# Knuth-Bendix Order and Its Decidability

Ting Zhang

Microsoft Research Asia

based on joint work with
Henny B. Sipma, Zohar Manna

Logic Colloquium 2008
July 6th, 2008

# Outline

# Outline

1. Introduction
2. Knuth-Bendix Order
3. Decidability of KBO
4. Conclusions and Open Problems

# Importance of Orderings

- Termination Proof. To rank system states:

$$\langle x = 3, y = 2 \rangle > \langle x = 3, y = 1 \rangle$$

# Importance of Orderings

- Termination Proof. To rank system states:

$$\langle x = 3, y = 2\rangle > \langle x = 3, y = 1\rangle$$

☞ Ordered Resolution. To restrict the search space:

$$\frac{A \vee C \qquad \neg A' \vee C'}{(C \vee C')\sigma} \qquad \begin{array}{c} \sigma = \mathbf{mgu}(A, A') \\ (\forall B \in C \cup C')\left[A\sigma \not< B\sigma\right] \end{array}$$

# Importance of Orderings

- Termination Proof. To rank system states:

$$\langle x = 3, y = 2 \rangle > \langle x = 3, y = 1 \rangle$$

- Ordered Resolution. To restrict the search space:

$$\frac{A \vee C \qquad \neg A' \vee C'}{(C \vee C')\sigma} \qquad \begin{array}{c} \sigma = \mathbf{mgu}(A, A') \\ (\forall B \in C \cup C')\left[ A\sigma \not< B\sigma \right] \end{array}$$

- ☞ Ordered Rewriting. To orient commutative equations:

$$\begin{array}{lll} x + y \to y + x & \text{if} & (x + y)\sigma > (y + x)\sigma \\ y + x \to x + y & \text{if} & (x + y)\sigma < (y + x)\sigma \end{array}$$

# Importance of Orderings

- Termination Proof. To rank system states:

$$\langle x = 3, y = 2 \rangle > \langle x = 3, y = 1 \rangle$$

- Ordered Resolution. To restrict the search space:

$$\frac{A \vee C \qquad \neg A' \vee C'}{(C \vee C')\sigma} \qquad \begin{array}{c} \sigma = \mathbf{mgu}(A, A') \\ (\forall B \in C \cup C')\left[A\sigma \not< B\sigma\right] \end{array}$$

- Ordered Rewriting. To orient commutative equations:

$$\begin{array}{lll} x + y \to y + x & \textit{if} & (x+y)\sigma > (y+x)\sigma \\ y + x \to x + y & \textit{if} & (x+y)\sigma < (y+x)\sigma \end{array}$$

Fundamental: Satisfiability Problem of Ordering Constraints

# Beyond Existential Fragments

TOTAL SIMPLIFICATION RULE: [KKR90, CT97]

$$\frac{s \;\rightarrow\; t \;\mid\; c}{s[v]_p \;\rightarrow\; t \;\mid\; (c \wedge c' \wedge s|_p = u)} \quad (u \;\rightarrow\; v \;\mid\; c')$$

# Beyond Existential Fragments

TOTAL SIMPLIFICATION RULE: [KKR90, CT97]

$$\frac{s \;\rightarrow\; t \;\mid\; c}{s[v]_p \;\rightarrow\; t \;\mid\; (c \wedge c' \wedge s|_p = u)} \quad (u \;\rightarrow\; v \;\mid\; c')$$

IT STATES THAT

$s \rightarrow t \mid c$ is simplified (at position $p$) to $s[v]_p \rightarrow t \mid (c \wedge c' \wedge s|_p = u)$ by $u \rightarrow v \mid c'$ provided

$$\mathcal{TA} \models \forall \mathcal{V}(s)\; (\, c \;\rightarrow \exists \mathcal{V}(u)\; (\, c' \;\wedge\; s|_p = u\,)\,) \quad ,$$

which necessarily involves quantifier alternation.

# Widely Used Orderings

| | Syntatic Nature | | Hybrid Nature |
|---|---|---|---|
| | RPO | | KBO |
| | MPO | LPO | |
| syntactic precedence | | | |
| multiset ordering | | | |
| lexicographical ordering | | | |
| numerical ordering | | | |

# Widely Used Orderings

| | Syntatic Nature | | Hybrid Nature |
|---|---|---|---|
| | RPO | | KBO |
| | MPO | LPO | |
| syntactic precedence | ✔ | | |
| multiset ordering | ✔ | | |
| lexicographical ordering | | | |
| numerical ordering | | | |

# Widely Used Orderings

| | Syntatic Nature | | Hybrid Nature |
|---|---|---|---|
| | RPO | | KBO |
| | MPO | LPO | |
| syntactic precedence | ✔ | ✔ | |
| multiset ordering | ✔ | | |
| lexicographical ordering | | ✔ | |
| numerical ordering | | | |

# Widely Used Orderings

<table>
<tr><td rowspan="3"></td><td colspan="2">Syntatic Nature</td><td>Hybrid Nature</td></tr>
<tr><td colspan="2">RPO</td><td rowspan="2">KBO</td></tr>
<tr><td>MPO</td><td>LPO</td></tr>
<tr><td>syntactic precedence</td><td>✔</td><td>✔</td><td>✔</td></tr>
<tr><td>multiset ordering</td><td>✔</td><td></td><td></td></tr>
<tr><td>lexicographical ordering</td><td></td><td>✔</td><td>✔</td></tr>
<tr><td>numerical ordering</td><td></td><td></td><td>✔</td></tr>
</table>

# Decidability Status

| | MPO | LPO | KBO |
|---|---|---|---|
| QFT | | | |
| UQT | | | |
| GQT | | | |

QFT: Quantifier-free Theory.

