t_3 \ldots t_n\| + \|t_2\| + 1 \end{array}$$

Conversely, assume that $\mathcal{C}_{\mathsf{any}}$ is not K. We prove that the formula holds if $\mathcal{C}_{\mathsf{any}} = \mathsf{S}$. The other cases are similar. If $\mathcal{C}_{\mathsf{any}} = \mathsf{S}$, any maximal reduction from $\mathsf{S}\ \overline{t_n}$ must be of the form:

$$\begin{array}{rl} \mathsf{S}\, t_1 \ldots t_n \longrightarrow_w^m & \mathsf{S}\, t_1' \ldots t_n' \\ \longrightarrow_w & t_1'\, t_3'\, (t_2'\, t_3')\, t_4' \ldots t_n' \\ \longrightarrow_w^{m'} & s \end{array}$$

where $\|s\| = 0$, $t_1 \longrightarrow_w^{m_1} t_1' \ldots t_n \longrightarrow_w^{m_n} t_n'$ and $m = m_1 + \cdots + m_n$. There is another method of reducing $\mathsf{S}\ \overline{t_n}$ to s:

$$\begin{array}{rl} \mathsf{S}\, t_1 \ldots t_n \longrightarrow_w & t_1\, t_3\, (t_2\, t_3)\, t_4 \ldots t_n \\ \longrightarrow_w^{m+m_3} & t_1'\, t_3'\, (t_2'\, t_3')\, t_4' \ldots t_n \\ \longrightarrow_w^{m'} & s \end{array}$$

Thus, we have that $\|\mathsf{S}\ \overline{t_n}\| = m + m' + 1 \leq m + m_3 + m' + 1 = \|(\mathsf{S}\ \overline{t_n})\!\downarrow^w\| + 1$. Since $m + m' + 1$ is the length of the maximal reduction, equality must hold.

Lemma 2. *Define a map F_∞ from $\mathcal{T}$ to $\mathcal{T}$ as follows:*

$$\begin{array}{lcl}
F_\infty(t) & = & t \quad \textit{if } \|t\| = 0 \\
F_\infty(\zeta\ \overline{t_n}) & = & \zeta\, t_1 \ldots t_{i-1}\, F_\infty(t_i)\, t_{i+1} \ldots t_n \\
 & & \textit{where } \|t_j\| = 0 \textit{ for } 1 \leq j < i \\
 & & \textit{and } \zeta \textit{ is not a fully applied combinator} \\
F_\infty(\boldsymbol{\mathcal{C}_3}\ \overline{t_n}) & = & (\boldsymbol{\mathcal{C}_3}\ \overline{t_n})\!\downarrow^w \\
F_\infty(\mathsf{I}\, t_1\, t_2 \ldots t_n) & = & t_1\, t_2 \ldots t_n \\
F_\infty(\mathsf{K}\, t_1\, t_2 \ldots t_n) & = & \begin{cases} t_1\, t_3 \ldots t_n & \textit{if } \|t_2\| = 0 \\ \mathsf{K}\, t_1\, F_\infty(t_2) \ldots t_n & \textit{otherwise} \end{cases}
\end{array}$$

The reduction strategy F_∞ is maximal.

Proof. By utilising Lemma 2.14 of [22], we have that F_∞ is maximal iff for all $m \geq 1$ and all t, $t \in m_w \implies F_\infty(t) \in (m-1)_w$. We proceed by induction on t.

If $t = \mathsf{f}\langle \overline{\tau} \rangle\ \overline{s'}\ u\ \overline{s_n}$ or $t = x\ \overline{s'}\ u\ \overline{s_n}$ where all members of $\overline{s'}$ are in normal form, $\|u\| > 0$, $u \in m^0_w, s_1 \in m^1_w \ldots s_n \in m^n_w$ and $m = m^0 + \cdots + m^n$, then $F_\infty(t) = \zeta\ \overline{s'}\ F_\infty(u)\ \overline{s_n}$. By the induction hypothesis $F_\infty(u) \in (m^0 - 1)_w$. Thus, $F_\infty(t) = \zeta\ \overline{s'}\ F_\infty(u)\ \overline{s_n} \in (m-1)_w$.

If $t = \boldsymbol{\mathcal{C}_{\mathsf{any}}}\ \overline{t_n}$, the proof splits into two:

$\boldsymbol{\mathcal{C}_{\mathsf{any}} \neq \mathsf{K}}$ or $\|t_1\| = 0$ Then $F_\infty(t) = (\boldsymbol{\mathcal{C}_{\mathsf{any}}}\ \overline{t_n})\!\downarrow^w$. By the fundamental lemma of maximality, we have $\|(\boldsymbol{\mathcal{C}_{\mathsf{any}}}\ \overline{t_n})\!\downarrow^w\| + 1 = \|\boldsymbol{\mathcal{C}_{\mathsf{any}}}\ \overline{t_n}\| \geq m$. Thus $\|(\boldsymbol{\mathcal{C}_{\mathsf{any}}}\ \overline{t_n})\!\downarrow^w\| \geq m-1$ and $F_\infty(t) \in (m-1)_n$.

$\boldsymbol{\mathcal{C}_{\mathsf{any}} = \mathsf{K}}$ and $\|t_1\| > 0$ By the fundamental lemma of maximality we have that $\|(\boldsymbol{\mathcal{C}_{\mathsf{any}}}\ \overline{t_n})\!\downarrow^w\| + \|t_1\| + 1 = \|\boldsymbol{\mathcal{C}_{\mathsf{any}}}\ \overline{t_n}\| \geq m$. By the induction hypothesis we have that $\|F_\infty(t_1)\| \geq \|t_1\| - 1$. Thus

$$\begin{array}{rr}
\|F_\infty(t)\| = & \|(t)\!\downarrow^w\| + \|F_\infty(t_1)\| + 1 \\
\geq & \|(t)\!\downarrow^w\| + \|t_1\| \\
= & \|t\| - 1 \\
\geq & m - 1
\end{array}$$

and so $F_\infty(t) \in (m-1)_w$.

3 Term Order

First, Becker et al.'s [2] graceful higher-order basic KBO is presented as it is utilised within our ordering. The presentation here differs slightly from that in [2] because we do not allow ordinal weightings and all function symbols have finite arities. Furthermore, we do not allow the use of different operators for the comparison of tuples, but rather restrict the comparison of tuples to use only the length-lexicographic extension of the base order. This is denoted $\gg_{\mathsf{hb}}^{\mathsf{length\_lex}}$. The length-lexicographic extension first compares the lengths of tuples and if these are equal, carries out a lexicographic comparison. For this section, terms are assumed to be untyped following the original presentation.

3.1 Graceful Higher-Order Basic KBO

Standard first-order KBO first compares the weights of terms, then compares their head-symbols and finally compares arguments recursively. When working with higher-order terms, the head symbol may be a variable. To allow the comparison of variable heads, a mapping ghd is introduced that maps variable heads to members of Σ that could possibly instantiate the head. This mapping *respects arities* if for any variable x, all members of $ghd(x)$ have arities greater or equal to that of x. The mapping can be extended to constant heads by taking $ghd(\mathsf{f}) = \{\mathsf{f}\}$. A substitution σ *respects* the mapping ghd, if for all variables x, $ghd(x\sigma) \subseteq ghd(x)$.

Let $\succ$ be a total well-founded ordering or *precedence* on Σ. The precedence $\succ$ is extended to arbitrary heads by defining $\zeta \succ \xi$ iff $\forall f \in ghd(\zeta)$ and $\forall g \in ghd(\xi), f \succ g$. Let w be a function from Σ to $\mathbb{N}$ that denotes the weight of a function symbol and $\mathcal{W}$ a function from $\mathcal{T}$ to $\mathbb{N}$ denoting the weight of a term. Let $\varepsilon \in \mathbb{N}_{>0}$. For all constants c, $w(\mathsf{c}) \geq \varepsilon$. The weight of a term is defined recursively:

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

The graceful higher-order basic Knuth-Bendix order $>_{\mathsf{hb}}$ is defined inductively as follows. Let $t = \zeta\ \bar{t}$ and $s = \xi\ \bar{s}$. Then $t >_{\mathsf{hb}} s$ if $vars_{\#}(s) \subseteq vars_{\#}(t)$ and any of the following are satisfied:

Z1 $\mathcal{W}(t) > \mathcal{W}(s)$
Z2 $\mathcal{W}(t) = \mathcal{W}(s)$ and $\zeta \succ \xi$
Z3 $\mathcal{W}(t) = \mathcal{W}(s), \zeta = \xi$ and $\bar{t} \gg_{\mathsf{hb}}^{\mathsf{length\_lex}} \bar{s}$

3.2 Combinator Orienting KBO

The combinator orienting KBO is the focus of this paper. It has the property that all ground instances of combinator axioms are oriented by it left-to-right. This is achieved by first comparing terms by the length of the longest weak reduction from the term and then using $>_{\mathsf{hb}}$. This simple approach runs into problems with regards to stability under substitution, a crucial feature for any ordering used in superposition.

Consider the terms $t = \mathsf{f}\,x\,\mathsf{a}$ and $s = x\,\mathsf{b}$. As the length of the maximum reduction from both terms is 0, the terms would be compared using $>_{\mathsf{hb}}$ resulting in $t \succ s$ as $\mathcal{W}(t) > \mathcal{W}(s)$. Now, consider the substitution $\theta = \{x \to \mathsf{I}\}$. Then, $\|s\theta\| = 1$ whilst $\|t\theta\| = 0$ resulting in $s\theta \succ t\theta$.

The easiest and most general way of obtaining an order which is stable under substitution would be to restrict the definition of the combinator orienting KBO to ground terms and then *semantically lift* it to non-ground terms as mentioned in the introduction. However, the semantic lifting of the ground order is non-computable and therefore useless for practical purposes. We therefore provide two approaches to achieving an ordering that can compare non-ground terms and is stable under substitution both of which approximate the semantic lifting. Both require some conditions on the forms of terms that can be compared. The first is simpler, but more conservative than the second.

First, in the spirit of Bentkamp et al. [3], we provide a translation that replaces "problematic" subterms of the terms to be compared with fresh variables. With this

approach, the simple variable condition of the standard KBO, $vars_{\#}(s) \subseteq vars_{\#}(t)$, ensures stability. However, this approach is over-constrained and prevents the comparison of terms such as $t = x\,\mathsf{a}$ and $s = x\,\mathsf{b}$ despite the fact that for all substitutions θ, $\|t\theta\| = \|s\theta\|$. Therefore, we present a second approach wherein no replacement of subterms occurs. This comes at the expense of a far more complex variable condition. Roughly, the condition stipulates that two terms are comparable if and only if the variables and relevant combinators are in identical positions in each.

Approach 1 Because the $>_{\mathsf{hb}}$ ordering is not defined over typed terms, type arguments are replaced by equivalent term arguments before comparison. The translation $(\!|\cdot|\!)$ from $\mathcal{T}$ to untyped terms is given below. First we define precisely the subterms that require replacing by variables.

