

# Formalization of Knuth–Bendix Orders for Lambda-Free Higher-Order Terms

Heiko Becker, Jasmin Christian Blanchette, Uwe Waldmann, and Daniel Wand

August 16, 2018

## Abstract

This Isabelle/HOL formalization defines Knuth–Bendix orders for higher-order terms without  $\lambda$ -abstraction and proves many useful properties about them. The main order fully coincides with the standard transfinite KBO with subterm coefficients on first-order terms. It appears promising as the basis of a higher-order superposition calculus.

## Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>2</b>  |
| <b>2</b> | <b>Utilities for Knuth–Bendix Orders for Lambda-Free Higher-Order Terms</b>                                     | <b>2</b>  |
| <b>3</b> | <b>The Applicative Knuth–Bendix Order for Lambda-Free Higher-Order Terms</b>                                    | <b>3</b>  |
| <b>4</b> | <b>The Graceful Standard Knuth–Bendix Order for Lambda-Free Higher-Order Terms</b>                              | <b>4</b>  |
| 4.1      | Setup   | 4         |
| 4.2      | Weights   | 4         |
| 4.3      | Inductive Definitions   | 5         |
| 4.4      | Irreflexivity   | 6         |
| 4.5      | Transitivity  | 6         |
| 4.6      | Subterm Property  | 6         |
| 4.7      | Compatibility with Functions  | 6         |
| 4.8      | Compatibility with Arguments  | 7         |
| 4.9      | Stability under Substitution  | 7         |
| 4.10     | Totality on Ground Terms  | 7         |
| 4.11     | Well-foundedness  | 7         |
| <b>5</b> | <b>The Graceful Basic Knuth–Bendix Order for Lambda-Free Higher-Order Terms</b>                                 | <b>8</b>  |
| <b>6</b> | <b>The Graceful Transfinite Knuth–Bendix Order with Subterm Coefficients for Lambda-Free Higher-Order Terms</b> | <b>9</b>  |
| 6.1      | Setup   | 9         |
| 6.2      | Weights and Subterm Coefficients  | 10        |
| 6.3      | Inductive Definitions   | 14        |
| 6.4      | Irreflexivity   | 15        |
| 6.5      | Transitivity  | 15        |
| 6.6      | Subterm Property  | 15        |
| 6.7      | Compatibility with Functions  | 15        |
| 6.8      | Compatibility with Arguments  | 16        |
| 6.9      | Stability under Substitution  | 16        |
| 6.10     | Totality on Ground Terms  | 16        |
| 6.11     | Well-foundedness  | 16        |
| <b>7</b> | <b>Knuth–Bendix Orders for Lambda-Free Higher-Order Terms</b>   | <b>17</b> |

# 1 Introduction

This Isabelle/HOL formalization defines Knuth–Bendix orders for higher-order terms without  $\lambda$ -abstraction and proves many useful properties about them. The main order fully coincides with the standard transfinite KBO with subterm coefficients on first-order terms. It appears promising as the basis of a higher-order superposition calculus.

We refer to our CADE-26 paper for details.<sup>1</sup>

## 2 Utilities for Knuth–Bendix Orders for Lambda-Free Higher-Order Terms

```
theory Lambda_Free_KBO_Util
imports Lambda_Free_RPOs.Lambda_Free_Term Lambda_Free_RPOs.Extension_Orders Polynomials.Polynomials
begin
```

```
locale kbo_basic_basis = gt_sym (>s)
  for gt_sym :: 's ⇒ 's ⇒ bool (infix >s 50) +
  fixes
    wt_sym :: 's ⇒ nat and
    ε :: nat and
    ground_heads_var :: 'v ⇒ 's set and
    extf :: 's ⇒ (('s, 'v) tm ⇒ ('s, 'v) tm ⇒ bool) ⇒ ('s, 'v) tm list ⇒ ('s, 'v) tm list ⇒
      bool
  assumes
    ε_gt_0: ε > 0 and
    wt_sym_ge_ε: wt_sym f ≥ ε and
    ground_heads_var_nonempty: ground_heads_var x ≠ {} and
    extf_ext_irrefl_before_trans: ext_irrefl_before_trans (extf f) and
    extf_ext_compat_list_strong: ext_compat_list_strong (extf f) and
    extf_ext_hd_or_tl: ext_hd_or_tl (extf f)
begin
```

```
lemma wt_sym_gt_0: wt_sym f > 0
  ⟨proof⟩
```

end

```
locale kbo_std_basis = ground_heads (>s) arity_sym arity_var
  for
    gt_sym :: 's ⇒ 's ⇒ bool (infix >s 50) and
    arity_sym :: 's ⇒ enat and
    arity_var :: 'v ⇒ enat +
  fixes
    wt_sym :: 's ⇒ 'n::{ord,semiring_1} and
    ε :: nat and
    δ :: nat and
    extf :: 's ⇒ (('s, 'v) tm ⇒ ('s, 'v) tm ⇒ bool) ⇒ ('s, 'v) tm list ⇒ ('s, 'v) tm list ⇒
      bool
  assumes
    ε_gt_0: ε > 0 and
    δ_le_ε: δ ≤ ε and
    arity_hd_ne_infinity_if_δ_gt_0: δ > 0 ⇒ arity_hd ζ ≠ ∞ and
    wt_sym_ge: wt_sym f ≥ of_nat (ε - the_enat (of_nat δ * arity_sym f)) and
    unary_wt_sym_0_gt: arity_sym f = 1 ⇒ wt_sym f = 0 ⇒ f >s g ∨ g = f and
    unary_wt_sym_0_imp_δ_eq_ε: arity_sym f = 1 ⇒ wt_sym f = 0 ⇒ δ = ε and
    extf_ext_irrefl_before_trans: ext_irrefl_before_trans (extf f) and
    extf_ext_compat_list_strong: ext_compat_list_strong (extf f) and
    extf_ext_hd_or_tl: ext_hd_or_tl (extf f) and
    extf_ext_snoc_if_δ_eq_ε: δ = ε ⇒ ext_snoc (extf f)
begin
```

<sup>1</sup>[https://www21.in.tum.de/~blanchet/lambda\\_free\\_kbo\\_conf.pdf](https://www21.in.tum.de/~blanchet/lambda_free_kbo_conf.pdf)

**lemma** *arity\_sym\_ne\_infinity\_if\_delta\_gt\_0*:  $\delta > 0 \implies \text{arity\_sym } f \neq \infty$   
 ⟨proof⟩

**lemma** *arity\_var\_ne\_infinity\_if\_delta\_gt\_0*:  $\delta > 0 \implies \text{arity\_var } x \neq \infty$   
 ⟨proof⟩

**lemma** *arity\_ne\_infinity\_if\_delta\_gt\_0*:  $\delta > 0 \implies \text{arity } s \neq \infty$   
 ⟨proof⟩

**lemma** *extf\_ext\_irrefl*: *ext\_irrefl* (*extf* *f*)  
 ⟨proof⟩

**lemma** *extf\_ext*: *ext* (*extf* *f*)  
 ⟨proof⟩

**lemma**  
*extf\_ext\_compat\_cons*: *ext\_compat\_cons* (*extf* *f*) **and**  
*extf\_ext\_compat\_snoc*: *ext\_compat\_snoc* (*extf* *f*) **and**  
*extf\_ext\_singleton*: *ext\_singleton* (*extf* *f*)  
 ⟨proof⟩

**lemma** *extf\_ext\_compat\_list*: *ext\_compat\_list* (*extf* *f*)  
 ⟨proof⟩

**lemma** *extf\_ext\_wf\_bounded*: *ext\_wf\_bounded* (*extf* *f*)  
 ⟨proof⟩

**lemmas** *extf\_mono\_strong* = *ext\_mono\_strong*[*OF* *extf\_ext*]

**lemmas** *extf\_mono* = *ext\_mono*[*OF* *extf\_ext*, *mono*]

**lemmas** *extf\_map* = *ext\_map*[*OF* *extf\_ext*]

**lemmas** *extf\_irrefl* = *ext\_irrefl.irrefl*[*OF* *extf\_ext\_irrefl*]

**lemmas** *extf\_trans\_from\_irrefl* =

*ext\_irrefl\_before\_trans.trans\_from\_irrefl*[*OF* *extf\_ext\_irrefl\_before\_trans*]