UQT: Unary Quantified Theory.

GQT: General Quantified Theory.

# Decidability Status

| | MPO | LPO | KBO |
|---|---|---|---|
| QFT | ✔ [JO91] [NRV99] | | |
| UQT | ✔ [NR00] | | |
| GQT | ? | | |

QFT: Quantifier-free Theory.

UQT: Unary Quantified Theory.

GQT: General Quantified Theory.

# Decidability Status

| | MPO | LPO | KBO |
|---|---|---|---|
| QFT | ✔ [JO91] [NRV99] | ✔ [Com90] [Nie93] | |
| UQT | ✔ [NR00] | ✔ [NR00] | |
| GQT | ? | ✖ [Tre92, CT97] | |

QFT: Quantifier-free Theory.

UQT: Unary Quantified Theory.

GQT: General Quantified Theory.

# Decidability Status

| | MPO | LPO | KBO |
|---|---|---|---|
| QFT | ✔ [JO91] [NRV99] | ✔ [Com90] [Nie93] | ✔ [KV00] [KV01] |
| UQT | ✔ [NR00] | ✔ [NR00] | ✔ [KV02] |
| GQT | ? | ✖ [Tre92, CT97] | ? |

QFT: Quantifier-free Theory.

UQT: Unary Quantified Theory.

GQT: General Quantified Theory.

# Decidability Status

| | MPO | LPO | KBO |
|---|---|---|---|
| QFT | ✔ [JO91] [NRV99] | ✔ [Com90] [Nie93] | ✔ [KV00] [KV01] |
| UQT | ✔ [NR00] | ✔ [NR00] | ✔ [KV02] |
| GQT | ? | ✖ [Tre92, CT97] | ✔ [ZSM05] |

QFT: Quantifier-free Theory.

UQT: Unary Quantified Theory.

GQT: General Quantified Theory.

# Our Approach

- ☞ Reduce term constraints to integer constraints.

  T. Zhang, H.B. Sipma, and Z. Manna, *Decision Procedures for Recursive Data Structures with Integer Constraints.* IJCAR'04.

# Our Approach

- Reduce term constraints to integer constraints.

  T. Zhang, H.B. Sipma, and Z. Manna, *Decision Procedures for Recursive Data Structures with Integer Constraints.* IJCAR'04.

☞ Reduce term quantifiers to integer quantifiers.

  T. Zhang, H.B. Sipma and Z. Manna, *Term Algebras with Length Function and Bounded Quantifier Alternation.* TPHOLs'04.

# Our Approach

Reduction from term domain to integer domain!

- Reduce term constraints to integer constraints.

  T. Zhang, H.B. Sipma, and Z. Manna, *Decision Procedures for Recursive Data Structures with Integer Constraints.* IJCAR'04.

- Reduce term quantifiers to integer quantifiers.

  T. Zhang, H.B. Sipma and Z. Manna, *Term Algebras with Length Function and Bounded Quantifier Alternation.* TPHOLs'04.

# Outline

1. Introduction
2. **Knuth-Bendix Order**
3. Decidability of KBO
4. Conclusions and Open Problems

# Term Algebras

A term algebra $\mathcal{TA} : \langle TA; \mathcal{C}, \mathcal{A}, \mathcal{S}, \mathcal{T} \rangle$ consists of

- $TA$: The term domain.

# Term Algebras

A term algebra $\mathcal{TA} : \langle TA; \mathcal{C}, \mathcal{A}, \mathcal{S}, \mathcal{T} \rangle$ consists of

- $TA$: The term domain.
- $\mathcal{C}$: set of constructors: $\alpha, \beta, \gamma, \ldots$.

# Term Algebras

A term algebra $\mathcal{TA}$ : $\langle TA; \mathcal{C}, \mathcal{A}, \mathcal{S}, \mathcal{T} \rangle$ consists of

- $TA$: The term domain.
- $\mathcal{C}$: set of constructors: $\alpha, \beta, \gamma, \ldots$.
- ☞ $\mathcal{A}$ : set of constants: $a, b, c, \ldots$. $\mathcal{A} \subseteq \mathcal{C}$.

# Term Algebras

A term algebra $\mathcal{TA}$ : $\langle TA; \mathcal{C}, \mathcal{A}, \mathcal{S}, \mathcal{T} \rangle$ consists of

- $TA$: The term domain.
- $\mathcal{C}$: set of constructors: $\alpha, \beta, \gamma, \ldots$.
- $\mathcal{A}$ : set of constants: $a, b, c, \ldots$. $\mathcal{A} \subseteq \mathcal{C}$.
- ☞ $\mathcal{S}$: set of selectors: $\alpha = (s_1^\alpha, \ldots, s_k^\alpha)$.