Definition 1 (Type-1 term). *Consider a term t of the form $\boldsymbol{\mathcal{C}}_{\mathsf{any}}\ \overline{t_n}$. If there exists a position p such $t|_p$ is a variable, then t is a type-1 term.*

Definition 2 (Type-2 term). *A term $x\ \overline{t_n}$ where $n > 0$ is a type-2 term.*

The translation from polymorphic terms to untyped terms with problematic subterms replaced:

$$(\!|t|\!) = \begin{cases} \tau & t \text{ is a type variable } \tau \\ \kappa\,(\!|\,\overline{\sigma_n}\,|\!) & t = \kappa(\,\overline{\sigma_n}\,) \\ x & t \text{ is a term variable } x \\ x_t & t \text{ is a type-1 or type-2 term} \\ \mathsf{f}\,(\!|\,\overline{\tau_n}\,|\!) & t = \mathsf{f}\langle\,\overline{\tau_n}\,\rangle \\ (\!|t_1|\!)(\!|t_2|\!) & t = t_1\,t_2 \end{cases}$$

An untyped term t weak reduces to an untyped term t' in one step if $t = u[s]_p$ and there exists a combinator axiom $l = r$ and substitution σ such that $(\!|l|\!)\sigma = s$ and $t' = u[(\!|r|\!)\sigma]_p$. The aim of the ordering presented here is to parametrise the superposition calculus. For this purpose, the property that for terms t and t', $t \longrightarrow_w t' \implies t \succ t'$, is desired. To this end, the following lemma is proved.

Lemma 3. *For all term ground polymorphic terms t and t', it is the case that $t \longrightarrow_w t' \iff (\!|t|\!) \longrightarrow_w (\!|t'|\!)$.*

Proof. The $\implies$ direction is proved by induction on t. If the reduction occurs at ϵ, then t is of the form $\boldsymbol{\mathcal{C}}_{\mathsf{any}}\langle\,\overline{\tau_n}\,\rangle\ \overline{t_n}$. We prove that the lemma holds if $\boldsymbol{\mathcal{C}}_{\mathsf{any}} = \mathbf{S}$. The other cases are similar. If $t = \mathbf{S}\langle\tau_1, \tau_2, \tau_3\rangle\ \overline{t_n}$, then $(\!|t|\!) = \mathbf{S}\,\tau_1\,\tau_2\,\tau_3\,(\!|t_0|\!)(\!|t_1|\!)(\!|t_2|\!)$ $(\!|t_3 \ldots t_n|\!) \longrightarrow_w (\!|t_0|\!)(\!|t_2|\!)((\!|t_1|\!)(\!|t_2|\!))(\!|t_3 \ldots t_n|\!) = (\!|t_0\,t_2\,(t_1\,t_2)\ t_3 \ldots t_n|\!) = (\!|t'|\!)$. Now assume that the reduction does not occur at ϵ. In this case, $t = \xi\ \overline{t_n}$, $t_i \longrightarrow_w t_i'$ and $t' = \xi\,t_0 \ldots t_{i-1}\,t_i'\,t_{i+1} \ldots t_n$. By the induction hypothesis, $(\!|t_i|\!) \longrightarrow_w (\!|t_i'|\!)$. Thus, $(\!|t|\!) = (\!|\xi|\!)(\!|\,\overline{t_n}\,|\!) \longrightarrow_w (\!|\xi|\!)(\!|\,\overline{t_{i-1}}\,|\!)(\!|t_i'|\!)(\!|t_{i+1} \ldots t_n|\!) = (\!|t'|\!)$.

The $\impliedby$ direction can be proved in a nearly identical manner.

Corollary 1. *A straightforward corollary of the above lemma is that for all term-ground polymorphic terms t, $\|t\| = \|(\!|t|\!)\|$.*

The combinator orienting Knuth-Bendix order (approach 1) $>_{\mathsf{ski1}}$ is defined as follows. For terms t and s, let $t' = (\!|t|\!)$ and $s' = (\!|s|\!)$. Then $t >_{\mathsf{ski1}} s$ if $vars_{\#}(s') \subseteq vars_{\#}(t')$ and:

R1 $\|t'\| > \|s'\|$ or,
R2 $\|t'\| = \|s'\|$ and $t' >_{\mathsf{hb}} s'$.

Approach 2 Using approach 1, terms $t = y\,\mathsf{a}$ and $s = y\,\mathsf{b}$ are incomparable. Both are type-2 terms and therefore $(\!|t|\!) = x_t$ and $(\!|s|\!) = x_s$. The variable condition obviously fails to hold between x_t and x_s. Therefore, we consider another approach which does not replace subterms with fresh variables. We introduce a new translation $[\![]\!]$ from $\mathcal{T}$ to untyped terms that merely replaces type arguments with equivalent term arguments and does not affect term arguments at all. The simpler translation comes at the cost of a more complex variable condition. Before the revised variable definition can be provided, some further terminology requires introduction.

Definition 3 (Safe Combinator). *Let $\boldsymbol{\mathcal{C}}_{\mathsf{any}}$ occur in t at position p and let p' be the shortest prefix of p such that $head(t|_{p'})$ is a combinator and for all positions p'' between p and p', $head(t|_{p''})$ is a combinator. Let p'' be a prefix of p of length one shorter than p' if such a position exists and ϵ otherwise. Then $\boldsymbol{\mathcal{C}}_{\mathsf{any}}$ is* safe *in t if $t|_{p'}$ is ground and $head(t|_{p''}) \notin \mathcal{V}$ and* unsafe *otherwise.*

Intuitively, unsafe combinators are those that could affect a variable on a longest reduction path or could become applied to a subterm of a substitution. For example, all combinators in the term $\mathsf{S}\,(\mathsf{K}\,\mathsf{I})\,\mathsf{a}\,x$ are unsafe because they affect x, whilst the combinator in $\mathsf{f}\,(\mathsf{I}\,\mathsf{b})\,y$ is safe. The combinators in $x\,(\mathsf{S}\,\mathsf{I})\,\mathsf{a}$ are unsafe because they could potentially interact with a term substituted for x.

Definition 4. *We say a subterm is* top-level *in a term t if it doesn't appear beneath an applied variable or fully applied combinator head in t.*

Definition 5 (Safe). *Let t_1 and t_2 be untyped terms. The predicate $safe(t_1, t_2)$ holds if for every position p in t_2 such that $t_2|_p = \boldsymbol{\mathcal{C}}_{\mathsf{any}}\ \overline{t_n}$ and $\boldsymbol{\mathcal{C}}_{\mathsf{any}}$ (not necessarily fully applied) is unsafe, then $t_1|_p = \boldsymbol{\mathcal{C}}_{\mathsf{any}}\ \overline{s_n}$ and for $1 \leq i \leq n$, $\|s_i\| \geq \|t_i\|$. Further, for all p in $pos(t_2)$ such that $t_2|_p = x\ \overline{t_n}$, then $t_1|_p = x\ \overline{s_n}$ and for $1 \leq i \leq n$, $\|s_i\| \geq \|t_i\|$.*

The definition of $safe$ ensures that if $safe(t, s)$ and $\|t\| \geq \|s\|$, then $\|t\sigma\| \geq \|s\sigma\|$ for any substitution σ a result we prove in Lemma 13. Consider terms $t = x\,(\mathsf{I}(\mathsf{I}\,(\mathsf{I}\,\mathsf{a})))\,\mathsf{b}$ and $s = x\,\mathsf{a}\,(\mathsf{I}\,(\mathsf{I}\,\mathsf{b}))$. We have that $\|t\| = 3 > \|s\| = 2$. However, it is not the case that $safe(t, s)$ because the condition that $\|t_i\| \geq \|s_i\|$ for all i is not met. $\|t_2\| = \|b\| = 0 < 2 = \|\mathsf{I}\,(\mathsf{I}\,\mathsf{b})\| = \|s_2\|$. Now consider the substitution $\sigma = \{x \to \mathsf{S}\,\mathsf{c}\}$. Because this substitution duplicates the second argument in s and t, $\|t\sigma\| = 4 < \|s\sigma\| = 5$ showing the importance of the $safe$ predicate in ensuring stability.

We draw out some obvious consequences of the definition of safety. Firstly, the predicate enjoys the subterm property in the following sense. If p is a position defined in terms t_1 and t_2, then $safe(t_1, t_2) \implies safe(t_1|_p, t_2|_p)$. Secondly, the predicate is transitive; $safe(t_1, t_2) \wedge safe(t_2, t_3) \implies safe(t_1, t_3)$.

There is a useful property that holds for non-ground terms t and s such that $safe(t, s)$.

Definition 6 (Semisafe). *Let t and s be untyped terms. Let $\mathcal{C}_{\mathsf{any}}\ \overline{s_n}$ be a term that occurs in s at p such that all head symbols above $\mathcal{C}_{\mathsf{any}}$ in s are combinators. Then $semisafe(t,s)$ if $t|_p = \mathcal{C}_{\mathsf{any}}\ \overline{t_n}$ and for $1 \leq i \leq n$, $\|t_i\| \geq \|s_i\|$.*

It is clearly the case that (t not ground) $\wedge$ (s not ground) $\wedge$ $safe(t,s) \implies semisafe(t,s)$. The implication does not hold in the other direction. A useful property of $semisafe$ is that it is stable under head reduction. If for terms t and s that reduce at their heads to t' and s' respective, we have $semisafe(t,s)$, then we have $semisafe(t',s')$.

Variable Condition:
Let $t' = [\![t]\!]$ and $s' = [\![s]\!]$ for polymorphic terms t and s. Let A be the multiset of all top-level, non-ground, first-order subterms in s' of the form $x\ \overline{s_n}$ (n may be 0) or $\mathcal{C}_{\mathsf{any}}\ \overline{t_n}$. Let B be a similarly defined multiset of subterms of t'. Then, $var\_cond(t',s')$ holds if there exists an injective total function f from A to B such that f only associates terms t_1 and t_2 if $safe(t_1,t_2)$.

For example $var\_cond(t,s)$ holds where $t = \mathsf{f}\,y\,(x\,\mathsf{a})$ and $s = \mathsf{g}\,(x\,b)$. In this case $A = \{x\,\mathsf{b}\}$ and $B = \{y, x\,\mathsf{a}\}$. There exists and injective total function from A to B that matches the requirements by relating $x\,\mathsf{b}$ to $x\,\mathsf{a}$. However, the variable condition does not hold in either direction if $t = \mathsf{f}\,y\,(x\,\mathsf{a})$ and $s = \mathsf{g}\,(x\,(\mathsf{I}\,\mathsf{b}))$. In this case, $x\,(\mathsf{I}\,\mathsf{b})$ cannot be related to $x\,a$ since the condition that $\|a\| \geq \|\mathsf{I}\,\mathsf{b}\|$ is not fulfilled.