**lemmas** *extf\_compat\_cons* = *ext\_compat\_cons.compat\_cons*[*OF* *extf\_ext\_compat\_cons*]

**lemmas** *extf\_compat\_append\_left* = *ext\_compat\_cons.compat\_append\_left*[*OF* *extf\_ext\_compat\_cons*]

**lemmas** *extf\_compat\_append\_right* = *ext\_compat\_snoc.compat\_append\_right*[*OF* *extf\_ext\_compat\_snoc*]

**lemmas** *extf\_compat\_list* = *ext\_compat\_list.compat\_list*[*OF* *extf\_ext\_compat\_list*]

**lemmas** *extf\_singleton* = *ext\_singleton.singleton*[*OF* *extf\_ext\_singleton*]

**lemmas** *extf\_wf\_bounded* = *ext\_wf\_bounded.wf\_bounded*[*OF* *extf\_ext\_wf\_bounded*]

**lemmas** *extf\_snoc\_if\_delta\_eq\_epsilon* = *ext\_snoc.snoc*[*OF* *extf\_ext\_snoc\_if\_delta\_eq\_epsilon*]

**lemma** *extf\_singleton\_nil\_if\_delta\_eq\_epsilon*:  $\delta = \epsilon \implies \text{extf } f \text{ gt } [s] []$   
 ⟨proof⟩

**end**

**sublocale** *kbo\_basic\_basis* < *kbo\_std\_basis* \_ \_  $\lambda$  .  $\infty$   $\lambda$  .  $\infty$  \_ \_ 0  
 ⟨proof⟩

**end**

### 3 The Applicative Knuth–Bendix Order for Lambda-Free Higher-Order Terms

**theory** *Lambda\_Free\_KBO\_App*  
**imports** *Lambda\_Free\_KBO\_Util*  
**abbrevs**  $>t = >_t$   
**and**  $\geq t = \geq_t$   
**begin**

This theory defines the applicative Knuth–Bendix order, a variant of KBO for  $\lambda$ -free higher-order terms. It

corresponds to the order obtained by applying the standard first-order KBO on the applicative encoding of higher-order terms and assigning the lowest precedence to the application symbol.

```

locale kbo_app = gt_sym (>s)
  for gt_sym :: 's ⇒ 's ⇒ bool (infix >s 50) +
  fixes
    wt_sym :: 's ⇒ nat and
    ε :: nat and
    ext :: (('s, 'v) tm ⇒ ('s, 'v) tm ⇒ bool) ⇒ ('s, 'v) tm list ⇒ ('s, 'v) tm list ⇒ bool
  assumes
    ε_gt_0: ε > 0 and
    wt_sym_ge_ε: wt_sym f ≥ ε and
    ext_ext_irrefl_before_trans: ext_irrefl_before_trans ext and
    ext_ext_compat_list: ext_compat_list ext and
    ext_ext_hd_or_tl: ext_hd_or_tl ext
begin

lemma ext_mono[mono]: gt ≤ gt' ⇒ ext gt ≤ ext gt'
  (proof)

fun wt :: ('s, 'v) tm ⇒ nat where
  wt (Hd (Var x)) = ε
| wt (Hd (Sym f)) = wt_sym f
| wt (App s t) = wt s + wt t

inductive gt :: ('s, 'v) tm ⇒ ('s, 'v) tm ⇒ bool (infix >t 50) where
  gt_wt: vars_mset t ⊇# vars_mset s ⇒ wt t > wt s ⇒ t >t s
| gt_sym_sym: wt_sym g = wt_sym f ⇒ g >s f ⇒ Hd (Sym g) >t Hd (Sym f)
| gt_sym_app: vars s = {} ⇒ wt t = wt s ⇒ t = Hd (Sym g) ⇒ is_App s ⇒ t >t s
| gt_app_app: vars_mset t ⊇# vars_mset s ⇒ wt t = wt s ⇒ t = App t1 t2 ⇒ s = App s1 s2 ⇒
  ext (>t) [t1, t2] [s1, s2] ⇒ t >t s

abbreviation ge :: ('s, 'v) tm ⇒ ('s, 'v) tm ⇒ bool (infix ≥t 50) where
  t ≥t s ≡ t >t s ∨ t = s

end

end

```

## 4 The Graceful Standard Knuth–Bendix Order for Lambda-Free Higher-Order Terms

```

theory Lambda_Free_KBO_Std
imports Lambda_Free_KBO_Util
abbrevs >t = >t
  and ≥t = ≥t
begin

```

This theory defines the standard version of the graceful Knuth–Bendix order for  $\lambda$ -free higher-order terms. Standard means that one symbol is allowed to have a weight of 0.

### 4.1 Setup

```

locale kbo_std = kbo_std_basis _ _ arity_sym arity_var wt_sym
  for
    arity_sym :: 's ⇒ enat and
    arity_var :: 'v ⇒ enat and
    wt_sym :: 's ⇒ nat
begin

```

### 4.2 Weights

```

primrec wt :: ('s, 'v) tm ⇒ nat where

```

$wt (Hd \zeta) = (LEAST w. \exists f \in ground\_heads \zeta. w = wt\_sym f + the\_enat (\delta * arity\_sym f))$   
 $| wt (App s t) = (wt s - \delta) + wt t$

**lemma**  $wt\_Hd\_Sym$ :  $wt (Hd (Sym f)) = wt\_sym f + the\_enat (\delta * arity\_sym f)$   
 $\langle proof \rangle$

**lemma**  $exists\_wt\_sym$ :  $\exists f \in ground\_heads \zeta. wt (Hd \zeta) = wt\_sym f + the\_enat (\delta * arity\_sym f)$   
 $\langle proof \rangle$

**lemma**  $wt\_le\_wt\_sym$ :  $f \in ground\_heads \zeta \implies wt (Hd \zeta) \leq wt\_sym f + the\_enat (\delta * arity\_sym f)$   
 $\langle proof \rangle$

**lemma**  $enat\_the\_enat\_delta\_times\_arity\_sym[simp]$ :  $enat (the\_enat (\delta * arity\_sym f)) = \delta * arity\_sym f$   
 $\langle proof \rangle$

**lemma**  $wt\_arg\_le$ :  $wt (arg s) \leq wt s$   
 $\langle proof \rangle$

**lemma**  $wt\_ge\_epsilon$ :  $wt s \geq \epsilon$   
 $\langle proof \rangle$

**lemma**  $wt\_ge\_delta$ :  $wt s \geq \delta$   
 $\langle proof \rangle$

**lemma**  $wt\_gt\_delta\_if\_superunary$ :  $arity\_hd (head s) > 1 \implies wt s > \delta$   
 $\langle proof \rangle$

**lemma**  $wt\_App\_delta$ :  $wt (App s t) = wt t \implies wt s = \delta$   
 $\langle proof \rangle$

**lemma**  $wt\_App\_ge\_fun$ :  $wt (App s t) \geq wt s$   
 $\langle proof \rangle$

**lemma**  $wt\_hd\_le$ :  $wt (Hd (head s)) \leq wt s$   
 $\langle proof \rangle$

**lemma**  $wt\_delta\_imp\_delta\_eq\_epsilon$ :  $wt s = \delta \implies \delta = \epsilon$   
 $\langle proof \rangle$

**lemma**  $wt\_ge\_arity\_head\_if\_delta\_gt\_0$ :  
**assumes**  $\delta\_gt\_0$ :  $\delta > 0$   
**shows**  $wt s \geq arity\_hd (head s)$   
 $\langle proof \rangle$

**lemma**  $wt\_ge\_num\_args\_if\_delta\_eq\_0$ :  
**assumes**  $\delta\_eq\_0$ :  $\delta = 0$   
**shows**  $wt s \geq num\_args s$   
 $\langle proof \rangle$

**lemma**  $wt\_ge\_num\_args$ :  $wary s \implies wt s \geq num\_args s$   
 $\langle proof \rangle$

### 4.3 Inductive Definitions

**inductive**  $gt :: ('s, 'v) tm \Rightarrow ('s, 'v) tm \Rightarrow bool$  (**infix**  $>_t$  50) **where**

$gt\_wt$ :  $vars\_mset t \supseteq \# vars\_mset s \implies wt t > wt s \implies t >_t s$   
 $| gt\_unary$ :  $wt t = wt s \implies \neg head t \leq_{hd} head s \implies num\_args t = 1 \implies$   
 $(\exists f \in ground\_heads (head t). arity\_sym f = 1 \wedge wt\_sym f = 0) \implies arg t >_t s \vee arg t = s \implies$   
 $t >_t s$   
 $| gt\_diff$ :  $vars\_mset t \supseteq \# vars\_mset s \implies wt t = wt s \implies head t >_{hd} head s \implies t >_t s$   
 $| gt\_same$ :  $vars\_mset t \supseteq \# vars\_mset s \implies wt t = wt s \implies head t = head s \implies$   
 $(\forall f \in ground\_heads (head t). extf f (>_t) (args t) (args s)) \implies t >_t s$