# Term Algebras

A term algebra $\mathcal{TA} : \langle TA; \mathcal{C}, \mathcal{A}, \mathcal{S}, \mathcal{T} \rangle$ consists of

- $TA$: The term domain.
- $\mathcal{C}$: set of constructors: $\alpha, \beta, \gamma, \ldots$.
- $\mathcal{A}$ : set of constants: $a, b, c, \ldots$. $\mathcal{A} \subseteq \mathcal{C}$.
- $\mathcal{S}$: set of selectors: $\alpha = (s_1^\alpha, \ldots, s_k^\alpha)$.
- ☞ $\mathcal{T}$: set of testers. $Is_\alpha$ for $\alpha \in \mathcal{C}$.

# Term Algebras

A term algebra $\mathcal{TA}: \langle TA; \mathcal{C}, \mathcal{A}, \mathcal{S}, \mathcal{T} \rangle$ consists of

- $TA$: The term domain.
- $\mathcal{C}$: set of constructors: $\alpha, \beta, \gamma, \ldots$.
- $\mathcal{A}$ : set of constants: $a, b, c, \ldots$. $\mathcal{A} \subseteq \mathcal{C}$.
- $\mathcal{S}$: set of selectors: $\alpha = (s_1^\alpha, \ldots, s_k^\alpha)$.
- $\mathcal{T}$: set of testers. $Is_\alpha$ for $\alpha \in \mathcal{C}$.

# Term Algebras

A term algebra $\mathcal{TA}$ : $\langle TA; \mathcal{C}, \mathcal{A}, \mathcal{S}, \mathcal{T} \rangle$ consists of

- $TA$: The term domain.
- $\mathcal{C}$: set of constructors: $\alpha, \beta, \gamma, \ldots$.
- $\mathcal{A}$ : set of constants: $a, b, c, \ldots$. $\mathcal{A} \subseteq \mathcal{C}$.
- $\mathcal{S}$: set of selectors: $\alpha = (s_1^\alpha, \ldots, s_k^\alpha)$.
- $\mathcal{T}$: set of testers. $Is_\alpha$ for $\alpha \in \mathcal{C}$.

**NOTE THAT**

Each element in $TA$ is uniquely generated by constructors.

# LISP List

SIGNATURE

$$\langle\ list; \{cons, nil\}; \{nil\}; \{car, cdr\}; \{Is_{nil}, Is_{cons}\}\ \rangle$$

# LISP List

**SIGNATURE**

$$\langle\ list; \{cons, nil\}; \{nil\}; \{car, cdr\}; \{Is_{nil}, Is_{cons}\}\ \rangle$$

**AXIOMS**

$$\begin{aligned} Is_{nil}(x) &\leftrightarrow \neg Is_{cons}(x) \\ x &= car(cons(x, y)) \\ y &= cdr(cons(x, y)) \\ Is_{nil}(x) &\leftrightarrow \{car, cdr\}^{+}(x) = x \\ Is_{cons}(x) &\leftrightarrow cons(car(x), cdr(x)) = x \end{aligned}$$

# Knuth-Bendix Order

A Knuth-Bendix order (KBO) $<^{kb}$ is parametrically defined with

- ☞ $w : TA \to \mathbb{N}$: a weight function such that

$$w(\alpha(t_1, \ldots, t_k)) = w(\alpha) + \sum_{i=1}^{k} w(t_i).$$

# Knuth-Bendix Order

A Knuth-Bendix order (KBO) $\prec^{kb}$ is parametrically defined with

- $w : TA \rightarrow \mathbb{N}$: a weight function such that

$$w(\alpha(t_1, \ldots, t_k)) = w(\alpha) + \sum_{i=1}^{k} w(t_i).$$

☞ $\prec^{\sigma}$: a linear precedence order on $C$ such that

$$\alpha_1 \prec^{\sigma} \alpha_2 \prec^{\sigma} \ldots \prec^{\sigma} \alpha_{|C|}.$$

# Knuth-Bendix Order

A KBO $\prec^{kb}$ is recursively defined (with respect to $w$ and $\prec^{\sigma}$) such that $u \prec^{kb} v$ if one of the following conditions holds.

- ☞ $w(u) < w(v)$

# Knuth-Bendix Order

A KBO $\prec^{kb}$ is recursively defined (with respect to $w$ and $\prec^{\sigma}$) such that $u \prec^{kb} v$ if one of the following conditions holds.

- $w(u) < w(v)$
- ☞ $w(u) = w(v)$ and $type(u) \prec^{\sigma} type(v)$

# Knuth-Bendix Order

A KBO $\prec^{kb}$ is recursively defined (with respect to $w$ and $\prec^{\sigma}$) such that $u \prec^{kb} v$ if one of the following conditions holds.