We now define the combinator orienting Knuth-Bendix order (approach 2) $>_{\mathsf{ski}}$. For terms t and s, let $t' = [\![t]\!]$ and $s' = [\![s']\!]$. Then $t >_{\mathsf{ski}} s$ if $var\_cond(t',s')$ and:

R1 $\|t'\| > \|s'\|$ or,
R2 $\|t'\| = \|s'\|$ and $t' >_{\mathsf{hb}} s'$.

Lemma 4. *For all ground instances of combinator axioms $l \approx r$, we have $l >_{\mathsf{ski}} r$.*

Proof. Since for all ground instances of the axioms $l \approx r$, we have $\|l\| > \|r\|$, the theorem follows by an application of R1.

It should be noted that for a non-ground instances of an axiom $l \approx r$, we do **not** necessarily have $l >_{\mathsf{ski}} r$ since l and r may be incomparable. This is no problem since the definition of $>_{\mathsf{ski}}$ could easily be amended to have $l >_{\mathsf{ski}} r$ by definition if $l \approx r$ is an instance of an axiom. Lemma 4 ensures that stability under substitution would not be affected by such an amendment.

4 Properties

Various properties of the order $>_{\mathsf{ski}}$ are proved here. The proofs can easily be modified to hold for the less powerful $>_{\mathsf{ski1}}$ ordering. In general, for an ordering to parameterise a superposition calculus, it needs to be a *simplification* ordering [18]. That is, superposition is parameterised by an irreflexive, transitive, total on ground-terms, compatible with contexts, stable under substitution and well-founded binary relation. Compatibility

with contexts can be relaxed at the cost of extra inferences [8,3,5]. A desirable property to have in our case is coincidence with first-order KBO, since without this, the calculus would not behave on first-order problems as standard first-order superposition would.

Theorem 1 (Irreflexivity). *For all terms s, it is not the case that $s >_{\mathsf{ski}} s$.*

Proof. Let $s' = [\![s]\!]$. It is obvious that $\|s'\| = \|s'\|$. Therefore $s >_{\mathsf{ski}} s$ can only be derived by rule R2. However, this is precluded by the irreflexivity of $>_{\mathsf{hb}}$.

Theorem 2 (Transitivity). *For terms s, t and u, if $s >_{\mathsf{ski}} t$ and $t >_{\mathsf{ski}} u$ then $s >_{\mathsf{ski}} u$.*

Proof. Let $s' = [\![s]\!]$, $t' = [\![t]\!]$ and $u' = [\![u]\!]$. First we prove that if $var\_cond(s', t')$ and $var\_cond(t', u')$, then $var\_cond(s', u')$. Since $var\_cond(t', u')$ holds, there exists a function f_1 from the the multiset of top-level, non-ground, first-order subterms in u' of the form $x\ \overline{args}$ or $\boldsymbol{\mathcal{C}}_{\mathsf{any}}\ \overline{args}$ to the like multiset in t' that meets the given requirements. There is a similar function f_2 for s' and t'. We show that $f_2 \circ f_1$ is a function with the desired characteristics for terms u' and s'. Since f_1 and f_2 are both injective and total, $f_2 \circ f_1$ must be injective and total. It remains to prove that $f_2 \circ f_1$ only relates a subterm u_1 of u' to a subterm s_1 of s' if $safe(u_1, s_1)$. This is a straightforward consequence of the transitivity of $safe$.

If $\|s'\| > \|t'\|$ or $\|t'\| > \|u'\|$ then $\|s'\| > \|u'\|$ and $s >_{\mathsf{ski}} u$ follows by an application of rule R1. Therefore, suppose that $\|s'\| = \|t'\| = \|u'\|$. Then it must be the case that $s' >_{\mathsf{hb}} t'$ and $t' >_{\mathsf{hb}} u'$. It follows from the transitivity of $>_{\mathsf{hb}}$ that $s' >_{\mathsf{hb}} u'$ and thus $s >_{\mathsf{ski}} u$.

Theorem 3 (Ground Totality). *Let s and t be ground terms that are not syntactically equal. Then either $s >_{\mathsf{ski}} t$ or $t >_{\mathsf{ski}} s$.*

Proof. Let $s' = [\![s]\!]$ and $t' = [\![t]\!]$. If $\|s'\| \neq \|t'\|$ then by R1 either $s >_{\mathsf{ski}} t$ or $t >_{\mathsf{ski}} s$. Otherwise, s' and t' are compared using $>_{\mathsf{hb}}$ and either $t' >_{\mathsf{hb}} s'$ or $s' >_{\mathsf{hb}} t'$ holds by the ground totality of $>_{\mathsf{hb}}$ and the injectivity of $[\![]\!]$.

Theorem 4 (Subterm Property for Ground Terms). *If t and s are ground and t is a proper subterm of s then $s >_{\mathsf{ski}} t$.*

Proof. Let $s' = [\![s]\!]$ and $t' = [\![t]\!]$. Since t is a subterm of s, t' is a subterm of s' and $\|s'\| \geq \|t'\|$ because any weak reduction in t' is also a weak reduction in s'. If $\|s'\| > \|t'\|$, the theorem follows by an application of R1. Otherwise s' and t' are compared using $>_{\mathsf{hb}}$ and $s' >_{\mathsf{hb}} t'$ holds by the subterm property of $>_{\mathsf{hb}}$. Thus $s >_{\mathsf{ski}} t$.

Next, a series of lemmas are proved that are utilised in the proof of the ordering's compatibility with contexts and stability under substitution. We prove two monotonicity properties Theorems 5 and 6. Both hold for non-ground terms, but to show this, it is required to show that the variable condition holds between terms $u[t]$ and $u[s]$ for t and s such that $t >_{\mathsf{ski}} s$. To avoid this complication, we prove the Lemmas for ground terms which suffices for our purposes. To avoid clutter, assume that terms mentioned in the statement of Lemmas 5 - 16 are all untyped, formed by translating polymorphic terms.

Lemma 5. $\|\zeta\ \overline{t_n}\| = \sum_{i=1}^{n} \|t_i\|$ *if ζ is not a fully applied combinator.*

Proof. Trivial.

Lemma 6. *Let $t = \zeta\ \overline{t_n}$. Then $\|t\| > \sum_{i=1}^{n} \|t_i\|$ if ζ is a fully applied combinator.*

Proof. Since any reduction from a t_i is also a reduction from t and there exists the reduction at the root, we have that $\|t\| \geq \sum_{i=1}^{n} \|t_i\| + 1$ which implies $\|t\| > \sum_{i=1}^{n} \|t_i\|$.

Lemma 7. *Let $\overline{t_n}$ be terms such that for each t_i, $head(t_i) \notin \{\mathsf{I}, \mathsf{K}, \mathsf{B}, \mathsf{C}, \mathsf{S}\}$. Let $\overline{t'_n}$ be terms with the same property. Moreover, let $\|t_i\| \geq \|t'_i\|$ for $1 \leq i \leq n$. Let $s = u[\overline{t_n}]$ and $s' = u[\overline{t'_n}]$ where each t_i and t'_i is at position p_i in s and s'. If the F_∞ redex in s is within t_i for some i, then the F_∞ redex in s' is within t'_i unless t'_i is in normal form.*

Proof. Proof is by induction on $|s| + |s'|$. If u has a hole at head position, then $s = \mathsf{f}\ \overline{r_m}\ \overline{s_{m''}}$ and $s' = \mathsf{g}\ \overline{v_{m'}}\ \overline{s'_{m''}}$ where $t_1 = \mathsf{f}\ \overline{r_m}$ and $t'_1 = \mathsf{g}\ \overline{v_{m'}}$. Assume that the F_∞ redex of s is in t_1. Further, assume that $\|t'_1\| > 0$. Then, for some i in $\{1 \ldots m'\}$, it must be the case that $\|v_i\| > 0$. Let j be the smallest index such that $\|v_j\| > 0$. Then by the definition of F_∞, $F_\infty(s') = \mathsf{g}\ v_1 \ldots v_{j-1}\ F_\infty(v_j)\ v_{j+1} \ldots v_{m'}\ \overline{s'_{m''}}$ and the F_∞ redex of s' is in t'_1.

Suppose that the F_∞ redex of s is not in t_1. This can only be the case if $\|t_1\| = 0$ in which case $\|t'_1\| = 0$ as well. In this case, by the definition of F_∞, $F_\infty(s) = \mathsf{f}\ \overline{r_m}\ s_1 \ldots s_{i-1}\ F_\infty(s_i)\ s_{i+1} \ldots s''_m$ where $\|s_j\| = 0$ for $1 \leq j < i$. Without loss of generality, assume that the F_∞ redex of s_i occurs inside t_i. Then t'_i must be a subterm of s'_i. Assume that $\|t'_i\| > 0$ and thus $\|s'_i\| > 0$. Since for all i, s_i and s'_i only differ at positions where one contains a t_j and the other contains a t'_j and $\|t_i\| \geq \|t'_i\|$ for $1 \leq i \leq m''$, we have that $\|s_j\| = 0$ implies $\|s'_j\| = 0$. Thus, using the definition of F_∞, $F_\infty(s') = \mathsf{g}\ \overline{v_{m'}}\ s'_1 \ldots s'_{i-1}\ F_\infty(s'_i)\ s'_{i+1} \ldots s'_{m''}$. The induction hypothesis can be applied to s_i and s'_i to conclude that the F_∞ redex of s'_i occurs inside t'_i. The lemma follows immediately.

If u does not have a hole at its head, then $s = \zeta\ \overline{s_n}$ and $s' = \zeta\ \overline{s'_n}$ where ζ is not a fully applied combinator other than K (if it was, the F_∞ redex would be at the head).

If ζ is not a combinator, the proof follows by a similar induction to above. Therefore, assume that $\zeta = \mathsf{K}$. It must be the case that $\|s_2\| > 0$ otherwise the F_∞ redex in s would be at the head and not within a t_i. By the definition of F_∞, $F_\infty(s) = \mathsf{K}\ s_1\ F_\infty(s_2)\ s_3 \ldots s_n$. Let the F_∞ redex of s_2 occur inside t_j. Then t'_j is a subterm of s'_2. If $\|t'_j\| > 0$ then $\|s'_2\| > 0$ and $F_\infty(s') = \mathsf{K}\ s'_1\ F_\infty(s'_2)\ s'_3 \ldots s'_n$. By the induction hypothesis, the F_∞ redex of s'_2 occurs in t'_j.

Lemma 8. *Let $\overline{t_n}$ be terms such that for $1 \leq i \leq n$, $head(t_i) \notin \{\mathsf{I}, \mathsf{K}, \mathsf{B}, \mathsf{C}, \mathsf{S}\}$. Then for all contexts $u[]_n$, if $u[\overline{t_n}] \longrightarrow_w u'$ then either:*

1. *$\exists i.u' = u[t_1, \ldots, \hat{t_i}, \ldots, t_n]$ where $t_i \longrightarrow_w \hat{t_i}$ or*
2. *$u' = \hat{u}\{x_1 \to t_1, \ldots, x_n \to t_n\}$ where $u[x_1, \ldots, x_n] \longrightarrow_w \hat{u}$*

Proof. Let $s = u[\overline{t_n}]$ and let $p_1, \ldots p_n$ be the positions of $\overline{t_n}$ in s. Since s is reducible, there must exist a p such that $s|_p$ is a redex.