**abbreviation**  $ge :: ('s, 'v) tm \Rightarrow ('s, 'v) tm \Rightarrow bool$  (**infix**  $\geq_t$  50) **where**

$$t \geq_t s \equiv t >_t s \vee t = s$$

**inductive**  $gt\_wt :: ('s, 'v) tm \Rightarrow ('s, 'v) tm \Rightarrow bool$  **where**  
 $gt\_wtI: vars\_mset\ t \supseteq\# vars\_mset\ s \Longrightarrow wt\ t > wt\ s \Longrightarrow gt\_wt\ t\ s$

**inductive**  $gt\_diff :: ('s, 'v) tm \Rightarrow ('s, 'v) tm \Rightarrow bool$  **where**  
 $gt\_diffI: vars\_mset\ t \supseteq\# vars\_mset\ s \Longrightarrow wt\ t = wt\ s \Longrightarrow head\ t >_{hd}\ head\ s \Longrightarrow gt\_diff\ t\ s$

**inductive**  $gt\_unary :: ('s, 'v) tm \Rightarrow ('s, 'v) tm \Rightarrow bool$  **where**  
 $gt\_unaryI: wt\ t = wt\ s \Longrightarrow \neg head\ t \leq_{hd}\ head\ s \Longrightarrow num\_args\ t = 1 \Longrightarrow$   
 $(\exists f \in ground\_heads\ (head\ t). arity\_sym\ f = 1 \wedge wt\_sym\ f = 0) \Longrightarrow arg\ t \geq_t s \Longrightarrow gt\_unary\ t\ s$

**inductive**  $gt\_same :: ('s, 'v) tm \Rightarrow ('s, 'v) tm \Rightarrow bool$  **where**  
 $gt\_sameI: vars\_mset\ t \supseteq\# vars\_mset\ s \Longrightarrow wt\ t = wt\ s \Longrightarrow head\ t = head\ s \Longrightarrow$   
 $(\forall f \in ground\_heads\ (head\ t). extf\ f\ (>_t)\ (args\ t)\ (args\ s)) \Longrightarrow gt\_same\ t\ s$

**lemma**  $gt\_iff\_wt\_unary\_diff\_same: t >_t s \longleftrightarrow gt\_wt\ t\ s \vee gt\_unary\ t\ s \vee gt\_diff\ t\ s \vee gt\_same\ t\ s$   
 $\langle proof \rangle$

**lemma**  $gt\_imp\_vars\_mset: t >_t s \Longrightarrow vars\_mset\ t \supseteq\# vars\_mset\ s$   
 $\langle proof \rangle$

**lemma**  $gt\_imp\_vars: t >_t s \Longrightarrow vars\ t \supseteq vars\ s$   
 $\langle proof \rangle$

## 4.4 Irreflexivity

**theorem**  $gt\_irrefl: wary\ s \Longrightarrow \neg s >_t s$   
 $\langle proof \rangle$

## 4.5 Transitivity

**lemma**  $gt\_imp\_wt\_ge: t >_t s \Longrightarrow wt\ t \geq wt\ s$   
 $\langle proof \rangle$

**lemma**  $not\_extf\_gt\_nil\_singleton\_if\_delta\_eq\_epsilon:$   
**assumes**  $wary\_s: wary\ s$  **and**  $delta\_eq\_epsilon: \delta = \epsilon$   
**shows**  $\neg extf\ f\ (>_t)\ []\ [s]$   
 $\langle proof \rangle$

**lemma**  $gt\_sub\_arg: wary\ (App\ s\ t) \Longrightarrow App\ s\ t >_t t$   
 $\langle proof \rangle$

**lemma**  $gt\_arg: wary\ s \Longrightarrow is\_App\ s \Longrightarrow s >_t arg\ s$   
 $\langle proof \rangle$

**theorem**  $gt\_trans: wary\ u \Longrightarrow wary\ t \Longrightarrow wary\ s \Longrightarrow u >_t t \Longrightarrow t >_t s \Longrightarrow u >_t s$   
 $\langle proof \rangle$

**lemma**  $gt\_antisym: wary\ s \Longrightarrow wary\ t \Longrightarrow t >_t s \Longrightarrow \neg s >_t t$   
 $\langle proof \rangle$

## 4.6 Subterm Property

**lemma**  $gt\_sub\_fun: App\ s\ t >_t s$   
 $\langle proof \rangle$

**theorem**  $gt\_proper\_sub: wary\ t \Longrightarrow proper\_sub\ s\ t \Longrightarrow t >_t s$   
 $\langle proof \rangle$

## 4.7 Compatibility with Functions

**theorem**  $gt\_compat\_fun:$   
**assumes**

$wary\_t: wary\ t$  and  
 $t'\_gt\_t: t' >_t t$   
**shows**  $App\ s\ t' >_t App\ s\ t$   
 ⟨proof⟩

## 4.8 Compatibility with Arguments

**theorem**  $gt\_compat\_arg$ :  
**assumes**  $wary\_s't: wary\ (App\ s'\ t)$  and  $s'\_gt\_s: s' >_t s$   
**shows**  $App\ s'\ t >_t App\ s\ t$   
 ⟨proof⟩

## 4.9 Stability under Substitution

**definition**  $extra\_wt :: ('v \Rightarrow ('s, 'v)\ tm) \Rightarrow ('s, 'v)\ tm \Rightarrow nat$  **where**  
 $extra\_wt\ \rho\ s = sum\_mset\ \{\#wt\ (\rho\ x) - wt\ (Hd\ (Var\ x)).\ x \in\# vars\_mset\ s\#\}$

**lemma**  
 $extra\_wt\_Var[simp]: extra\_wt\ \rho\ (Hd\ (Var\ x)) = wt\ (\rho\ x) - wt\ (Hd\ (Var\ x))$  and  
 $extra\_wt\_Sym[simp]: extra\_wt\ \rho\ (Hd\ (Sym\ f)) = 0$  and  
 $extra\_wt\_App[simp]: extra\_wt\ \rho\ (App\ s\ t) = extra\_wt\ \rho\ s + extra\_wt\ \rho\ t$   
 ⟨proof⟩

**lemma**  $extra\_wt\_substeq$ :  
**assumes**  $vars\_s: vars\_mset\ t \supseteq\# vars\_mset\ s$   
**shows**  $extra\_wt\ \rho\ t \geq extra\_wt\ \rho\ s$   
 ⟨proof⟩

**lemma**  $wt\_subst$ :  
**assumes**  $wary\_rho: wary\_subst\ \rho$  and  $wary\_s: wary\ s$   
**shows**  $wt\ (subst\ \rho\ s) = wt\ s + extra\_wt\ \rho\ s$   
 ⟨proof⟩

**theorem**  $gt\_subst$ :  
**assumes**  $wary\_rho: wary\_subst\ \rho$   
**shows**  $wary\ t \Longrightarrow wary\ s \Longrightarrow t >_t s \Longrightarrow subst\ \rho\ t >_t subst\ \rho\ s$   
 ⟨proof⟩

## 4.10 Totality on Ground Terms

**theorem**  $gt\_total\_ground$ :  
**assumes**  
 $extf\_total: \bigwedge f. ext\_total\ (extf\ f)$  and  
 $gr\_t: ground\ t$  and  
 $gr\_s: ground\ s$   
**shows**  $t >_t s \vee s >_t t \vee t = s$   
 ⟨proof⟩

## 4.11 Well-foundedness

**abbreviation**  $gtw :: ('s, 'v)\ tm \Rightarrow ('s, 'v)\ tm \Rightarrow bool$  (**infix**  $>_{tw}$  50) **where**  
 $(>_{tw}) \equiv \lambda t\ s. wary\ t \wedge wary\ s \wedge t >_t s$

**abbreviation**  $gtwg :: ('s, 'v)\ tm \Rightarrow ('s, 'v)\ tm \Rightarrow bool$  (**infix**  $>_{twg}$  50) **where**  
 $(>_{twg}) \equiv \lambda t\ s. ground\ t \wedge t >_{tw}\ s$

**lemma**  $ground\_gt\_unary$ :  
**assumes**  $gr\_t: ground\ t$   
**shows**  $\neg gt\_unary\ t\ s$   
 ⟨proof⟩

**theorem**  $gt\_wf$ :  $wfP\ (\lambda s\ t. t >_{tw}\ s)$   
 ⟨proof⟩

end

end

## 5 The Graceful Basic Knuth–Bendix Order for Lambda-Free Higher-Order Terms

```
theory Lambda_Free_KBO_Basic
imports Lambda_Free_KBO_Std
begin
```