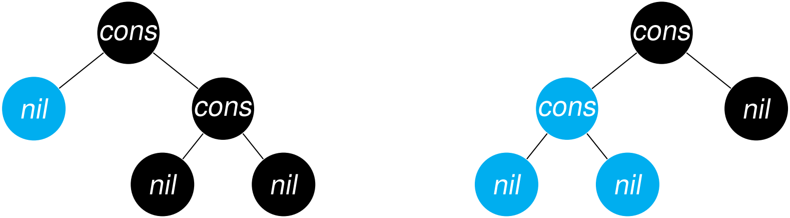
- $w(u) < w(v)$
- $w(u) = w(v)$ and $type(u) \prec^{\sigma} type(v)$
- ☞ $w(u) = w(v)$, $u \equiv \alpha(u_1, \ldots, u_k)$, $v \equiv \alpha(v_1, \ldots, v_k)$, *and*

$$\exists i \left( 1 \leq i \leq k \ \wedge \ u_i \prec^{kb} v_i \ \wedge \ \forall j \ ( 1 \leq j < i \ \rightarrow \ u_j = v_j ) \right).$$

# Example: Knuth-Bendix Order

Consider the KBO on LISP list structure parameterized with

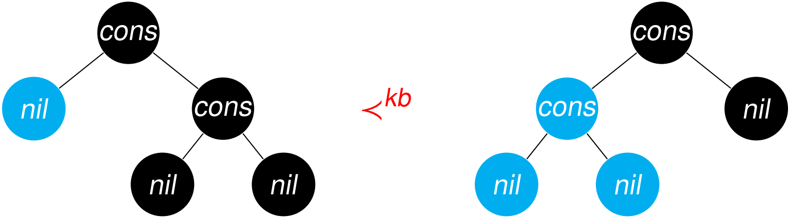
$$w(cons) = w(nil) = 1 \quad and \quad nil <^{\sigma} cons.$$



# Example: Knuth-Bendix Order

Consider the KBO on LISP list structure parameterized with

$$w(cons) = w(nil) = 1 \quad and \quad nil <^{\sigma} cons.$$



# Outline

1. Introduction
2. Knuth-Bendix Order
3. **Decidability of KBO**
4. Conclusions and Open Problems

# Quantifier Elimination

☞ Suffices to eliminate ∃-quantifiers from primitive formulas

$$\exists \bar{\boldsymbol{x}} \; ( A_1(\bar{\boldsymbol{x}}) \wedge \ldots \wedge A_n(\bar{\boldsymbol{x}}) ) \, ,$$

where $A_i(\bar{\boldsymbol{x}})$ $(1 \leq i \leq n)$ are literals.

# Quantifier Elimination

- Suffices to eliminate ∃-quantifiers from primitive formulas

$$\exists \bar{\boldsymbol{x}} \; (\, A_1(\bar{\boldsymbol{x}}) \land \ldots \land A_n(\bar{\boldsymbol{x}}) \,)\,,$$

where $A_i(\bar{\boldsymbol{x}})$ $(1 \leq i \leq n)$ are literals.

☞ Suffices to assume $A_i \not\equiv x = t$ if $x \notin t$, because

$$\exists x \, (x = t \land \varphi(x, \bar{\boldsymbol{y}})) \;\leftrightarrow\; \varphi(t, \bar{\boldsymbol{y}}).$$

# Selector Language

For simplicity, we only use selectors and testers in our language.

# Selector Language

For simplicity, we only use selectors and testers in our language.

NOTATION

☞ The depth of $x$ in a selector term $t$ is the number of selectors in $t$. For example, the depth of $x$ in $s_1(\dots(s_n(x)\dots))$ is $n$.

# Selector Language

For simplicity, we only use selectors and testers in our language.

NOTATION

- The depth of $x$ in a selector term $t$ is the number of selectors in $t$. For example, the depth of $x$ in $s_1(\ldots(s_n(x)\ldots))$ is $n$.
- ☞ By $depth_\varphi(x)$, we mean the maximum depth of $x$ in $\varphi$.

# Selector Language

For simplicity, we only use selectors and testers in our language.

**NOTATION**

- The depth of $x$ in a selector term $t$ is the number of selectors in $t$. For example, the depth of $x$ in $s_1(\ldots(s_n(x)\ldots))$ is $n$.
- By $depth_\varphi(x)$, we mean the maximum depth of $x$ in $\varphi$.

☞ Formulas are assumed to be type complete, i.e., the type of every term is asserted by a tester literal.

# Selector Language

For simplicity, we only use selectors and testers in our language.

NOTATION

- The depth of $x$ in a selector term $t$ is the number of selectors in $t$. For example, the depth of $x$ in $s_1(\ldots(s_n(x)\ldots))$ is $n$.
- By $depth_{\varphi}(x)$, we mean the maximum depth of $x$ in $\varphi$.
- Formulas are assumed to be type complete, i.e., the type of every term is asserted by a tester literal.

☞ Selector terms are assumed to be proper. For example, $car(x) \neq cdr(x)$ abbreviates $car(x) \neq cdr(x) \ \wedge \ Is_{cons}(x)$.

# Main Idea

☞ Solved Form. Eliminating $\exists x$ from $(\exists x)\varphi(x, \bar{\boldsymbol{y}})$ is easy once

$\varphi(x, \bar{\boldsymbol{y}})$ *is solved in* $x$.