If $p > p_i$ for some i, there exists a $p' \neq \epsilon$ such that $p = p_i p'$. Then, $u[t_1, \ldots, t_i, \ldots, t_n]|_{p_i} = t_i[\mathcal{C}_{\mathsf{any}}\ \overline{r_n}]_{p'} \longrightarrow_w t_i[(\mathcal{C}_{\mathsf{any}}\ \overline{r_n})\downarrow^w]_{p'}$. Let $\hat{t_i} = t_i[(\mathcal{C}_{\mathsf{any}}\ \overline{r_n})\downarrow^w]_{p'}$. We thus have that $t_i \longrightarrow_w \hat{t_i}$ and thus $u[t_1, \ldots, t_i, \ldots, t_n] \longrightarrow_w u[t_1, \ldots, \hat{t_i}, \ldots, t_n]$.

It cannot be the case that $p = p_i$ for any i because $head(t_i)$ is not a combinator for any t_i. In the case where $p < p_i$ or $p \parallel p_i$ for all i, we have that $u[\overline{t_n}] = (u[\overline{x_n}])\sigma$ and $u[\overline{x_n}]|_p$ is a redex where $\sigma = \{\overline{x_n} \to \overline{t_n}\}$. Let $\hat{u}$ be formed from $u[\overline{x_n}]$ by reducing its redex at p. Then :

$$\begin{aligned} s = u[\overline{t_n}] &= (u[\overline{x_n}])\sigma \\ &\longrightarrow_w \hat{u}\sigma \\ &= \hat{u}\{x_1 \to t_1 \ldots x_n \to t_n\} \end{aligned}$$

Lemma 9. *Let $\overline{t_n}$ be terms such that for each t_i, $head(t_i) \notin \{\mathsf{I}, \mathsf{K}, \mathsf{B}, \mathsf{C}, \mathsf{S}\}$. Let $\overline{t'_n}$ be terms with the same property. Then:*

1. *If $\|t_i\| = \|t'_i\|$ for all i in $\{1, \ldots, n\}$, then $\|u[\overline{t_n}]\| = \|u[\overline{t'_n}]\|$ for all n holed contexts u.*
2. *If $\|t_j\| > \|t'_j\|$ for some $j \in \{1, \ldots, n\}$ and $\|t_i\| \geq \|t'_i\|$ for $i \neq j$, then $\|u[\overline{t_n}]\| > \|u[\overline{t'_n}]\|$ for all n holed contexts u.*

Proof. Let $p_1, \ldots, p_n$ be the positions of the holes in u and let $s = u[\overline{t_n}]$ and $s' = u[\overline{t'_n}]$. Proof is by induction on $\|s\| + \|s'\|$. We prove part (1) first:

Assume that $\|u[\overline{t_n}]\| = 0$. Then $\|t_i\| = 0$ for $1 \leq i \leq n$. Now assume that $\|u[\overline{t'_n}]\| \neq 0$. Then there must exist some position p such that $s'|_p$ is a redex. We have that $p \neq p_i$ for all p_i as $head(t'_i) \notin \{\mathsf{I}, \mathsf{K}, \mathsf{B}, \mathsf{C}, \mathsf{S}\}$. Assume $p > p_i$ for some p_i. But then, $\|t'_i\| > 0$ which contradicts the fact that $\|t_i\| = \|t'_i\|$ for all i. Therefore, for all p_i either $p < p_i$ or $p \parallel p_i$. But then, if $s'|_p$ is a redex, so must $s|_p$ be, contradicting the fact that $\|u[\overline{t_n}]\| = 0$. Thus, we conclude that $\|u[\overline{t'_n}]\| = 0$.

Assume that $\|u[\overline{t_n}]\| > 0$. Let $u' = F_\infty(s)$. By Lemma 8 either $u' = u[t_1, \ldots, \hat{t_i}, \ldots, t_n]$ where $t_i \longrightarrow_w \hat{t_i}$ for $1 \leq i \leq n$ or $u' = \hat{u}\{\overline{x_n} \to \overline{t_n}\}$ where $u[\overline{x_n}] \longrightarrow_w \hat{u}$. In the first case, by Lemma 7 and $\|t_i\| = \|t'_i\|$ we have $F_\infty(s') = u'' = u[t'_1, \ldots, \hat{t'_i}, \ldots, t'_n]$ where $t'_i \longrightarrow_w \hat{t'_i}$. By the induction hypothesis $\|u'\| = \|u''\|$ and thus $\|s\| = \|s'\|$. In the second case, $F_\infty(s') = u'' = \hat{u}\{\overline{x_n} \to \overline{t'_n}\}$ where $u[\overline{x_n}] \longrightarrow_w \hat{u}$. Again, the induction hypothesis can be used to show $\|u'\| = \|u''\|$ and the theorem follows.

We now prove part (2); $\|u[\overline{t_n}]\|$ must be greater than 0. Again, let $u' = F_\infty(s)$ and $u'' = F_\infty(s')$. If $u' = u[t_1, \ldots, \hat{t_i}, \ldots, t_n]$ and $\|t'_i\| \neq 0$, then by Lemma 7 $u'' = u[t'_1, \ldots, \hat{t'_i}, \ldots, t'_n]$ where $t'_i \longrightarrow_w \hat{t'_i}$ unless $\|t'_i\| = 0$ and the lemma follows by the induction hypothesis.

If $\|t'_i\| = 0$, consider terms u' and s'. If $\|\hat{t_i}\| > 0$ or $\|t_j\| > \|t'_j\|$ for some $j \neq i$, then the induction hypothesis can be used to show $\|u'\| > \|s'\|$ and therefore $\|s\| = \|u'\| + 1 > \|s'\|$. Otherwise, $\|t_j\| = \|t'_j\|$ for all $j \neq i$ and $\|\hat{t_i}\| = 0 = \|t'_i\|$. Part 1 of this lemma can be used to show that $\|u'\| = \|s'\|$ and thus $\|s\| = \|u'\| + 1 > \|s'\|$. If $u' = \hat{u}\{\overline{x_n} \to \overline{t_n}\}$, then $u'' = \hat{u}\{\overline{x_n} \to \overline{t'_n}\}$ and the lemma follows by the induction hypothesis.

Theorem 5 (Compatibility with Contexts). *For ground terms s and t, such that $head(s)$, $head(t) \notin \{\mathsf{I}, \mathsf{K}, \mathsf{B}, \mathsf{C}, \mathsf{S}\}$, and $s >_{\mathsf{ski}} t$, then $u[s] >_{\mathsf{ski}} u[t]$ for all ground contexts $u[]$.*

Proof. Let $s' = [\![s]\!]$, $t' = [\![t]\!]$ and $u' = [\![u]\!]$. By Lemma 9 Part 2, we have that if $\|s'\| > \|t'\|$, then $\|u'[s']\| > \|u'[t']\|$. Thus, if $s >_{\mathsf{ski}} t$ was derived by R1, $u[s] >_{\mathsf{ski}} u[t]$

follows by R1. Otherwise, $s >_{\mathsf{ski}} t$ is derived by R2 and $\|s'\| = \|t'\|$. By Lemma 9 Part 1, $\|u'[s']\| = \|u'[t']\|$ follows. Thus, $u'[s']$ is compared with $u'[t']$ by R2 and $u[s] >_{\mathsf{ski}} u[t]$ by the compatibility with contexts of $>_{\mathsf{hb}}$.

Corollary 2 (Compatibility with Arguments). *If $s >_{\mathsf{ski}} t$ and $head(s)$ and $head(t)$ are not combinators then $s\,u >_{\mathsf{ski}} t\,u$.*

Proof. This is just a special case of Theorem 5.

Lemma 10. $\|s\| > \|t\| \implies \|u\langle\langle s\rangle\rangle\| > \|u\langle\langle t\rangle\rangle\|$ *and* $\|s\| = \|t\| \implies \|u\langle\langle s\rangle\rangle\| = \|u\langle\langle t\rangle\rangle\|$.

Proof. Proceed by induction on the size of the context u. If u is the empty context, both parts of the theorem hold trivially.

The inductive case is proved for the first implication of the lemma first. If u is not the empty context, $u\langle\langle s\rangle\rangle$ is of the form $u'\langle\langle\zeta\, t_1 \ldots t_{i-1}, s, t_{i+1} \ldots t_n\rangle\rangle$. By the definition of a stable subterm ζ cannot be a fully applied combinator and thus by Lemma 5 we have :

$$\begin{aligned}\|\zeta\, t_1 \ldots t_{i-1}, s, t_{i+1} \ldots t_n\| &= \sum_{\substack{j=1\\ j\neq i}}^{n} \|t_j\| + \|s\| \\ &> \sum_{\substack{j=1\\ j\neq i}}^{n} \|t_j\| + \|t\| \\ &= \|\zeta\, t_1 \ldots t_{i-1}, t, t_{i+1} \ldots t_n\|\end{aligned}$$

If ζ is not a combinator, then $\|u'\langle\langle\zeta\, t_1 \ldots t_{i-1}, s, t_{i+1} \ldots t_n\rangle\rangle\| > \|u'\langle\langle\zeta\ t_1 \ldots t_{i-1}, t, t_{i+1} \ldots t_n\rangle\rangle\|$ follows from Lemma 9 Part 2. Otherwise, ζ is a partially applied combinator and u' is a smaller stable context than u. The induction hypothesis can be used to conclude that $\|u'\langle\langle\zeta\, t_1 \ldots t_{i-1}, s, t_{i+1} \ldots t_n\rangle\rangle\| > \|u'\langle\langle\zeta\, t_1 \ldots t_{i-1}, t, t_{i+1} \ldots t_n\rangle\rangle\|$ and thus that $\|u\langle\langle s\rangle\rangle\| > \|u\langle\langle t\rangle\rangle\|$. The proof of the inductive case for the second implication of the lemma is almost identical.

Theorem 6 (Compatibility with Stable Contexts). *For all stable ground contexts $u\langle\langle\rangle\rangle$ and ground terms s and t, if $s >_{\mathsf{ski}} t$ then $u\langle\langle s\rangle\rangle >_{\mathsf{ski}} u\langle\langle t\rangle\rangle$.*

Proof. If $\|s\| > \|t\|$ then by Lemma 10, $\|u\langle\langle s\rangle\rangle\| > \|u\langle\langle t\rangle\rangle\|$ holds and then by an application of R1 we have $u\langle\langle s\rangle\rangle >_{\mathsf{ski}} u\langle\langle t\rangle\rangle$. Otherwise, if $\|s\| = \|t\|$, then by Lemma 10 we have that $\|u\langle\langle s\rangle\rangle\| = \|u\langle\langle t\rangle\rangle\|$. Thus $u\langle\langle s\rangle\rangle$ and $u\langle\langle t\rangle\rangle$ are compared using $>_{\mathsf{hb}}$. By the compatibility with contexts of $>_{\mathsf{hb}}$, $[\![u\langle\langle s\rangle\rangle]\!] >_{\mathsf{hb}} [\![u\langle\langle t\rangle\rangle]\!]$ holds and then by ofan application of R2 $u\langle\langle s\rangle\rangle >_{\mathsf{ski}} u\langle\langle t\rangle\rangle$ is true.