This theory defines the basic version of the graceful Knuth–Bendix order (KBO) for  $\lambda$ -free higher-order terms. Basic means that all symbols must have a positive weight. The results are lifted from the standard KBO.

```
locale kbo_basic = kbo_basic_basis _ _ _ ground_heads_var
  for ground_heads_var :: 'v  $\Rightarrow$  's set
begin
```

```
sublocale kbo_std: kbo_std _ _ _ 0 _  $\lambda$  .  $\infty$   $\lambda$  .  $\infty$ 
  <proof>
```

```
fun wt :: ('s, 'v) tm  $\Rightarrow$  nat where
  wt (Hd  $\zeta$ ) = (LEAST w.  $\exists f \in$  ground_heads  $\zeta$ . w = wt_sym f)
| wt (App s t) = wt s + wt t
```

```
inductive gt :: ('s, 'v) tm  $\Rightarrow$  ('s, 'v) tm  $\Rightarrow$  bool (infix  $>_t$  50) where
  gt_wt: vars_mset t  $\supseteq$  # vars_mset s  $\Longrightarrow$  wt t  $>$  wt s  $\Longrightarrow$  t  $>_t$  s
| gt_diff: vars_mset t  $\supseteq$  # vars_mset s  $\Longrightarrow$  wt t = wt s  $\Longrightarrow$  head t  $>_{hd}$  head s  $\Longrightarrow$  t  $>_t$  s
| gt_same: vars_mset t  $\supseteq$  # vars_mset s  $\Longrightarrow$  wt t = wt s  $\Longrightarrow$  head t = head s  $\Longrightarrow$ 
  ( $\forall f \in$  ground_heads (head s). extf f ( $>_t$ ) (args t) (args s))  $\Longrightarrow$  t  $>_t$  s
```

```
lemma arity_hd_eq_inf[simp]: arity_hd  $\zeta$  =  $\infty$ 
  <proof>
```

```
lemma waryI[intro, simp]: wary s
  <proof>
```

```
lemma basic_wt_eq_wt: wt s = kbo_std.wt s
  <proof>
```

```
lemma
  basic_gt_and_gt_le_gt: ( $\lambda$  t s. t  $>_t$  s  $\wedge$  local.kbo_std.gt t s)  $\leq$  kbo_std.gt and
  gt_and_basic_gt_le_basic_gt: ( $\lambda$  t s. local.kbo_std.gt t s  $\wedge$  t  $>_t$  s)  $\leq$  ( $>_t$ )
  <proof>
```

```
lemma basic_gt_iff_lt: t  $>_t$  s  $\longleftrightarrow$  kbo_std.gt t s
  <proof>
```

```
theorem gt_irrefl:  $\neg$  s  $>_t$  s
  <proof>
```

```
theorem gt_trans: u  $>_t$  t  $\Longrightarrow$  t  $>_t$  s  $\Longrightarrow$  u  $>_t$  s
  <proof>
```

```
theorem gt_proper_sub: proper_sub s t  $\Longrightarrow$  t  $>_t$  s
  <proof>
```

```
theorem gt_compat_fun: t'  $>_t$  t  $\Longrightarrow$  App s t'  $>_t$  App s t
  <proof>
```

```
theorem gt_compat_arg: s'  $>_t$  s  $\Longrightarrow$  App s' t  $>_t$  App s t
  <proof>
```



**theorem** *gt\_subst*:  $wary\_subst\ \rho \implies t >_t s \implies subst\ \rho\ t >_t subst\ \rho\ s$   
 ⟨*proof*⟩

**theorem** *gt\_wf*:  $wfP\ (\lambda s\ t.\ t >_t s)$   
 ⟨*proof*⟩

**end**

**end**

## 6 The Graceful Transfinite Knuth–Bendix Order with Subterm Coefficients for Lambda-Free Higher-Order Terms

**theory** *Lambda\_Free\_TKBO\_Coefs*  
**imports** *Lambda\_Free\_KBO\_Util Nested\_Multisets\_Ordinals.Signed\_Syntactic\_Ordinal*  
**abbrevs**  $=_p =_p$   
**and**  $>_p = >_p$   
**and**  $\geq_p = \geq_p$   
**and**  $>_t = >_t$   
**and**  $\geq_t = \geq_t$   
**and**  $!h =_h$   
**begin**

This theory defines the graceful transfinite Knuth–Bendix order (KBO) with subterm coefficients for  $\lambda$ -free higher-order terms. The proof was developed by copying that of the standard KBO and generalizing it along two axes: subterm coefficients and ordinals. Both features complicate the definitions and proofs substantially.

### 6.1 Setup

**hide-const** (open) *Complex.arg*

**locale** *tkbo\_coefs* = *kbo\_std\_basis* \_ \_ *arity\_sym* *arity\_var* *wt\_sym*  
**for**  
*arity\_sym* ::  $'s \Rightarrow enat$  **and**  
*arity\_var* ::  $'v \Rightarrow enat$  **and**  
*wt\_sym* ::  $'s \Rightarrow hmultiset$  +  
**fixes** *coef\_sym* ::  $'s \Rightarrow nat \Rightarrow hmultiset$   
**assumes** *coef\_sym\_gt\_0*:  $coef\_sym\ f\ i > 0$   
**begin**

**abbreviation**  $\delta_h$  ::  $hmultiset$  **where**  
 $\delta_h \equiv of\_nat\ \delta$

**abbreviation**  $\varepsilon_h$  ::  $hmultiset$  **where**  
 $\varepsilon_h \equiv of\_nat\ \varepsilon$

**abbreviation** *arity\_sym<sub>h</sub>* ::  $'s \Rightarrow hmultiset$  **where**  
*arity\_sym<sub>h</sub>*  $f \equiv hmsset\_of\_enat\ (arity\_sym\ f)$

**abbreviation** *arity\_var<sub>h</sub>* ::  $'v \Rightarrow hmultiset$  **where**  
*arity\_var<sub>h</sub>*  $f \equiv hmsset\_of\_enat\ (arity\_var\ f)$

**abbreviation** *arity\_hd<sub>h</sub>* ::  $('s, 'v) hd \Rightarrow hmultiset$  **where**  
*arity\_hd<sub>h</sub>*  $f \equiv hmsset\_of\_enat\ (arity\_hd\ f)$

**abbreviation** *arity<sub>h</sub>* ::  $('s, 'v) tm \Rightarrow hmultiset$  **where**  
*arity<sub>h</sub>*  $s \equiv hmsset\_of\_enat\ (arity\ s)$

**lemma** *arity<sub>h</sub>\_conv*:  $arity\_h\ s = arity\_hd\_h\ (head\ s) - of\_nat\ (num\_args\ s)$   
 ⟨*proof*⟩

**lemma**  $arity_h\_App[simp]$ :  $arity_h (App\ s\ t) = arity_h\ s - 1$   
 ⟨proof⟩

**lemmas**  $wary\_App_h[intro]$  =  $wary\_App[folded\ of\_nat\ lt\_hmset\ of\_enat\_iff]$

**lemmas**  $wary\_AppE_h$  =  $wary\_AppE[folded\ of\_nat\ lt\_hmset\ of\_enat\_iff]$

**lemmas**  $wary\_num\_args\_le\_arity\_head_h$  =  
 $wary\_num\_args\_le\_arity\_head[folded\ of\_nat\_le\_hmset\ of\_enat\_iff]$

**lemmas**  $wary\_apps_h$  =  $wary\_apps[folded\ of\_nat\_le\_hmset\ of\_enat\_iff]$

**lemmas**  $wary\_cases\_apps_h[consumes\ 1, case\_names\ apps]$  =  
 $wary\_cases\_apps[folded\ of\_nat\_le\_hmset\ of\_enat\_iff]$

**lemmas**  $ground\_heads\_arity_h$  =  $ground\_heads\_arity[folded\ hmset\ of\_enat\_le]$

**lemmas**  $some\_ground\_head\_arity_h$  =  $some\_ground\_head\_arity[folded\ hmset\ of\_enat\_le]$

**lemmas**  $\varepsilon_h\_gt\_0$  =  $\varepsilon\_gt\_0[folded\ of\_nat\_less\_hmset, unfolded\ of\_nat\_0]$

**lemmas**  $\delta_h\_le\_e_h$  =  $\delta\_le\_e[folded\ of\_nat\_le\_hmset]$

**lemmas**  $arity\_hd_h\_lt\_w\_if\_delta_h\_gt\_0$  =  $arity\_hd\_ne\_infinity\_if\_delta\_gt\_0$   
 $[folded\ of\_nat\_less\_hmset, unfolded\ of\_nat\_0, folded\ hmset\ of\_enat\_lt\_iff\_ne\_infinity]$