# Main Idea

- Solved Form. Eliminating $\exists x$ from $(\exists x)\varphi(x, \bar{\mathbf{y}})$ is easy once

$$\varphi(x, \bar{\mathbf{y}}) \textit{ is solved in } x.$$

☞ Depth Reduction. Transforming $\varphi(x, \bar{\mathbf{y}})$ into a solved form amounts to peeling off selectors in front of $x$, since

$$\varphi(x, \bar{\mathbf{y}}) \textit{ solved in } x \textit{ if and only if } \mathit{depth}_{\varphi}(x) = 0.$$

# Solved Form

☞ $\varphi(x, \bar{\mathbf{y}})$ is solved in $x$ if it is in the form

$$\bigwedge_{i \leq m} u_i \prec^{kb} x \ \wedge \ \bigwedge_{j \leq n} x \prec^{kb} v_j \ \wedge \ \varphi'(\bar{\mathbf{y}}),$$

where $x$ does not appear in $u_i$, $v_i$ and $\varphi'$.

# Solved Form

- $\varphi(x, \bar{\mathbf{y}})$ is solved in $x$ if it is in the form

$$\bigwedge_{i \le m} u_i \prec^{kb} x \;\wedge\; \bigwedge_{j \le n} x \prec^{kb} v_j \;\wedge\; \varphi'(\bar{\mathbf{y}}),$$

where $x$ does not appear in $u_i$, $v_i$ and $\varphi'$.

☞ If $\varphi(x, \bar{\mathbf{y}})$ is solved in $x$, then $(\exists x)\, \varphi(x, \bar{\mathbf{y}})$ simplifies to

$$\varphi'(\bar{\mathbf{y}}) \;\wedge\; \bigwedge_{i \le m, j \le n} u_i \prec_2^{kb} v_j$$

where $x \prec_n^{kb} y$, called gap order, states there is an increasing chain from $x$ to $y$ of length at least $n$.

# Depth Reduction: Case 1

All occurrences of $x$ have depth greater than 0.

In this case, $\exists x \varphi(x, \bar{\boldsymbol{y}})$ must be in the form

$$\exists x \varphi'(s_1^{\alpha}(x), \ldots, s_k^{\alpha}(x), \bar{\boldsymbol{y}}),$$

which can be rewritten to

$$\exists x_1, \ldots, \exists x_k \varphi'(x_1, \ldots, x_k, \bar{\boldsymbol{y}}).$$

# Depth Reduction: Case 2

Some occurrences of $x$ have depth 0 and some do not.

Decompose 0-depth occurrences of $x$ in terms of

$$s_1^\alpha(x), \ldots, s_k^\alpha(x),$$

which amounts to expressing $x \prec_n^{kb} t$ and $t \prec_n^{kb} x$ using

$$s_1^\alpha(x), \ldots, s_k^\alpha(x).$$

# Depth Reduction: Case 2

Some occurrences of $x$ have depth 0 and some do not.

Decompose 0-depth occurrences of $x$ in terms of

$$s_1^{\alpha}(x), \ldots, s_k^{\alpha}(x),$$

which amounts to expressing $x \prec_n^{kb} t$ and $t \prec_n^{kb} x$ using

$$s_1^{\alpha}(x), \ldots, s_k^{\alpha}(x).$$

Then apply the reduction as in Case 1!

# Language Extensions

- Decompose $\prec^{kb}$ into three disjoint suborders $\prec^{w}$, $\prec^{p}$ and $\prec^{l}$.

# Language Extensions

- Decompose $\prec^{kb}$ into three disjoint suborders $\prec^{w}$, $\prec^{p}$ and $\prec^{l}$.
- ☞ Extend $\prec^{w}$, $\prec^{p}$ and $\prec^{l}$ to $\prec^{w}_{n}$, $\prec^{p}_{n}$ and $\prec^{l}_{n}$, respectively.

# Language Extensions

- Decompose $\prec^{kb}$ into three disjoint suborders $\prec^{w}$, $\prec^{p}$ and $\prec^{l}$.
- Extend $\prec^{w}$, $\prec^{p}$ and $\prec^{l}$ to $\prec^{w}_{n}$, $\prec^{p}_{n}$ and $\prec^{l}_{n}$, respectively.
- ☞ Add boundary functions to delineate gap orders.

# Language Extensions

- Decompose $\prec^{kb}$ into three disjoint suborders $\prec^{w}$, $\prec^{p}$ and $\prec^{l}$.
- Extend $\prec^{w}$, $\prec^{p}$ and $\prec^{l}$ to $\prec^{w}_{n}$, $\prec^{p}_{n}$ and $\prec^{l}_{n}$, respectively.
- Add boundary functions to delineate gap orders.

☞ Add Presburger arithmetic explicitly to represent the weight function.

# Language Extensions

- Decompose $\prec^{kb}$ into three disjoint suborders $\prec^{w}$, $\prec^{p}$ and $\prec^{l}$.
- Extend $\prec^{w}$, $\prec^{p}$ and $\prec^{l}$ to $\prec^{w}_{n}$, $\prec^{p}_{n}$ and $\prec^{l}_{n}$, respectively.
- Add boundary functions to delineate gap orders.
- Add Presburger arithmetic explicitly to represent the weight function.

☞ Extend all aforementioned notions to tuples of terms.