We next prove stability under substitution. In order to prove this, it needs to be shown that for untyped terms s and t and all substitutions σ:

1. $var\_cond(s,t)$ implies $var\_cond(s\sigma, t\sigma)$.
2. $var\_cond(s,t)$ and $\|s\| \geq \|t\|$ imply $\|s\sigma\| \geq \|t\sigma\|$

The first is proved in Lemma 15. A slightly generalised version of (2) is proved in Lemma 14. Lemmas 11 - 13 are helper lemmas used in the proof of the above two properties.

Lemma 11. *For a single hole context $u\langle\langle\rangle\rangle$ such that the hole does not occur below a fully applied combinator and any term t, $\|u\langle\langle t\rangle\rangle\| = \|u\langle\langle\rangle\rangle\| + \|t\|$.*

Proof. Proceed by induction on the size of u. If u is the empty context the theorem follows trivially. Therefore, assume that $u = \mathsf{f}\, t \ldots t_{i-1}\, u'\langle\langle\rangle\rangle\, t_{i+1} \ldots t_n$. By Lemma 5 $\|u\langle\langle\rangle\rangle\| = \sum_{i=1}^{n} \|t_i\| + \|u'\langle\langle\rangle\rangle\|$. Because u' is a smaller context than u, the induction hypothesis can be used to show $\|u'\langle\langle t\rangle\rangle\| = \|u'\langle\langle\rangle\rangle\| + \|t\|$. Thus:

$$\begin{aligned}
\|u\langle\langle t\rangle\rangle\| &= \sum_{i=1}^{n} \|t_i\| + \|u'\langle\langle t\rangle\rangle\| \\
&= \sum_{i=1}^{n} \|t_i\| + \|u'\langle\langle\rangle\rangle\| + \|t\| \\
&= \|u\langle\langle\rangle\rangle\| + \|t\|
\end{aligned}$$

proving the theorem.

Lemma 12. *Let $\overline{t_n}$ and $\overline{s_n}$ be terms such that for $n_1 \ldots n_n \in \mathbb{N}$ and for $1 \leq i \leq n$, $\|t_i\| \geq \|s_i\| + n_i$. Further, let $t = t_1\, t_2 \ldots t_n$ and $s = s_1\, s_2 \ldots s_n$. Assume that $semisafe(t, s)$ holds. Then $\|t\| \geq \|s\| + \sum_{i=1}^{n} n_i$.*

Proof. Assume that $head(s)$ is not a fully applied combinator. By an application of Lemmas 5 and 6 it follows that, $\|t\| \geq \|t_1\| + \cdots + \|t_n\| \geq \|s_1\| + \cdots + \|s_n\| + n_1 + \cdots + n_n$ and the Lemma follows.

If $head(s)$ is a fully applied combinator, we proceed by induction on $\|t\| + \|s\|$. The proof splits into two depending on whether $head(s_1)$ (and therefore $head(t_1)$) is a fully applied combinator or not.

If $s_1 = \mathcal{C}_{\mathsf{any}}\ \overline{s'_m}$, the semisafe condition ensures that $t_1 = \mathcal{C}_{\mathsf{any}}\ \overline{t'_m}$. Let $t' = F_\infty(t_0)$ and $s' = F_\infty(s_0)$. Then, $\|t_1\| = \|t'\|+1$ and $\|s_1\| = \|s'\|+1$, so $\|t'\| \geq \|s'\|+ n_1$. Since $\|t'\, t_2 \ldots t_n\|+\|s'\, s_2 \ldots s_n\| < \|t\|+\|s\|$ and $semisafe(t'\, t_2 \ldots t_n, s'\, s_2 \ldots s_n)$, the induction hypothesis can be invoked to conclude that $\|t'\, t_2 \ldots t_n\| \geq \|s'\, s_2 \ldots s_n\|+ \sum_{i=1}^{n} n_i$. From this, $\|t\| \geq \|s\| + \sum_{i=1}^{n} n_i$ follows since $\|t\| = 1 + \|t'\, t_2 \ldots t_n\|$ and $\|s\| = 1 + \|s'\, s_2 \ldots s_n\|$.

If $head(s_1)$ is a partially applied combinator, the proof follows by a case analysis of the particular combinator along with its argument number. We provide the proofs for all cases relating to the S-, K- and I-combinators.

Let $s_1 = \mathsf{I}$. Then $t_1 = \mathsf{I}$ and $n_1 = 0$. For terms $t' = F_\infty(t) = t_2 \ldots t_n$ and $s' = F_\infty(s) = s_2 \ldots s_n$, the induction hypothesis gives that $\|t'\| \geq \|s'\| + \sum_{i=2}^{n} n_i$. Then $\|t\| \geq \|s\| + \sum_{i=2}^{n} n_i + 0 = \|s\| + \sum_{i=1}^{n} n_i$.

Let $s_1 = \mathsf{K}$. Then $t_1 = \mathsf{K}$ and $n_1 = 0$. By the fundamental lemma of maximality, $\|t\| = \|t_2\, t_4 \ldots t_n\| + \|t_3\|$ and $\|s\| = \|s_2\, s_4 \ldots s_n\| + \|s_3\|$. The induction hypothesis can be used on the terms $t_2\, t_4 \ldots t_n$ and $s_2\, s_4 \ldots s_n$ and t_3 and s_3 to show $\|t_2\, t_4 \ldots t_n\| \geq \|s_2\, s_4 \ldots s_n\| + n_2 + n_4 + \cdots + n_n$ and $\|t_3\| \geq \|s_3\| + n_3$. From this, it can be concluded that $\|t\| \geq \|s\| + \sum_{i=2}^{n} n_i + 0 = \sum_{i=1}^{n} n_i$.

Let $s_1 = \mathsf{K}\, s'$. Then $t_1 = \mathsf{K}\, t'$. Since $\|t_1\| \geq \|s_1\| + n_1$ and any weak-reductions in t_1 must occur in t' and any weak reductions in s_1 must occur in s', we have that

$\|t'\| \geq \|s'\| + n_1$. By the fundamental lemma of maximality, $\|t\| = \|t'\, t_3 \ldots t_n\| + \|t_2\|$ and $\|s\| = \|s'\, s_3 \ldots s_n\| + \|s_2\|$. The induction hypothesis can be used on the terms $t'\, t_3 \ldots t_n$ and $s'\, s_3 \ldots s_n$ and t_2 and s_2 to show $\|t'\, t_3 \ldots t_n\| \geq \|s'\, s_3 \ldots s_n\| + n_1 + n_3 + \cdots + n_n$ and $\|t_2\| \geq \|s_2\| + n_2$. From this, it can be concluded that $\|t\| \geq \|s\| + \sum_{i=1}^{n} n_i$.

Assume that $s_1 = \mathsf{S}$. Then $t_1 = \mathsf{S}$ and $n_1 = 0$. Let $t' = F_\infty(t) = t_2\, t_4\, (t_3\, t_4)\, t_5 \ldots t_n$ and $s' = F_\infty(s) = s_2\, s_4\, (s_3\, s_4)\, s_5 \ldots s_n$. The induction hypothesis can be utilised on $t_3\, t_4$ and $s_3\, s_4$ to give $\|t_3\, t_4\| \geq \|s_3\, s_4\| + n_3 + n_4$. The induction hypothesis can be used a second time on t' and s' to give $\|t'\| \geq \|s'\| + n_2 + n_4 + n_3 + n_4 + n_5 + \cdots + n_n$. Since $n_2 + n_4 + n_3 + n_4 + n_5 + \cdots + n_n \geq \sum_{i=2}^{n} n_i$, it can be concluded that $\|t\| \geq \|s\| + \sum_{i=2}^{n} n_i + 0 = \sum_{i=1}^{n} n_i$.

Assume that $s_1 = \mathsf{S}\, s'$. Then $t_1 = \mathsf{S}\, t'$ and $\|t'\| \geq \|s'\| + n_1$. Let $t' = F_\infty(t) = t'\, t_3\, (t_2\, t_3)\, t_4 \ldots t_n$ and $s' = F_\infty(s) = s'\, s_3\, (s_2\, s_3)\, s_4 \ldots s_n$. The induction hypothesis can be utilised on $t_2\, t_3$ and $s_2\, s_3$ to give $\|t_2\, t_3\| \geq \|s_2\, s_3\| + n_2 + n_3$. The induction hypothesis can be used a second time on t' and s' to give $\|t'\| \geq \|s'\| + n_1 + n_3 + n_2 + n_3 + n_4 + \cdots + n_n$. Since $n_1 + n_3 + n_2 + n_3 + n_4 + \cdots + n_n \geq \sum_{i=1}^{n} n_i$, it can be concluded that $\|t\| \geq \|s\| + \sum_{i=1}^{n} n_i$.

Assume that $s_1 = \mathsf{S}\, s'_1\, s'_2$. Then $t_1 = \mathsf{S}\, t'_1\, t'_2$. Further, $\|t'_1\| \geq \|s'_1\| + n'_1$ and $\|t'_2\| \geq \|s'_2\| + n'_2$ where $n'_1 + n'_2 = n_1$. Let $t' = F_\infty(t) = t'_1\, t_2\, (t'_2\, t_2)\, t_3 \ldots t_n$ and $s' = F_\infty(s) = s'_1\, s_2\, (s'_2\, s_2)\, s_3 \ldots s_n$. The induction hypothesis can be utilised on $t'_2\, t_2$ and $s'_2\, s_2$ to give $\|t'_2\, t_2\| \geq \|s'_2\, s_2\| + n'_2 + n_2$. The induction hypothesis can be used a second time on t' and s' to give $\|t'\| \geq \|s'\| + n'_1 + n_2 + n'_2 + n_2 + n_3 + \cdots + n_n$. Since $n'_1 + n_2 + n'_2 + n_2 + n_3 + \cdots + n_n \geq \sum_{i=1}^{n} n_i$, it can be concluded that $\|t\| \geq \|s\| + \sum_{i=1}^{n} n_i$.

Lemma 13. *Let t and s be non-ground terms such that $\|t\| \geq \|s\| + m$ for some $m \in \mathbb{N}$ and $safe(t, s)$. Then, for any substitution σ, $\|t\sigma\| \geq \|s\sigma\| + m$ and $safe(t\sigma, s\sigma)$.*

Proof. By induction on $|t| + |s|$. Assume that $head(s)$ is not a fully applied combinator or variable. Let $t = \zeta\ \overline{t_n}$ and $s = \mathsf{f}\ \overline{s_n}$. Without loss of generality, it can be assumed that ζ and f have the same number of arguments since if they did not, the term whose head had fewer arguments could be padded with an arbitrary constant d. Since $\|d\| = \|d\sigma\| = 0$ for all substitutions σ this has no impact on the lemma. For $1 \leq i \leq n$, we have that $\|t_i\| \geq \|s_i\| + n_i$ for $n_i \in \mathbb{I}$ where n_i could possibly be negative and $\sum_{i=1}^{n} n_i = m$.