**lemma**  $wt\_sym\_ge_h$ :  $wt\_sym\ f \geq \varepsilon_h - \delta_h * arity\_sym_h\ f$   
 ⟨proof⟩

**lemmas**  $unary\_wt\_sym\_0\_gt_h$  =  $unary\_wt\_sym\_0\_gt[folded\ hmset\ of\_enat\_inject, unfolded\ hmset\ of\_enat\_1]$

**lemmas**  $unary\_wt\_sym\_0\_imp\_delta_h\_eq\_e_h$  =  $unary\_wt\_sym\_0\_imp\_delta\_eq\_e$   
 $[folded\ of\_nat\_inject\_hmset, unfolded\ of\_nat\_0]$

**lemmas**  $extf\_ext\_snoc\_if\_delta_h\_eq\_e_h$  =  $extf\_ext\_snoc\_if\_delta\_eq\_e[folded\ of\_nat\_inject\_hmset]$

**lemmas**  $extf\_snoc\_if\_delta_h\_eq\_e_h$  =  $ext\_snoc.snoc[OF\ extf\_ext\_snoc\_if\_delta_h\_eq\_e_h]$

**lemmas**  $arity\_sym_h\_lt\_w\_if\_delta_h\_gt\_0$  =  $arity\_sym\_ne\_infinity\_if\_delta\_gt\_0$   
 $[folded\ of\_nat\_less\_hmset\ hmset\ of\_enat\_lt\_iff\_ne\_infinity, unfolded\ of\_nat\_0]$

**lemmas**  $arity\_var_h\_lt\_w\_if\_delta_h\_gt\_0$  =  $arity\_var\_ne\_infinity\_if\_delta\_gt\_0$   
 $[folded\ of\_nat\_less\_hmset\ hmset\ of\_enat\_lt\_iff\_ne\_infinity, unfolded\ of\_nat\_0]$

**lemmas**  $arity_h\_lt\_w\_if\_delta_h\_gt\_0$  =  $arity\_ne\_infinity\_if\_delta\_gt\_0$   
 $[folded\ of\_nat\_less\_hmset\ hmset\ of\_enat\_lt\_iff\_ne\_infinity, unfolded\ of\_nat\_0]$

**lemmas**  $warywary\_subst\_subst_h\_conv$  =  $wary\_subst\_def[folded\ hmset\ of\_enat\_le]$

**lemmas**  $extf\_singleton\_nil\_if\_delta_h\_eq\_e_h$  =  $extf\_singleton\_nil\_if\_delta\_eq\_e[folded\ of\_nat\_inject\_hmset]$

**lemma**  $arity\_sym_h\_if\_delta_h\_gt\_0\_E$ :  
**assumes**  $\delta\_gt\_0$ :  $\delta_h > 0$   
**obtains**  $n$  **where**  $arity\_sym_h\ f = of\_nat\ n$   
 ⟨proof⟩

**lemma**  $arity\_var_h\_if\_delta_h\_gt\_0\_E$ :  
**assumes**  $\delta\_gt\_0$ :  $\delta_h > 0$   
**obtains**  $n$  **where**  $arity\_var_h\ f = of\_nat\ n$   
 ⟨proof⟩

## 6.2 Weights and Subterm Coefficients

**abbreviation**  $zhmset\_of\_tpoly$  ::  $('a, hmset) tpoly \Rightarrow ('a, zhmultiset) tpoly$  **where**  
 $zhmset\_of\_tpoly \equiv map\_tpoly (\lambda x. x) zhmset\_of$

**abbreviation**  $eval\_ztpoly$  ::  $('a \Rightarrow zhmultiset) \Rightarrow ('a, hmset) tpoly \Rightarrow zhmultiset$  **where**  
 $eval\_ztpoly\ A\ p \equiv eval\_tpoly\ A\ (zhmset\_of\_tpoly\ p)$

**lemma**  $eval\_tpoly\_eq\_eval\_ztpoly[simp]$ :  
 $zhmset\_of\ (eval\_tpoly\ A\ p) = eval\_ztpoly\ (\lambda v. zhmset\_of\ (A\ v))\ p$   
 ⟨proof⟩

**definition**  $min\_ground\_head$  ::  $('s, 'v) hd \Rightarrow 's$  **where**  
 $min\_ground\_head\ \zeta =$   
 $(SOME\ f. f \in ground\_heads\ \zeta \wedge$   
 $(\forall g \in ground\_heads\ \zeta. wt\_sym\ g + \delta_h * arity\_sym_h\ g \geq wt\_sym\ f + \delta_h * arity\_sym_h\ f))$

**datatype**  $'va\ pvar$  =  
 $PWt\ 'va$

| *PCoef* 'va nat

**primrec** *min\_passign* :: 'v pvar  $\Rightarrow$  *hmultiset* **where**  
  *min\_passign* (*PWt* x) = *wt\_sym* (*min\_ground\_head* (*Var* x))  
| *min\_passign* (*PCoef* \_ \_) = 1

**abbreviation** *min\_zpassign* :: 'v pvar  $\Rightarrow$  *zhmultiset* **where**  
  *min\_zpassign* v  $\equiv$  *zhmset\_of* (*min\_passign* v)

**lemma** *min\_zpassign\_simps*[*simp*]:  
  *min\_zpassign* (*PWt* x) = *zhmset\_of* (*wt\_sym* (*min\_ground\_head* (*Var* x)))  
  *min\_zpassign* (*PCoef* x i) = 1  
  <*proof*>

**definition** *legal\_passign* :: ('v pvar  $\Rightarrow$  *hmultiset*)  $\Rightarrow$  *bool* **where**  
  *legal\_passign* A  $\longleftrightarrow$  ( $\forall x. A x \geq \text{min\_passign } x$ )

**definition** *legal\_zpassign* :: ('v pvar  $\Rightarrow$  *zhmultiset*)  $\Rightarrow$  *bool* **where**  
  *legal\_zpassign* A  $\longleftrightarrow$  ( $\forall x. A x \geq \text{min\_zpassign } x$ )

**lemma** *legal\_min\_passign*: *legal\_passign* *min\_passign*  
  <*proof*>

**lemma** *legal\_min\_zpassign*: *legal\_zpassign* *min\_zpassign*  
  <*proof*>

**lemma** *assign\_ge\_0*[*intro*]: *legal\_zpassign* A  $\Longrightarrow$  A x  $\geq$  0  
  <*proof*>

**definition**  
  *eq\_tpoly* :: ('v pvar, *hmultiset*) *tpoly*  $\Rightarrow$  ('v pvar, *hmultiset*) *tpoly*  $\Rightarrow$  *bool* (**infix** =<sub>p</sub> 50)  
**where**  
  *q* =<sub>p</sub> *p*  $\longleftrightarrow$  ( $\forall A. \text{legal\_zpassign } A \longrightarrow \text{eval\_ztpoly } A q = \text{eval\_ztpoly } A p$ )

**definition**  
  *ge\_tpoly* :: ('v pvar, *hmultiset*) *tpoly*  $\Rightarrow$  ('v pvar, *hmultiset*) *tpoly*  $\Rightarrow$  *bool* (**infix**  $\geq_p$  50)  
**where**  
  *q*  $\geq_p$  *p*  $\longleftrightarrow$  ( $\forall A. \text{legal\_zpassign } A \longrightarrow \text{eval\_ztpoly } A q \geq \text{eval\_ztpoly } A p$ )

**definition**  
  *gt\_tpoly* :: ('v pvar, *hmultiset*) *tpoly*  $\Rightarrow$  ('v pvar, *hmultiset*) *tpoly*  $\Rightarrow$  *bool* (**infix** ><sub>p</sub> 50)  
**where**  
  *q* ><sub>p</sub> *p*  $\longleftrightarrow$  ( $\forall A. \text{legal\_zpassign } A \longrightarrow \text{eval\_ztpoly } A q > \text{eval\_ztpoly } A p$ )