# Suborders

**WEIGHT ORDER**

$$u \prec^w v \quad \overset{def}{=\!=} \quad w(u) < w(v)$$

# Suborders

**WEIGHT ORDER**

$$u \prec^{w} v \quad \stackrel{def}{=\!=} \quad w(u) < w(v)$$

**PRECEDENCE ORDER**

$$u \prec^{p} v \quad \stackrel{def}{=\!=} \quad w(u) = w(s) \;\; \& \;\; type(u) \prec^{\sigma} type(v)$$

# Suborders

**WEIGHT ORDER**

$$u \prec^{w} v \quad \stackrel{def}{=\!=} \quad w(u) < w(v)$$

**PRECEDENCE ORDER**

$$u \prec^{p} v \quad \stackrel{def}{=\!=} \quad w(u) = w(s) \;\; \& \;\; type(u) \prec^{\sigma} type(v)$$

**LEXICOGRAPHICAL ORDER**

$$u \prec^{l} v \quad \stackrel{def}{=\!=} \quad w(u) = w(v) \;\; \& \;\; type(u) = type(v) \;\; \& \;\; u \prec^{kb} v$$

# Gap Orders

**KB GAP ORDER**

$$u \prec_n^{kb} v \quad \stackrel{def}{=\!=} \quad (\exists u_1 \cdots \exists u_n)\left(u \prec^{kb} u_1 \prec^{kb} \cdots \prec^{kb} u_n \preceq^{kb} v\right)$$

# Gap Orders

**KB GAP ORDER**

$$u \prec_n^{kb} v \quad \stackrel{def}{=\!=} \quad (\exists u_1 \cdots \exists u_n)\left(u \prec^{kb} u_1 \prec^{kb} \cdots \prec^{kb} u_n \preceq^{kb} v\right)$$

**WEIGHT GAP ORDER**

$$u \prec_n^{w} v \quad \stackrel{def}{=\!=} \quad u \prec_n^{kb} v \ \wedge \ u \prec^{w} v$$

# Gap Orders

**KB GAP ORDER**

$$u \prec_n^{kb} v \quad \overset{def}{=\!=} \quad (\exists u_1 \cdots \exists u_n)\Big(u \prec^{kb} u_1 \prec^{kb} \cdots \prec^{kb} u_n \preceq^{kb} v\Big)$$

**WEIGHT GAP ORDER**

$$u \prec_n^{w} v \quad \overset{def}{=\!=} \quad u \prec_n^{kb} v \ \wedge \ u \prec^{w} v$$

**PRECEDENCE GAP ORDER**

$$u \prec_n^{p} v \quad \overset{def}{=\!=} \quad u \prec_n^{kb} v \ \wedge \ u \prec^{p} v$$

# Gap Orders

**KB GAP ORDER**

$$u <_n^{kb} v \quad \stackrel{def}{=\!=} \quad (\exists u_1 \cdots \exists u_n)\left(u <^{kb} u_1 <^{kb} \cdots <^{kb} u_n \leq^{kb} v\right)$$

**WEIGHT GAP ORDER**

$$u <_n^{w} v \quad \stackrel{def}{=\!=} \quad u <_n^{kb} v \ \wedge \ u <^{w} v$$

**PRECEDENCE GAP ORDER**

$$u <_n^{p} v \quad \stackrel{def}{=\!=} \quad u <_n^{kb} v \ \wedge \ u <^{p} v$$

**LEXICOGRAPHICAL GAP ORDER**

$$u <_n^{l} v \quad \stackrel{def}{=\!=} \quad u <_n^{kb} v \ \wedge \ u <^{l} v$$

# Boundary Functions

$0^w, 1^w : \mathbb{N} \to TA$; $0^p, 1^p : \mathbb{N}^2 \to TA$ such that

$0^w(n)$ : *the smallest term of weight $n$*

$0^p(n, p)$ : *the smallest term of weight $n$ and type $\alpha_p$*

$1^w(n)$ : *the largest term of weight $n$*

$1^p(n, p)$ : *the largest term of weight $n$ and type $\alpha_p$*

# Boundary Functions

$0^w, 1^w : \mathbb{N} \to TA$; $0^p, 1^p : \mathbb{N}^2 \to TA$ such that

$0^w(n)$ : *the* smallest *term of weight* $n$

$0^p(n,p)$ : *the* smallest *term of weight* $n$ *and type* $\alpha_p$

$1^w(n)$ : *the* largest *term of weight* $n$

$1^p(n,p)$ : *the* largest *term of weight* $n$ *and type* $\alpha_p$

**EXAMPLE**

$$u <_5^w v \leftrightarrow \bigvee_{n_1+n_2+n_3=5} u <_{n_1}^{pl} 1^w_{(u^w)} <_{n_2}^w 0^w_{(v^w)} <_{n_3}^{pl} v$$

# Counting Constraints

$CNT_n(x)$ states that

*there are at least $n + 1$ distinct $TA$-terms of weight $x$.*

In particular, $CNT_0(x)$ (or $Tree(x)$) states that

*$x$ is a legitimate weight of a term.*

# Counting Constraints

$CNT_n(x)$ states that

*there are at least $n+1$ distinct $TA$-terms of weight $x$.*

In particular, $CNT_0(x)$ (or $Tree(x)$) states that

*$x$ is a legitimate weight of a term.*

**EXAMPLE**

$$0^w_{(x)} <^{pl}_n 1^w_{(x)} \leftrightarrow CNT_n(x)$$

# Counting Constraints

$CNT_n(x)$ states that

*there are at least $n+1$ distinct TA-terms of weight $x$.*

In particular, $CNT_0(x)$ (or $Tree(x)$) states that

*$x$ is a legitimate weight of a term.*

**EXAMPLE**

$$0^w_{(x)} <^{pl}_n 1^w_{(x)} \leftrightarrow CNT_n(x)$$

**NOTE THAT**

$CNT_n(x)$ is expressible in Presburger arithmetic.