We proceed to show that for $1 \leq i \leq n$ and all σ, $\|t_i\sigma\| \geq \|s_i\sigma\| + n_i$ and $safe(t_i\sigma, s_i\sigma)$. For some i, if t_i and s_i are ground then it is obviously the case that $\|t_i\sigma\| \geq \|s_i\sigma\|$ and $safe(t_i\sigma, s_i\sigma)$. If s_i is ground and t_i is not, $\|t_i\sigma\| \geq \|t_i\| \geq (\|s_i\| + n_i) = (\|s_i\sigma\| + n_i)$. If s_i is not ground, t_i cannot be ground by $safe(t, s)$. Therefore, the induction hypothesis can be applied to show $\|t_i\sigma\| \geq \|s_i\sigma\| + n_i$ and that $safe(t_i\sigma, s_i\sigma)$. The fact that for $1 \leq i \leq n$, $\|t_i\sigma\| \geq \|s_i\sigma\| + n_i$ and $safe(t_i\sigma, s_i\sigma)$ implies that $\|t\sigma\| = \sum_{i=1}^{n} \|t_i\sigma\| \geq \sum_{i=1}^{n} \|s_i\sigma\| + \sum_{i=1}^{n} n_i = \|s\sigma\| + m$. Further, it also implies $safe(t\sigma, s\sigma)$.

Assume that $head(s)$ is a variable. Then, by the safety condition, $head(t)$ is the same variable, $t = x\ \overline{t_n}$ and $s = x\ \overline{s_n}$. For any substitution σ, $t\sigma$ is of the form $\xi\ \overline{r_{m'}}\ \overline{t_n\sigma}$ and $s\sigma$ is of the form $\xi\ \overline{r_{m'}}\ \overline{s_n\sigma}$. If ξ is a variable or non-fully applied

combinator, the proof proceeds as per the previous case. If ξ is a fully applied combinator we proceed as follows. It is obvious that $safe(r_i, r_i)$ for $1 \leq i \leq m'$. For some i such that $1 \leq i \leq n$, if t_i and s_i are ground, $safe(t_i\sigma, s_i\sigma)$ follows from $safe(t, s)$. Likewise, if only s_i is ground. If s_i is not ground, t_i cannot be ground due to the fact that $safe(t, s)$ and from the induction hypothesis $safe(t_i\sigma, s_i\sigma)$ follows. The fact that for $1 \leq i \leq m'$, $safe(r_i, r_i)$ along with $safe(t_i\sigma, s_i\sigma)$ for $1 \leq i \leq n$ implies $safe(t\sigma, s\sigma)$.

Further, by the induction hypothesis (along with non-ground assumptions as per the previous paragraph), we have that $\|t_i\sigma\| \geq \|s_i\sigma\| + n_i$ for $1 \leq i \leq n$ and $n_1 \ldots n_n \in \mathbb{N}$ such that $\sum_{i=1}^{n} n_i = m$. We also have that $\|r_i\| \geq \|r_i\| + 0$ for $1 \leq i \leq m'$. This along with the fact that $semisafe(t\sigma, s\sigma)$, allows Lemma 12 to be applied to $t\sigma$ and $s\sigma$ to conclude that $\|t\sigma\| \geq \|s\sigma\| + \sum_{i=1}^{n} n_i = \|s\sigma\| + m$.

The final case where $head(t)$ and $head(s)$ are both fully applied combinators is the same as the previous case when ξ is a fully applied combinator without the need to deal with the r_i.

Lemma 14. *For terms t and s such that $var\_cond(t, s)$ holds and $\|t\| \geq \|s\| + n$ for some $n \in \mathbb{N}$, for all substitutions σ, $\|t\sigma\| \geq \|s\sigma\| + n$.*

Proof. If s and t are ground, the theorem is trivial. If s is ground, then $\|t\sigma\| \geq \|t\| \geq \|s\| + n$. If s is not ground, then $var\_cond(t, s)$ implies that t is not ground. Therefore, assume that neither is ground. If $head(s)$ (and therefore $head(t)$ by the variable condition) are fully applied combinators or variables, then $var\_cond(t, s)$ implies $safe(t, s)$ and Lemma 13 can be invoked to prove the lemma. Therefore, assume that both have non-variable, non-fully applied combinator heads.

Let $t = u\langle\!\langle \overline{t_m} \rangle\!\rangle$ and $s = u'\langle\!\langle \overline{s_m} \rangle\!\rangle$ where $\overline{s_m}$ are all the non-ground, top-level, first-order subterms of the form $x\ \overline{args}$ or $\mathcal{C}_{\mathsf{any}}\ \overline{args}$ in s. By the variable condition, we have that there exists a total injective function respecting the given conditions from the s_i to non-ground, top-level, first-order subterms of t of the form $x\ \overline{args}$ or $\mathcal{C}_{\mathsf{any}}\ \overline{args}$. Let $\overline{t_m}$ be the terms related to $\overline{s_m}$ by this function. Without loss of generality, assume that that this function relates s_1 to t_1, s_2 to t_2 and so on. For $1 \leq i \leq m$, $\|t_i\| = \|s_i\| + m_i$ for $m_i \in \mathbb{N}$. This follows from the fact that since t_i and s_i are both non-ground and $safe(t_i, s_i)$, we have $semisafe(t_i, s_i)$ and can therefore invoke Lemma 12.

Let $m' = \|u\langle\langle\rangle\rangle\| - \|u'\langle\langle\rangle\rangle\|$. Note that m' could be negative. By Lemma 11, $\|t\| = \|u\langle\langle\rangle\rangle\| + \sum_{i=1}^{m} \|t_i\|$ and $\|s\| = \|u'\langle\langle\rangle\rangle\| + \sum_{i=1}^{m} \|s_i\|$. Thus, $\|t\| = \|s\| + m' + \sum_{i=1}^{m} m_i$. Therefore, $m' + \sum_{i=1}^{m} m_i \geq n$. Lemma 13 can be used to show that for all i, $\|t_i\sigma\| \geq \|s_i\sigma\| + m_i$. Because $u'\langle\langle\rangle\rangle$ is ground, it follows $\|u\sigma\langle\langle\rangle\rangle\| - \|u'\sigma\langle\langle\rangle\rangle\| \geq m'$. To conclude the proof:

$$\begin{aligned} \|t\sigma\| &= \|u\sigma\langle\!\langle \overline{t_m\sigma} \rangle\!\rangle\| \\ &= \|u\sigma\langle\langle\rangle\rangle\| + \sum_{i=1}^{m} \|t_i\sigma\| \end{aligned}$$

$$
\begin{aligned}
&\geq && \|u'\sigma\langle\langle\rangle\rangle\| + \sum_{i=1}^{m} \|s_i\sigma\| + m' + \sum_{i=1}^{m} m_i \\
&\geq && \|u'\sigma\langle\langle\rangle\rangle\| + \sum_{i=1}^{m} \|s_i\sigma\| + n \\
&= && \|s\sigma\| + n
\end{aligned}
$$

Lemma 15. *For terms t and s such that $var\_cond(t, s)$ holds and for all substitutions σ, $var\_cond(t\sigma, s\sigma)$.*

Proof. Let $t = u\langle\langle \overline{t_m} \rangle\rangle$ and $s = u'\langle\langle \overline{s_m} \rangle\rangle$ where $\overline{s_m}$ are all the non-ground, top-level, first-order subterms of the form $x\ \overline{args}$ or $\mathcal{C}_{\mathsf{any}}\ \overline{args}$ in s. By the variable condition, we have that there exists a total injective function respecting the given conditions from the s_i to non-ground, top-level, first-order subterms of t of the form $x\ \overline{args}$ or $\mathcal{C}_{\mathsf{any}}\ \overline{args}$. Let $\overline{t_m}$ be the terms related to $\overline{s_m}$ by this function. Without loss of generality, assume that that this function relates s_1 to t_1, s_2 to t_2 and so on. By the definition of the variable condition, we have that u' must be ground. This implies that any non-ground subterms of $s\sigma$ must be subterms of some $s_i\sigma$ for $1 \leq i \leq m$.

Assume that for some i and $p \in pos(s_i\sigma)$, $s_i\sigma|_p$ is a non-ground, top-level, first-order subterm of the form $x\ \overline{args}$ or $\mathcal{C}_{\mathsf{any}}\ \overline{args}$. We show that $t_i\sigma|_p$ is a non-ground, top-level, first-order subterm of $t\sigma$ and $safe(t_i|_p, s_i|_p)$. This implies the existence of a total, injective function from the multiset of non-ground, top-level first-order subterms in $s\sigma$ to the like multiset of $t\sigma$ in turn proving $var\_cond(t\sigma, s\sigma)$.

From Lemma 13, it can be shown that for $1 \leq i \leq m$, $safe(t_i\sigma, s_i\sigma)$. By the subterm property of safety, this implies that $safe(t_i\sigma|_p, s_i\sigma|_p)$.

Assume that $t_i\sigma|_p$ is ground. Since $s_i\sigma|_p$ is non-ground, this implies that there exists a position p' such that $s_i\sigma|_{pp'}$ is a variable whilst either pp' is not defined in $t_i\sigma$ or it is not a variable position. Both contradict the fact that $safe(t_i\sigma, s_i\sigma)$.

Assume that $t_i\sigma|_p$ is not top-level. That is, $t_i\sigma|_p$ occurs beneath a variable or fully applied combinator. If this variable or combinator was introduced by σ, then there must exist a term in t_i at some prefix p' of p of the form $x\ \overline{args_n}$ such that $x\sigma$ is of the form $\zeta\ \overline{r}$ where ζ is a variable or fully applied combinator. By $safe(t_i, s_i)$, it can be concluded that $s_i|_{p'} = x\ \overline{args'_n}$. Since $head(x\sigma)$ is a variable or fully applied combiantor, this contradicts the fact that $s_i\sigma|_p$ is top-level in $s\sigma$. If this variable or combinator was not introduced by σ, then it must have occurred in t_i. But then by safety, it must have occurred in s_i at the same position again leading to a contradiction.

Finally, $s_i\sigma|_p$ being a first-order subterm implies that $t_i\sigma|_p$ must be first-order completing the proof.

Lemma 16. *Let t be a polymorphic term and σ be a substitution. We define a new substitution ρ such that the domain of ρ is $dom(\sigma)$. Define $y\rho = [\![y\sigma]\!]$. For all terms t, $[\![t\sigma]\!] = [\![t]\!]\rho$.*

Proof. Via a straightforward induction on t.