**lemma** *gt\_tpoly\_imp\_ge*[*intro*]: *q* ><sub>p</sub> *p*  $\Longrightarrow$  *q*  $\geq_p$  *p*  
  <*proof*>

**lemma** *eq\_tpoly\_refl*[*simp*]: *p* =<sub>p</sub> *p*  
  <*proof*>

**lemma** *ge\_tpoly\_refl*[*simp*]: *p*  $\geq_p$  *p*  
  <*proof*>

**lemma** *gt\_tpoly\_irrefl*:  $\neg p >_p p$   
  <*proof*>

**lemma**  
  *eq\_eq\_tpoly\_trans*: *r* =<sub>p</sub> *q*  $\Longrightarrow$  *q* =<sub>p</sub> *p*  $\Longrightarrow$  *r* =<sub>p</sub> *p* **and**  
  *eq\_ge\_tpoly\_trans*: *r* =<sub>p</sub> *q*  $\Longrightarrow$  *q*  $\geq_p$  *p*  $\Longrightarrow$  *r*  $\geq_p$  *p* **and**  
  *eq\_gt\_tpoly\_trans*: *r* =<sub>p</sub> *q*  $\Longrightarrow$  *q* ><sub>p</sub> *p*  $\Longrightarrow$  *r* ><sub>p</sub> *p* **and**  
  *ge\_eq\_tpoly\_trans*: *r*  $\geq_p$  *q*  $\Longrightarrow$  *q* =<sub>p</sub> *p*  $\Longrightarrow$  *r*  $\geq_p$  *p* **and**  
  *ge\_ge\_tpoly\_trans*: *r*  $\geq_p$  *q*  $\Longrightarrow$  *q*  $\geq_p$  *p*  $\Longrightarrow$  *r*  $\geq_p$  *p* **and**  
  *ge\_gt\_tpoly\_trans*: *r*  $\geq_p$  *q*  $\Longrightarrow$  *q* ><sub>p</sub> *p*  $\Longrightarrow$  *r* ><sub>p</sub> *p* **and**

$gt\_eq\_tpoly\_trans: r >_p q \implies q =_p p \implies r >_p p$  **and**  
 $gt\_ge\_tpoly\_trans: r >_p q \implies q \geq_p p \implies r >_p p$  **and**  
 $gt\_gt\_tpoly\_trans: r >_p q \implies q >_p p \implies r >_p p$   
 <proof>

**primrec**  $coef\_hd :: ('s, 'v) hd \Rightarrow nat \Rightarrow ('v\ pvar, hmultiset) tpoly$  **where**  
 $coef\_hd (Var\ x)\ i = PVar\ (PCoef\ x\ i)$   
 $| coef\_hd (Sym\ f)\ i = PNum\ (coef\_sym\ f\ i)$

**lemma**  $coef\_hd\_gt\_0$ :  
**assumes**  $legal\_zpassign\ A$   
**shows**  $eval\_ztpoly\ A\ (coef\_hd\ \zeta\ i) > 0$   
 <proof>

**primrec**  $coef :: ('s, 'v) tm \Rightarrow nat \Rightarrow ('v\ pvar, hmultiset) tpoly$  **where**  
 $coef (Hd\ \zeta)\ i = coef\_hd\ \zeta\ i$   
 $| coef (App\ s\ \_)\ i = coef\ s\ (i + 1)$

**lemma**  $coef\_apps[simp]: coef\ (apps\ s\ ss)\ i = coef\ s\ (i + length\ ss)$   
 <proof>

**lemma**  $coef\_gt\_0: legal\_zpassign\ A \implies eval\_ztpoly\ A\ (coef\ s\ i) > 0$   
 <proof>

**lemma**  $exists\_min\_ground\_head$ :  
 $\exists f. f \in ground\_heads\ \zeta \wedge$   
 $(\forall g \in ground\_heads\ \zeta. wt\_sym\ g + \delta_h * arity\_sym_h\ g \geq wt\_sym\ f + \delta_h * arity\_sym_h\ f)$   
 <proof>

**lemma**  $min\_ground\_head\_Sym[simp]: min\_ground\_head\ (Sym\ f) = f$   
 <proof>

**lemma**  $min\_ground\_head\_in\_ground\_heads: min\_ground\_head\ \zeta \in ground\_heads\ \zeta$   
 <proof>

**lemma**  $min\_ground\_head\_min$ :  
 $f \in ground\_heads\ \zeta \implies$   
 $wt\_sym\ f + \delta_h * arity\_sym_h\ f \geq wt\_sym\ (min\_ground\_head\ \zeta) + \delta_h * arity\_sym_h\ (min\_ground\_head\ \zeta)$   
 <proof>

**lemma**  $min\_ground\_head\_antimono$ :  
 $ground\_heads\ \zeta \subseteq ground\_heads\ \xi \implies$   
 $wt\_sym\ (min\_ground\_head\ \zeta) + \delta_h * arity\_sym_h\ (min\_ground\_head\ \zeta)$   
 $\geq wt\_sym\ (min\_ground\_head\ \xi) + \delta_h * arity\_sym_h\ (min\_ground\_head\ \xi)$   
 <proof>

**primrec**  $wt0 :: ('s, 'v) hd \Rightarrow ('v\ pvar, hmultiset) tpoly$  **where**  
 $wt0 (Var\ x) = PVar\ (PWt\ x)$   
 $| wt0 (Sym\ f) = PNum\ (wt\_sym\ f)$

**lemma**  $wt0\_ge\_min\_ground\_head$ :  
 $legal\_zpassign\ A \implies eval\_ztpoly\ A\ (wt0\ \zeta) \geq zhmsset\_of\ (wt\_sym\ (min\_ground\_head\ \zeta))$   
 <proof>

**lemma**  $eval\_ztpoly\_nonneg: legal\_zpassign\ A \implies eval\_ztpoly\ A\ p \geq 0$   
 <proof>

**lemma**  $in\_zip\_imp\_size\_lt\_apps: (s, y) \in set\ (zip\ ss\ ys) \implies size\ s < size\ (apps\ (Hd\ \zeta)\ ss)$   
 <proof>

**function**  $wt :: ('s, 'v) tm \Rightarrow ('v\ pvar, hmultiset) tpoly$  **where**  
 $wt\ (apps\ (Hd\ \zeta)\ ss) =$   
 $PSum\ ([wt0\ \zeta, PNum\ (\delta_h * (arity\_sym_h\ (min\_ground\_head\ \zeta) - of\_nat\ (length\ ss)))]\ @$

$\text{map } (\lambda(s, i). \text{PMult } [\text{coef\_hd } \zeta \ i, \text{wt } s]) (\text{zip } ss [0..<\text{length } ss])$   
 <proof>

**termination**

<proof>

**definition**

$\text{wt\_args} :: \text{nat} \Rightarrow ('v \text{ pvar} \Rightarrow \text{zhmultiset}) \Rightarrow ('s, 'v) \text{ hd} \Rightarrow ('s, 'v) \text{ tm list} \Rightarrow \text{zhmultiset}$

**where**

$\text{wt\_args } i \ A \ \zeta \ ss = \text{sum\_list}$   
 $(\text{map } (\text{eval\_ztpoly } A \circ (\lambda(s, i). \text{PMult } [\text{coef\_hd } \zeta \ i, \text{wt } s])) (\text{zip } ss [i..<i + \text{length } ss]))$

**lemma**  $\text{wt\_Hd}[simp]$ :  $\text{wt } (\text{Hd } \zeta) = \text{PSum } [\text{wt0 } \zeta, \text{PNum } (\delta_h * \text{arity\_sym}_h (\text{min\_ground\_head } \zeta))]$

<proof>

**lemma**  $\text{coef\_hd\_cong}$ :

$(\forall x \in \text{vars\_hd } \zeta. \forall i. A (\text{PCoef } x \ i) = B (\text{PCoef } x \ i)) \implies$   
 $\text{eval\_ztpoly } A (\text{coef\_hd } \zeta \ i) = \text{eval\_ztpoly } B (\text{coef\_hd } \zeta \ i)$   
 <proof>

**lemma**  $\text{wt0\_cong}$ :

**assumes**  $\text{pwt\_eq}$ :  $\forall x \in \text{vars\_hd } \zeta. A (\text{PWt } x) = B (\text{PWt } x)$   
**shows**  $\text{eval\_ztpoly } A (\text{wt0 } \zeta) = \text{eval\_ztpoly } B (\text{wt0 } \zeta)$   
 <proof>

**lemma**  $\text{wt\_cong}$ :

**assumes**  
 $\forall x \in \text{vars } s. A (\text{PWt } x) = B (\text{PWt } x)$  **and**  
 $\forall x \in \text{vars } s. \forall i. A (\text{PCoef } x \ i) = B (\text{PCoef } x \ i)$   
**shows**  $\text{eval\_ztpoly } A (\text{wt } s) = \text{eval\_ztpoly } B (\text{wt } s)$   
 <proof>