# Knuth-Bendix Order with Presburger Arithmetic

$$\mathcal{KBO}^+ = \langle\, \mathcal{TA}, \mathcal{PA}, (.)^w, \prec_n^\sharp, 0^*(\ldots), 1^*(\ldots) \,\rangle$$

where $n \in \mathbb{N}$, $\sharp \in \{w, p, l\}$, $* \in \{w, p\}$,

$$\begin{aligned} (.)^w &: \textit{weight function}, \\ \prec_n^\sharp &: \textit{gap orders}, \\ 0^*(\ldots), 1^*(\ldots) &: \textit{boundary functions} \end{aligned}$$

# Knuth-Bendix Order with Presburger Arithmetic

$$\mathcal{KBO}^{+} = \langle\, \mathcal{TA}, \mathcal{PA}, (.)^{w}, \prec_{n}^{\sharp}, 0^{*}(\ldots), 1^{*}(\ldots) \,\rangle$$

where $n \in \mathbb{N}$, $\sharp \in \{w, p, l\}$, $* \in \{w, p\}$,

$$\begin{aligned} (.)^{w} &: \textit{weight function}, \\ \prec_{n}^{\sharp} &: \textit{gap orders}, \\ 0^{*}(...), 1^{*}(...) &: \textit{boundary functions} \end{aligned}$$

**EXAMPLE**

$$\exists x : TA \left( 0^{w}_{(x^{w})} \prec_{2}^{l} x \prec_{3}^{l} 1^{w}_{(x^{w})} \right)$$

# Quantifier Elimination for Knuth-Bendix Order

**INPUT:** $(\exists \bar{\boldsymbol{x}})\ \varphi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$

**while** $\bar{\boldsymbol{x}} \neq \emptyset$ **do**

**if** $(\forall x \in \bar{\boldsymbol{x}})\ depth_{\varphi}(x) > 0$ **then**

Depth Reduction: [ VARIABLE SELECTION / DECOMPOSITION / SIMPLIFICATION ]

**else** $\{(\exists x \in \bar{\boldsymbol{x}})\ depth_{\varphi}(x) = 0\}$

Elimination

**end if**

**end while**

# Variable Selection

Select a variable $x \in \bar{\boldsymbol{x}}$ such that $s_i^\alpha(x)$ appears in $\varphi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$.

# Variable Selection

Select a variable $x \in \bar{\boldsymbol{x}}$ such that $s_i^{\alpha}(x)$ appears in $\varphi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$.

**NOTE THAT**

The selection is done in depth-first manner; we always choose variables generated in the previous round.

# Decomposition

☞ Rewrite $(\exists \bar{\boldsymbol{x}})\ \varphi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ to

$$\exists x_1 \dots \exists x_k \exists \bar{\boldsymbol{x}} \left( \bigwedge_{1 \le i \le k} s_i^{\alpha}(x) = x_i \ \wedge\ \varphi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \right).$$

# Decomposition

- Rewrite $(\exists \bar{\boldsymbol{x}})\ \varphi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ to

$$\exists x_1 \ldots \exists x_k \exists \bar{\boldsymbol{x}} \left( \bigwedge_{1 \le i \le k} s_i^{\alpha}(x) = x_i \ \wedge\ \varphi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \right).$$

- Rewrite $x \prec_n^{\sharp} t$ and $t \prec_n^{\sharp} x$ to quantifier-free formulas where $x$ only occurs in $s_1^{\alpha}(x), \ldots, s_k^{\alpha}(x)$.

# Decomposition

☞ Rewrite $(\exists \bar{\mathbf{x}})\ \varphi(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ to

$$\exists x_1 \dots \exists x_k \exists \bar{\mathbf{x}} \left( \bigwedge_{1 \le i \le k} s_i^{\alpha}(x) = x_i \ \wedge\ \varphi(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \right).$$

☞ Rewrite $x <_n^{\sharp} t$ and $t <_n^{\sharp} x$ to quantifier-free formulas where $x$ only occurs in $s_1^{\alpha}(x), \dots, s_k^{\alpha}(x)$.

RESULTING IN

$$\exists x_1 \dots \exists x_k \exists \bar{\mathbf{x}} \left( \bigwedge_{1 \le i \le k} s_i^{\alpha}(x) = x_i \ \wedge\ \varphi'(s_1^{\alpha}(x), \dots, s_k^{\alpha}(x), (\bar{\mathbf{x}} \setminus x), \bar{\mathbf{y}}) \right).$$

# Simplification

**NOW**

$$\exists x_1 \ldots \exists x_k \exists \bar{\boldsymbol{x}} \left( \bigwedge_{1 \le i \le k} s_i^\alpha(x) = x_i \;\wedge\; \varphi'(s_1^\alpha(x), \ldots, s_k^\alpha(x), (\bar{\boldsymbol{x}} \setminus x), \bar{\boldsymbol{y}}) \right).$$

# Simplification

NOW

$$\exists x_1 \ldots \exists x_k \exists \bar{\mathbf{x}} \left( \bigwedge_{1 \le i \le k} s_i^\alpha(x) = x_i \;\wedge\; \varphi'(s_1^\alpha(x), \ldots, s_k^\alpha(x), (\bar{\mathbf{x}} \setminus x), \bar{\mathbf{y}}) \right).$$

☞ Replace $s_i^\alpha(x)$ by $x_i$ in $\varphi'$.