Theorem 7 (Stability under Substitution). *If $s >_{\mathsf{ski}} t$ then $s\sigma >_{\mathsf{ski}} t\sigma$ for all substitutions σ that respect the ghd mapping.*

Proof. Let $s' = [\![s]\!]$ and $t' = [\![t]\!]$. Let ρ be defined as per Lemma 16. First, we show that if R1 was used to derive $s >_{\mathsf{ski}} t$ and thus $\|s'\| > \|t'\|$ then $\|s'\rho\| > \|t'\rho\|$ and thus $s\sigma >_{\mathsf{ski}} t\sigma$ because $[\![s\sigma]\!] = s'\rho$ and $[\![t\sigma]\!] = t'\rho$.

From Lemma 15 and $var\_cond(s', t')$, $var\_cond(s'\rho, t'\rho)$ holds. Furthermore, if $\|s'\| > \|t'\|$, then by Lemma 14 $\|s'\rho\| > \|t'\rho\|$ and $s\sigma >_{\mathsf{ski}} t\sigma$ by an application of R1.

On the other hand, if $\|s'\| = \|t'\|$, then R2 was used to derive $s >_{\mathsf{ski}} t$. By Lemma 14 $\|s'\rho\| \geq \|t'\rho\|$. If $\|s'\rho\| > \|t'\rho\|$, then this is the same as the former case. Otherwise $\|s'\rho\| = \|t'\rho\|$ and $s'\rho$ and $t\rho$ are compared using R2. From the stability under substitution of $>_{\mathsf{hb}}$, $s'\rho >_{\mathsf{hb}} t'\rho$ follows and $s\sigma >_{\mathsf{ski}} t\sigma$ can be concluded.

Theorem 8 (Well-foundedness). *There exists no infinite descending chain of comparisons $s_1 >_{\mathsf{ski}} s_2 >_{\mathsf{ski}} s_3 \cdots$.*

Proof. Assume that such a chain exists. For each $s_i >_{\mathsf{ski}} s_{i+1}$ derived by R1, we have that $\|s_i\| > \|s_{i+1}\|$. For each $s_i >_{\mathsf{ski}} s_{i+1}$ derived by R2, we have that $\|s_i\| = \|s_{i+1}\|$. Therefore the number of times $s_i >_{\mathsf{ski}} s_{i+1}$ by R1 in the infinite chain must be finite and there must exist some m such that for all $n > m$, $s_n >_{\mathsf{ski}} s_{n+1}$ by R2. Therefore, there exists an infinite sequence of $>_{\mathsf{hb}}$ comparisons $[\![s_m]\!] >_{\mathsf{hb}} [\![s_{m+1}]\!] >_{\mathsf{hb}} [\![s_{m+2}]\!] \cdots$. This contradicts the well-foundedness of $>_{\mathsf{hb}}$.

Theorem 9 (Coincidence with First-Order KBO). *Let $>_{\mathsf{fo}}$ be the first-oder KBO as described by Becker et al. in [2]. Assume that $>_{\mathsf{ski}}$ and $>_{\mathsf{fo}}$ are parameterised by the same precedence $\succ$ and that $>_{\mathsf{fo}}$ always compares tuples using the lexicographic extension operator. Then $>_{\mathsf{ski}}$ and $>_{\mathsf{fo}}$ always agree on first-order terms.*

Proof. Let $t' = [\![t]\!]$ and $s' = [\![s]\!]$. Since s and t are first-order, $\|s'\| = 0$ and $\|t'\| = 0$. Thus, s' and t' will always be compared by $>_{\mathsf{hb}}$. Since $>_{\mathsf{hb}}$ coincides with $>_{\mathsf{fo}}$ on first-order terms, so does $>_{\mathsf{ski}}$.

5 Examples

To give a flavour of how the ordering behaves, we provide a number of examples.

Example 1. Consider the terms (ignoring type arguments) $t = \mathsf{S}\,(\mathsf{K}\,\mathsf{a})\,\mathsf{b}\,\mathsf{c}$ and $s = \mathsf{f}\,\mathsf{c}\,\mathsf{e}$. From the definition of the translation $[\![]\!]$, we have that $[\![t]\!] = \mathsf{S}\,(\mathsf{K}\,\mathsf{a})\,\mathsf{b}\,\mathsf{c}$ and $[\![s]\!] = \mathsf{f}\,\mathsf{c}\,\mathsf{e}$. Since $\|\mathsf{S}\,(\mathsf{K}\,\mathsf{a})\,\mathsf{b}\,\mathsf{c}\| = 2$ and $\|\mathsf{f}\,\mathsf{c}\,\mathsf{e}\| = 0$, we have that $t >_{\mathsf{ski}} s$.

Example 2. Consider the terms $t = \mathsf{f}\,(\mathsf{g}\,\mathsf{b})\,\mathsf{e}\,\mathsf{d}$ and $s = \mathsf{I}\,\mathsf{a}$. Here $s >_{\mathsf{ski}} t$ despite the fact that s is syntactically smaller than t because s has a maximum reduction of 1 as opposed to 0 of t.

Example 3. Consider terms $t = \mathsf{f}\,(\mathsf{I}\,\mathsf{d})\,(\mathsf{S}\,x\,\mathsf{a}\,\mathsf{b})$ and $s = \mathsf{g}\,(\mathsf{S}\,x\,(\mathsf{h}\,\mathsf{d})\,\mathsf{b})$. The two terms are comparable as the variable condition relates subterm $\mathsf{S}\,x\,(\mathsf{h}\,\mathsf{d})\,\mathsf{b}$ in s to subterm $\mathsf{S}\,x\,\mathsf{a}\,\mathsf{b}$ in t. The unsafe combinator S and variable x are in the same position in each subterm. As $\|t\| > \|s\|$, $t >_{\mathsf{ski}} s$.

Example 4. Consider terms $t = \mathsf{f}\,(\mathbf{I}\,\mathsf{d})\,(\mathbf{S}\,x\,\mathsf{a}\,y)$ and $s = \mathsf{g}\,(\mathbf{S}\,x\,(\mathsf{h}\,y)\,\mathsf{b})$. This is very similar to the previous example, but in this case the terms are incomparable. Let s' be a name for the subterm $(\mathbf{S}\,x\,(\mathsf{h}\,y)\,\mathsf{b})$ in s and t' a name for the subterm $(\mathbf{S}\,x\,\mathsf{a}\,y)$. The variable y occurs in different positions in s' and t'. Therefore, s' cannot be related to t by the variable condition and the two terms are incomparable.

Example 5. Consider terms $t = \mathsf{f}\,(x\,(\mathsf{g}\,(\mathbf{K}\,\mathbf{I}\,\mathsf{a}\,\mathsf{b})))$ and $s = \mathsf{h}\,(\mathbf{I}\,\mathsf{a})\,(x\,\mathsf{c})$. The variable condition holds between t and s by relating $(x\,(\mathsf{g}\,(\mathbf{K}\,\mathbf{I}\,\mathsf{a}\,\mathsf{b})))$ to $(x\,\mathsf{c})$. The combinator $\mathbf{I}$ in s is not unsafe and therefore does not need to be related to a combinator in t.

Since $\|t\| = 2 > \|s\| = 1$, $t >_{\mathsf{ski}} s$. Intuitively, this is safe because a substitution for x in t can duplicate $(\mathsf{g}\,(\mathbf{K}\,\mathbf{I}\,\mathsf{a}\,\mathsf{b}))$ whose maximum reduction length is 2 whilst a substitution for x in s can only duplicate c whose maximum reduction length is 0.

6 Extending to β-Reduction

The ordering that has been explored above orients all ground instances of combinator equations left-to-right and as a result has the property that for ground terms t_1 and t_2 if $t_1 \longrightarrow_w^+ t_2$ then $t_1 >_{\mathsf{ski}} t_2$. It would be useful if an ordering with a similar property with respect to β-reduction could be developed and this is what is explored in this section. We first define some terminology that will be used throughout the section, then present an ordering analogous to the $>_{\mathsf{ski}}$ ordering.

Polymorphic types and type declarations are as defined previously. Below the set of *raw λ-terms* is defined.

$$\textbf{Terms} \quad \mathcal{T} ::= x \mid \mathsf{f}\langle\,\overline{\tau_n}\,\rangle \mid \lambda x.t \mid t_{1\tau_1\to\tau_2}t_{2\tau_1}$$
$$\text{where } x \in \mathcal{V},\, t, t_1, t_2 \in \mathcal{T},\, \mathsf{f} \in \Sigma,\, \mathsf{f} : \Pi\,\overline{\alpha_n}\,.\sigma \text{ and } \overline{\tau_n} \text{ are types}$$

If x is of type τ and t is of type σ then the type of $\lambda x.t$ is $\tau \to \sigma$. The type of the term $\mathsf{f}\langle\,\overline{\tau_n}\,\rangle$ is $\sigma\{\,\overline{\alpha_n} \to \overline{\tau_n}\,\}$. Variables that are bound by the λ binder are known as *bound* variables and all other variables are *free* variables. A term that contains no type variables or free term variables is known as a *ground* term. The consistent renaming of a bound variable throughout a term is known as *α-renaming*. For example the term $\lambda x.\mathsf{f}\,x$ can be α-renamed to $\lambda y.\mathsf{f}\,y$. Raw λ-terms can be partitioned into equivalence classes modulo α-renaming. These equivalence classes are known as λ-terms.

Positions over λ-terms are defined similarly to positions over combinatory terms with the added definition $pos(\lambda x.t) = \{\epsilon\} \cup \{1.p \mid p \in pos(t)\}$. First-order subterms are defined exactly as before. However, stable subterms require redefining. Again, stable subterms are a subset of first-order subterms. By definition, if $p.2 \in pos(t)$ for some term t, then the subterm at $p.2$ is first-order. The subterm is *stable* if $head(t|_p)$ is not a λ-expression or $head(t|_p) = \lambda\,\overline{y_n}\,.t'$, $head(t')$ is not a λ-expression or bound variable and $t|_{p.2}$ is not amongst the first n arguments of $head(t|_p)$.

Beta-reduction is defined on λ-terms as follows. A term of the form $(\lambda x.t)\,t'$ β-reduces to $t\{x \to t'\}$ where the substitution is assumed to be capture avoiding by the α-renaming of t where necessary. If term t β-reduces to t', this is symbolised $t \longrightarrow_\beta t'$.

By an overload of notation, $\|t\|$ is used to denote the length of the longest β-reduction from t. For typed λ-terms, β-reduction is terminating and confluent and a maximal strategy is known [22].

6.1 Beta-compatible Ordering

The ordering first compares λ-terms by the length of the longest β-reduction and then by the graceful higher-order KBO. We overload the translation $[\![]\!]$ to be a translation from λ-terms to untyped terms. It replaces λ-binders by a unary function symbol lam and replaces bound variables by the correct De Bruijn index. The use of De Bruijn indices ensures that $[\![]\!]$ remains an injective function. The definition of type-1 subterms requires updating as well. The translation of types is as given previously. The extended term translation follows the definition.