**lemma**  $\text{ground\_eval\_ztpoly\_wt\_eq}$ :  $\text{ground } s \implies \text{eval\_ztpoly } A (\text{wt } s) = \text{eval\_ztpoly } B (\text{wt } s)$

<proof>

**lemma**  $\text{exists\_wt\_sym}$ :

**assumes**  $\text{legal}$ :  $\text{legal\_zpassign } A$   
**shows**  $\exists f \in \text{ground\_heads } \zeta. \text{eval\_ztpoly } A (\text{wt } (\text{Hd } \zeta)) \geq \text{zhmset\_of } (\text{wt\_sym } f + \delta_h * \text{arity\_sym}_h \ f)$   
 <proof>

**lemma**  $\text{wt\_ge\_}\varepsilon_h$ :

**assumes**  $\text{legal}$ :  $\text{legal\_zpassign } A$   
**shows**  $\text{eval\_ztpoly } A (\text{wt } s) \geq \text{zhmset\_of } \varepsilon_h$   
 <proof>

**lemma**  $\text{wt\_args\_ge\_length\_times\_}\varepsilon_h$ :

**assumes**  $\text{legal}$ :  $\text{legal\_zpassign } A$   
**shows**  $\text{wt\_args } i \ A \ \zeta \ ss \geq \text{of\_nat } (\text{length } ss) * \text{zhmset\_of } \varepsilon_h$   
 <proof>

**lemma**  $\text{wt\_ge\_}\delta_h$ :  $\text{legal\_zpassign } A \implies \text{eval\_ztpoly } A (\text{wt } s) \geq \text{zhmset\_of } \delta_h$

<proof>

**lemma**  $\text{wt\_gt\_}0$ :  $\text{legal\_zpassign } A \implies \text{eval\_ztpoly } A (\text{wt } s) > 0$

<proof>

**lemma**  $\text{wt\_gt\_}\delta_h$ \_if\_superunary:

**assumes**  
 $\text{legal}$ :  $\text{legal\_zpassign } A$  **and**  
 $\text{superunary}$ :  $\text{arity\_hd}_h (\text{head } s) > 1$   
**shows**  $\text{eval\_ztpoly } A (\text{wt } s) > \text{zhmset\_of } \delta_h$   
 <proof>

**lemma**  $\text{wt\_App\_plus\_}\delta_h$ \_ge:

$eval\_ztpoly\ A\ (wt\ (App\ s\ t)) + zhmsset\_of\ \delta_h$   
 $\geq eval\_ztpoly\ A\ (wt\ s) + eval\_ztpoly\ A\ (coef\ s\ 0) * eval\_ztpoly\ A\ (wt\ t)$   
 <proof>

**lemma**  $wt\_App\_fun\_delta_h$ :  
**assumes**  
 $legal: legal\_zpassign\ A$  **and**  
 $wt\_st: eval\_ztpoly\ A\ (wt\ (App\ s\ t)) = eval\_ztpoly\ A\ (wt\ t)$   
**shows**  $eval\_ztpoly\ A\ (wt\ s) = zhmsset\_of\ \delta_h$   
 <proof>

**lemma**  $wt\_App\_arg\_delta_h$ :  
**assumes**  
 $legal: legal\_zpassign\ A$  **and**  
 $wt\_st: eval\_ztpoly\ A\ (wt\ (App\ s\ t)) = eval\_ztpoly\ A\ (wt\ s)$   
**shows**  $eval\_ztpoly\ A\ (wt\ t) = zhmsset\_of\ \delta_h$   
 <proof>

**lemma**  $wt\_App\_ge\_fun$ :  $wt\ (App\ s\ t) \geq_p wt\ s$   
 <proof>

**lemma**  $wt\_App\_ge\_arg$ :  $wt\ (App\ s\ t) \geq_p wt\ t$   
 <proof>

**lemma**  $wt\_delta_h\_imp\_delta_h\_eq\_epsilon_h$ :  
**assumes**  
 $legal: legal\_zpassign\ A$  **and**  
 $wt\_s\_eq\_delta: eval\_ztpoly\ A\ (wt\ s) = zhmsset\_of\ \delta_h$   
**shows**  $\delta_h = \epsilon_h$   
 <proof>

**lemma**  $wt\_ge\_vars$ :  $wt\ t \geq_p wt\ s \implies vars\ t \supseteq vars\ s$   
 <proof>

**lemma**  $sum\_coefs\_ge\_num\_args\_if\_delta_h\_eq\_0$ :  
**assumes**  
 $legal: legal\_passign\ A$  **and**  
 $delta\_eq\_0: \delta_h = 0$  **and**  
 $wary\_s: wary\ s$   
**shows**  $sum\_coefs\ (eval\_tpoly\ A\ (wt\ s)) \geq num\_args\ s$   
 <proof>

### 6.3 Inductive Definitions

**inductive**  $gt :: ('s, 'v)\ tm \Rightarrow ('s, 'v)\ tm \Rightarrow bool$  (**infix**  $>_t$  50) **where**  
 $gt\_wt: wt\ t >_p wt\ s \implies t >_t s$   
 $| gt\_unary: wt\ t \geq_p wt\ s \implies \neg head\ t \leq_{hd} head\ s \implies num\_args\ t = 1 \implies$   
 $(\exists f \in ground\_heads\ (head\ t). arity\_sym\ f = 1 \wedge wt\_sym\ f = 0) \implies arg\ t >_t s \vee arg\ t = s \implies$   
 $t >_t s$   
 $| gt\_diff: wt\ t \geq_p wt\ s \implies head\ t >_{hd} head\ s \implies t >_t s$   
 $| gt\_same: wt\ t \geq_p wt\ s \implies head\ t = head\ s \implies$   
 $(\forall f \in ground\_heads\ (head\ t). extf\ f\ (>_t)\ (args\ t)\ (args\ s)) \implies t >_t s$

**abbreviation**  $ge :: ('s, 'v)\ tm \Rightarrow ('s, 'v)\ tm \Rightarrow bool$  (**infix**  $\geq_t$  50) **where**  
 $t \geq_t s \equiv t >_t s \vee t = s$

**inductive**  $gt\_wt :: ('s, 'v)\ tm \Rightarrow ('s, 'v)\ tm \Rightarrow bool$  **where**  
 $gt\_wtI: wt\ t >_p wt\ s \implies gt\_wt\ t\ s$

**inductive**  $gt\_unary :: ('s, 'v)\ tm \Rightarrow ('s, 'v)\ tm \Rightarrow bool$  **where**  
 $gt\_unaryI: wt\ t \geq_p wt\ s \implies \neg head\ t \leq_{hd} head\ s \implies num\_args\ t = 1 \implies$   
 $(\exists f \in ground\_heads\ (head\ t). arity\_sym\ f = 1 \wedge wt\_sym\ f = 0) \implies arg\ t \geq_t s \implies gt\_unary\ t\ s$

**inductive**  $gt\_diff :: ('s, 'v)\ tm \Rightarrow ('s, 'v)\ tm \Rightarrow bool$  **where**

$gt\_diffI: wt\ t \geq_p wt\ s \implies head\ t >_{hd}\ head\ s \implies gt\_diff\ t\ s$

**inductive**  $gt\_same :: ('s, 'v)\ tm \Rightarrow ('s, 'v)\ tm \Rightarrow bool$  **where**  
 $gt\_sameI: wt\ t \geq_p wt\ s \implies head\ t = head\ s \implies$   
 $(\forall f \in ground\_heads\ (head\ t).\ extf\ f\ (>t)\ (args\ t)\ (args\ s)) \implies gt\_same\ t\ s$

**lemma**  $gt\_iff\_wt\_unary\_diff\_same: t >_t s \iff gt\_wt\ t\ s \vee gt\_unary\ t\ s \vee gt\_diff\ t\ s \vee gt\_same\ t\ s$   
 $\langle proof \rangle$

**lemma**  $gt\_imp\_wt: t >_t s \implies wt\ t \geq_p wt\ s$   
 $\langle proof \rangle$

**lemma**  $gt\_imp\_vars: t >_t s \implies vars\ t \supseteq vars\ s$   
 $\langle proof \rangle$

## 6.4 Irreflexivity

**theorem**  $gt\_irrefl: wary\ s \implies \neg s >_t s$   
 $\langle proof \rangle$

## 6.5 Transitivity

**lemma**  $not\_extf\_gt\_nil\_singleton\_if\_delta\_eq\_epsilon:$   
**assumes**  $wary\_s: wary\ s$  **and**  $\delta\_eq\_epsilon: \delta_h = \epsilon_h$   
**shows**  $\neg extf\ f\ (>t)\ []\ [s]$   
 $\langle proof \rangle$