Definition 7 (Type-1 term). *Consider a term t of the form $(\lambda\, \overline{y_n}\, .t')\ \overline{t_n}$ or $x\ \overline{t_n}$. If there exists a position p such that $t|_p$ is a free variable, then t is a type-1 term.*

$$[\![t]\!] = \begin{cases} x & t = x \text{ and } x \text{ is free} \\ \mathsf{db}_i & t \text{ is a bound variable with } i \text{ binders} \\ & \text{between this occurrence and its binder} \\ x_t & t \text{ is a type-1 term} \\ \mathsf{lam}\ [\![t']\!] & \text{if } t = \lambda x.t' \\ \mathsf{f}\ [\![\,\overline{\tau_n}\,]\!] & \text{if } t = \mathsf{f}\langle\,\overline{\tau_n}\,\rangle \\ [\![t_1]\!][\![t_2]\!] & \text{if } t = t_1 t_2 \end{cases}$$

Then for λ-terms, t and s, Let $t' = [\![t]\!]$ and $s' = [\![s]\!]$. We have $t >_{\beta\mathsf{kbo}} s$ if:

R1 $\|t'\| > \|s'\|$ or,
R2 $\|t'\| = \|s'\|$ and $[\![t']\!] >_{\mathsf{hb}} [\![s']\!]$

The definition of β-reduction on untyped terms is the obvious one. Most of the proofs follow almost exactly as in the combinatory case with the added complication of having to deal with terms of the form $\lambda x.t$ when doing a case analysis on terms. We therefore do not repeat them.

7 Conclusion and Discussion

We have presented an ordering that orients all ground instances of **S**, **C**, **B**, **K** and **I** axioms left-to-right. The ordering enjoys many other useful properties such as stability under substitution, compatibility with stable contexts, ground totality and transitivity. In as yet unpublished work, we have used this ordering to parameterise a complete superposition calculus for HOL [5]. Lack of full compatibility with context has not been an obstacle. In the standard first-order proof of the completeness of superposition, compatibility with contexts is used in model construction to rule out the need for superpostion

inferences beneath variables [18]. Thus, by utilising $>_{\mathsf{ski}}$, some superposition is required beneath variables. However, because terms with functional heads are compatible with all contexts, such inference are quite restricted.

The $>_{\mathsf{ski}}$ ordering presented here is able to compare non-ground terms that cannot be compared by any ordering used to parameterise Bentkamp et al.'s lambda superposition calculus [3]. They define terms to be β-equivalence classes. Non-ground terms are compared using a quasiorder, $\succsim$, such that $t \succsim s$ iff for all grounding substitutions θ, $t\theta \succeq s\theta$. Consider terms $t = x\,\mathsf{a}\,\mathsf{b}$ and $s = x\,\mathsf{b}\,\mathsf{a}$ and grounding substitutions $\theta_1 = x \to \lambda x\,y\,.\mathsf{f}\,y\,x$ and $\theta_2 = x \to \lambda x\,y\,.\mathsf{f}\,x\,y$. By ground totality of $\succ$ it must be the case that either $\mathsf{f}\,\mathsf{a}\,\mathsf{b} \succ \mathsf{f}\,\mathsf{b}\,\mathsf{a}$ or $\mathsf{f}\,\mathsf{b}\,\mathsf{a} \succ \mathsf{f}\,\mathsf{a}\,\mathsf{b}$. Without loss of generality assume the first. Then, neither $t \succsim s$ nor $s \succsim t$ since $t\theta_1 = \mathsf{f}\,\mathsf{b}\,\mathsf{a} \prec \mathsf{f}\,\mathsf{a}\,\mathsf{b} = s\theta_1$ and $t\theta_2 = \mathsf{f}\,\mathsf{a}\,\mathsf{b} \succ \mathsf{f}\,\mathsf{b}\,\mathsf{a} = s\theta_2$.

The $>_{\mathsf{ski}}$ ordering (or the $>_{\beta\mathsf{kbo}}$ ordering) allows weak reduction (or β-reduction) to be treated as part of the superposition calculus. This allows terms t and t' such that $t \longrightarrow_w^+ t'$ (or $t \longrightarrow_\beta^+ t'$) to be considered separate terms resulting in terms such as t and s given above being comparable. Since $\|t\| = \|s\|$, t and s are compared using $>_{\mathsf{hb}}$ with stability under substitution ensured by the stability under substitution of $>_{\mathsf{hb}}$.

Many of the definitions that have been provided here are conservative and can be tightened to allow the comparison of a far larger class of non-ground terms without losing stability under substitution. In further work, we hope to thoroughly explore such refinements.

The definition of a stable subterm can be further sharpened to allow compatibility with a greater number of contexts. We provide the intuitive idea. Consider the context $\mathsf{K}\,(\mathsf{S}\,[])\,s$. This is not a stable context because the innermost head symbol to the hole that is not a partially applied combinator is the fully applied K. However, for terms t and t' such that $\|t\| > \|t'\|$, we must have that $\|\mathsf{K}\,(\mathsf{S}\,t)\,s\| > \|\mathsf{K}\,(\mathsf{S}\,t')\,s\|$ because the terms t and t' can never become applied on a longest reduction path. This suggests the following improvement to the definition of instability.

Definition 8 (Stable Subterm). *Let* $\mathsf{LPP}(t, p)$ *be a partial function that takes a term* t*, a position* p *and returns the longest prefix* p' *of* p *such that* $head(t|_{p'})$ *is not a partially applied combinator* ***or a*** K ***applied to two arguments or an*** I ***applied to a single argument*** *if such a position exists. For a position* $p \in pos(t)$*,* p *is a stable position in* t *if* $\mathsf{LPP}(t, p)$ *is not defined or* $head(t|_{\mathsf{LPP}(t,p)})$ *is not a combinator. A* stable subterm *is a subterm occurring at a stable position.*

In future work, we plan to investigate improvements such as the above. However, we feel that the ordering in the form presented here is a strong base for starting work on a complete superposition calculus for HOL based on the combinatory calculus.

Acknowledgements Thanks to Jasmin Blanchette, Alexander Bentkamp and Petar Vukmirović for many discussions on aspects of this research. We would also like to thank reviewers of this paper, whose comments have done much to shape this paper. The first author thanks the family of James Elson for funding his research.

References

1. Barendregt, H.P.: The Lambda Calculus: Its Syntax and Semantics. Elsevier Science Publishers B.V., Amsterdam, Netherlands, 2nd edn. (1984)
2. Becker, H., Blanchette, J.C., Waldmann, U., Wand, D.: A transfinite Knuth–Bendix order for lambda-free higher-order terms. In: de Moura, L. (ed.) CADE-26. LNCS, vol. 10395, pp. 432–453. Springer (2017)
3. Bentkamp, A., Blanchette, J.C., Tourret, S., Vukmirović, P., Waldmann, U.: Superposition with lambdas. In: Fontaine, P. (ed.) CADE-27. LCNS, vol. 11716, pp. 55–73. Springer (2019)
4. Benzmüller, C., Sultana, N., Paulson, L.C., Theiß, F.: The higher-order prover Leo-II. Journal of Automated Reasoning **55**(4), 389–404 (2015). https://doi.org/10.1007/s10817-015-9348-y
5. Bhayat, A., Reger, G.: A combinator-based superposition calculus for higher-order logic (technical report). Technical report, University of Mancester (2020), https://github.com/vprover/vampire_publications/blob/master/paper_drafts/comb_sup_report.pdf
6. Blanchette, J.C., Waldmann, U., Wand, D.: A lambda-free higher-order recursive path order. In: Esparza, J., Murawski, A.S. (eds.) FoSSaCS 2017. LNCS, vol. 10203, pp. 461–479. Springer (2017)
7. Blanqui, F., Jouannaud, J.P., Rubio, A.: The computability path ordering: The end of a quest. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **5213 LNCS**, 1–14 (2008). https://doi.org/10.1007/978-3-540-87531-41
8. Bofill, M., Godoy, G., Nieuwenhuis, R., Rubio, A.: Paramodulation with non-monotonic orderings. Proceedings - Symposium on Logic in Computer Science (08 1999)
9. Brown, C.E.: Satallax: An automatic higher-order prover. In: Gramlich, B., Miller, D., Sattler, U. (eds.) Automated Reasoning. pp. 111–117. Springer (2012)
10. Czajka, Ł., Kaliszyk, C.: Hammer for Coq: Automation for dependent type theory. Journal of Automated Reasoning **61**(1), 423–453 (Jun 2018)
11. Graf, P.: Substitution tree indexing, pp. 117–131. Springer Berlin Heidelberg, Berlin, Heidelberg (1995). https://doi.org/10.1007/3-540-59200-8_52, http://dx.doi.org/10.1007/3-540-59200-8_52
12. Hindley, J.R., Seldin, J.P.: Lambda-Calculus and Combinators: An Introduction. Cambridge University Press, New York, NY, USA, 2nd edn. (2008)
13. Jouannaud, J.P., Rubio, A.: Polymorphic Higher-Order Recursive Path Orderings. Journal of the ACM **54**(1) (2007). https://doi.org/10.1145/1206035.1206037
14. Kerber, M.: How to prove higher order theorems in first order logic. In: IJCAI. pp. 137–142 (01 1991)
15. Kop, C., van Raamsdonk, F.: A higher-order iterative path ordering. In: Logic for Programming, Artificial Intelligence, and Reasoning. vol. 5330 LNCS, pp. 697–711. Springer (2008). https://doi.org/10.1007/978-3-540-89439-1_48
16. Lindblad, F.: https://github.com/frelindb/agsyHOL, accessed: 25-09-2019
17. Meng, J., Paulson, L.C.: Translating higher-order clauses to first-order clauses. Journal of Automated Reasoning **40**(1), 35–60 (Jan 2008). https://doi.org/10.1007/s10817-007-9085-y, https://doi.org/10.1007/s10817-007-9085-y
18. Nieuwenhuis, R., Rubio, A.: Handbook of Automated Reasoning, vol. 1, chap. Paramodulation-Based Theorem Proving, pp. 371–443. Elsevier Press and MIT press (08 2001). https://doi.org/10.1016/B978-044450813-3/50009-6
19. Sekar, R., Ramakrishnan, I., Voronkov, A.: Term indexing. In: Robinson, A., Voronkov, A. (eds.) Handbook of Automated Reasoning, vol. II, chap. 26, pp. 1853–1964. Elsevier Science (2001)

20. Steen, A.: Extensional Paramodulation for Higher-Order Logic and its Effective Implementation Leo-III. Ph.D. thesis, Freie Universität Berlin (2018)
21. Sultana, N., Blanchette, J.C., Paulson, L.C.: Leo-II and Satallax on the Sledgehammer test bench. Journal of Applied Logic **11**(1), 91 – 102 (2013). https://doi.org/https://doi.org/10.1016/j.jal.2012.12.002
22. van Raamsdonk, F., Severi, P., Sørensen, M., Xi, H.: Perpetual reductions in lambda calculus. Information and Computation **149**(2), 173–225 (1999). https://doi.org/10.1006/inco.1998.2750