**lemma**  $gt\_sub\_arg: wary\ (App\ s\ t) \implies App\ s\ t >_t t$   
 $\langle proof \rangle$

**lemma**  $gt\_arg: wary\ s \implies is\_App\ s \implies s >_t arg\ s$   
 $\langle proof \rangle$

**theorem**  $gt\_trans: wary\ u \implies wary\ t \implies wary\ s \implies u >_t t \implies t >_t s \implies u >_t s$   
 $\langle proof \rangle$

**lemma**  $gt\_antisym: wary\ s \implies wary\ t \implies t >_t s \implies \neg s >_t t$   
 $\langle proof \rangle$

## 6.6 Subterm Property

**lemma**  $gt\_sub\_fun: App\ s\ t >_t s$   
 $\langle proof \rangle$

**theorem**  $gt\_proper\_sub: wary\ t \implies proper\_sub\ s\ t \implies t >_t s$   
 $\langle proof \rangle$

## 6.7 Compatibility with Functions

**lemma**  $gt\_compat\_fun:$   
**assumes**  
 $wary\_t: wary\ t$  **and**  
 $t'\_gt\_t: t' >_t t$   
**shows**  $App\ s\ t' >_t App\ s\ t$   
 $\langle proof \rangle$

**theorem**  $gt\_compat\_fun\_strong:$   
**assumes**  
 $wary\_t: wary\ t$  **and**  
 $t'\_gt\_t: t' >_t t$   
**shows**  $apps\ s\ (t' \# us) >_t apps\ s\ (t \# us)$   
 $\langle proof \rangle$

## 6.8 Compatibility with Arguments

**theorem** *gt\_compat\_arg\_weak*:

**assumes**

*wary\_st*: *wary* (*App s t*) **and**

*wary\_s't*: *wary* (*App s' t*) **and**

*coef\_s'\_0\_ge\_s*: *coef s' 0*  $\geq_p$  *coef s 0* **and**

*s'\_gt\_s*: *s' ><sub>t</sub> s*

**shows** *App s' t ><sub>t</sub> App s t*

*<proof>*

## 6.9 Stability under Substitution

**primrec**

*subst\_zpassign* :: (*'v*  $\Rightarrow$  (*'s*, *'v*) *tm*)  $\Rightarrow$  (*'v pvar*  $\Rightarrow$  *zhmultiset*)  $\Rightarrow$  *'v pvar*  $\Rightarrow$  *zhmultiset*

**where**

*subst\_zpassign*  $\rho$  *A* (*PWt x*) =

*eval\_ztpoly* *A* (*wt* ( $\rho$  *x*)) - *zhmset\_of* ( $\delta_n * \text{arity\_sym}_n$  (*min\_ground\_head* (*Var x*)))

| *subst\_zpassign*  $\rho$  *A* (*PCoef x i*) = *eval\_ztpoly* *A* (*coef* ( $\rho$  *x*) *i*)

**lemma** *legal\_subst\_zpassign*:

**assumes**

*legal*: *legal\_zpassign* *A* **and**

*wary\_rho*: *wary\_subst*  $\rho$

**shows** *legal\_zpassign* (*subst\_zpassign*  $\rho$  *A*)

*<proof>*

**lemma** *wt\_subst*:

**assumes**

*legal*: *legal\_zpassign* *A* **and**

*wary\_rho*: *wary\_subst*  $\rho$

**shows** *wary s*  $\Longrightarrow$  *eval\_ztpoly* *A* (*wt* (*subst*  $\rho$  *s*)) = *eval\_ztpoly* (*subst\_zpassign*  $\rho$  *A*) (*wt s*)

*<proof>*

**theorem** *gt\_subst*:

**assumes** *wary\_rho*: *wary\_subst*  $\rho$

**shows** *wary t*  $\Longrightarrow$  *wary s*  $\Longrightarrow$  *t ><sub>t</sub> s*  $\Longrightarrow$  *subst*  $\rho$  *t*  $><sub>t</sub>$  *subst*  $\rho$  *s*

*<proof>*

## 6.10 Totality on Ground Terms

**lemma** *wt\_total\_ground*:

**assumes**

*gr\_t*: *ground t* **and**

*gr\_s*: *ground s*

**shows** *wt t*  $>_p$  *wt s*  $\vee$  *wt s*  $>_p$  *wt t*  $\vee$  *wt t*  $=_p$  *wt s*

*<proof>*

**theorem** *gt\_total\_ground*:

**assumes**

*extf\_total*:  $\bigwedge f. \text{ext\_total}$  (*extf*) **and**

*gr\_t*: *ground t* **and**

*gr\_s*: *ground s*

**shows** *t ><sub>t</sub> s*  $\vee$  *s ><sub>t</sub> t*  $\vee$  *t = s*

*<proof>*

## 6.11 Well-foundedness

**abbreviation** *gtw* :: (*'s*, *'v*) *tm*  $\Rightarrow$  (*'s*, *'v*) *tm*  $\Rightarrow$  *bool* (**infix**  $>_{tw}$  50) **where**

$(>_{tw}) \equiv \lambda t s. \text{wary } t \wedge \text{wary } s \wedge t >_t s$

**abbreviation** *gtwg* :: (*'s*, *'v*) *tm*  $\Rightarrow$  (*'s*, *'v*) *tm*  $\Rightarrow$  *bool* (**infix**  $>_{twg}$  50) **where**

$(>_{twg}) \equiv \lambda t s. \text{ground } t \wedge t >_{tw} s$



**lemma** *ground\_gt\_unary*:  
**assumes** *gr\_t*: *ground t*  
**shows**  $\neg$  *gt\_unary t s*  
*<proof>*

**theorem** *gt\_wf*: *wfP* ( $\lambda s t. t >_{tw} s$ )  
*<proof>*

**end**

**end**

## 7 Knuth–Bendix Orders for Lambda-Free Higher-Order Terms

**theory** *Lambda\_Free\_KBOs*  
**imports** *Lambda\_Free\_KBO\_App* *Lambda\_Free\_KBO\_Basic* *Lambda\_Free\_TKBO\_Coefs*  
**begin**

**locale** *simple\_kbo\_instances*  
**begin**

**definition** *arity\_sym* :: *nat*  $\Rightarrow$  *enat* **where**  
*arity\_sym n* =  $\infty$

**definition** *arity\_var* :: *nat*  $\Rightarrow$  *enat* **where**  
*arity\_var n* =  $\infty$

**definition** *ground\_head\_var* :: *nat*  $\Rightarrow$  *nat set* **where**  
*ground\_head\_var x* = *UNIV*

**definition** *gt\_sym* :: *nat*  $\Rightarrow$  *nat*  $\Rightarrow$  *bool* **where**  
*gt\_sym g f*  $\longleftrightarrow$   $g > f$

**definition**  $\varepsilon$  :: *nat* **where**  
 $\varepsilon$  = 1

**definition**  $\delta$  :: *nat* **where**  
 $\delta$  = 0

**definition** *wt\_sym* :: *nat*  $\Rightarrow$  *nat* **where**  
*wt\_sym n* = 1

**definition** *wt\_sym\_h* :: *nat*  $\Rightarrow$  *hmultiset* **where**  
*wt\_sym\_h n* = 1

**definition** *coef\_sym\_h* :: *nat*  $\Rightarrow$  *nat*  $\Rightarrow$  *hmultiset* **where**  
*coef\_sym\_h n i* = 1

**sublocale** *kbo\_app*: *kbo\_app* *gt\_sym* *wt\_sym*  $\varepsilon$  *len\_lexext*  
*<proof>*

**sublocale** *kbo\_basic*: *kbo\_basic* *gt\_sym* *wt\_sym*  $\varepsilon$   $\lambda f. len\_lexext$  *ground\_head\_var*  
*<proof>*

**sublocale** *kbo\_std*: *kbo\_std* *ground\_head\_var* *gt\_sym*  $\varepsilon$   $\delta$   $\lambda f. len\_lexext$  *arity\_sym* *arity\_var* *wt\_sym*  
*<proof>*

**sublocale** *tkbo\_coefs*: *tkbo\_coefs* *ground\_head\_var* *gt\_sym*  $\varepsilon$   $\delta$   $\lambda f. len\_lexext$  *arity\_sym* *arity\_var*  
*wt\_sym\_h* *coef\_sym\_h*  
*<proof>*

**end**

end