# Simplification

NOW

$$\exists x_1 \ldots \exists x_k \exists \bar{\boldsymbol{x}} \left( \bigwedge_{1 \le i \le k} s_i^\alpha(x) = x_i \ \wedge \ \varphi'(s_1^\alpha(x), \ldots, s_k^\alpha(x), (\bar{\boldsymbol{x}} \setminus x), \bar{\boldsymbol{y}}) \right).$$

- Replace $s_i^\alpha(x)$ by $x_i$ in $\varphi'$.
- ☞ Remove $\bigwedge_{1 \le i \le k} s_i^\alpha(x) = x_i$ from the matrix.

# Simplification

NOW

$$\exists x_1 \ldots \exists x_k \exists \bar{\mathbf{x}} \left( \bigwedge_{1 \le i \le k} s_i^\alpha(x) = x_i \;\wedge\; \varphi'(s_1^\alpha(x), \ldots, s_k^\alpha(x), (\bar{\mathbf{x}} \setminus x), \bar{\mathbf{y}}) \right).$$

- Replace $s_i^\alpha(x)$ by $x_i$ in $\varphi'$.
- Remove $\bigwedge_{1 \le i \le k} s_i^\alpha(x) = x_i$ from the matrix.
- ☞ Remove $\exists x$ from the prenex.

# Simplification

**NOW**

$$\exists x_1 \ldots \exists x_k \exists \bar{\mathbf{x}} \left( \bigwedge_{1 \le i \le k} s_i^{\alpha}(x) = x_i \ \wedge \ \varphi'(s_1^{\alpha}(x), \ldots, s_k^{\alpha}(x), (\bar{\mathbf{x}} \setminus x), \bar{\mathbf{y}}) \right).$$

- Replace $s_i^{\alpha}(x)$ by $x_i$ in $\varphi'$.
- Remove $\bigwedge_{1 \le i \le k} s_i^{\alpha}(x) = x_i$ from the matrix.
- Remove $\exists x$ from the prenex.

**RESULTING IN**

$$\exists x_1 \ldots \exists x_k \exists (\bar{\mathbf{x}} \setminus x) \left( \varphi'((\bar{\mathbf{x}} \setminus x), x_1, \ldots, x_k, \bar{\mathbf{y}}) \right).$$

# Elimination

NOW

$$\exists x \left( \bigwedge_{i \le m} u_i \prec^{kb} x \ \wedge \ \bigwedge_{j \le n} x \prec^{kb} v_j \ \wedge \ \varphi'(\bar{\boldsymbol{y}}) \right),$$

# Elimination

NOW

$$\exists x \left( \bigwedge_{i \le m} u_i \prec^{kb} x \ \wedge \ \bigwedge_{j \le n} x \prec^{kb} v_j \ \wedge \ \varphi'(\bar{\mathbf{y}}) \right),$$

which simplifies to

$$\begin{array}{ll} & u_{i'} \prec_2^{kb} v_{j'} \ \wedge \ \varphi'(\bar{\mathbf{y}}) \\ \wedge & \text{“}u_{i'} \text{ is the greatest of } \{u_i \mid i \le m\}\text{”} \\ \wedge & \text{“}v_{j'} \text{ is the smallest of } \{v_j \mid j \le n\}\text{”}. \end{array}$$

# Technical Tricks

- Elimination of Equalities.

$$\exists x \left( x = 0^{w}_{((car(x))^{w}+5)} \;\wedge\; car(x) \prec^{p}_{4} cdr(x) \right).$$

# Technical Tricks

- Elimination of Equalities.

$$\exists x \left( x = 0^{w}_{((car(x))^{w}+5)} \land car(x) <^{p}_{4} cdr(x) \right).$$

- ☞ Simplification of Selector Terms.

$$car(0^{w}_{((car(x))^{w})}).$$

# Technical Tricks

- Elimination of Equalities.

$$\exists x \left( \, x = 0^{w}_{((car(x))^{w}+5)} \ \wedge \ car(x) <^{p}_{4} cdr(x) \, \right).$$

- Simplification of Selector Terms.

$$car(0^{w}_{((car(x))^{w})}).$$

☞ Elimination of Negations.

$$\neg\big(car(x) <^{w}_{3} cdr(x)\big).$$

# Technical Tricks

- Elimination of Equalities.

$$\exists x \left( x = 0^{w}_{((car(x))^{w}+5)} \ \wedge \ car(x) <^{p}_{4} cdr(x) \right).$$

- Simplification of Selector Terms.

$$car(0^{w}_{((car(x))^{w})}).$$

- Elimination of Negations.

$$\neg\big(car(x) <^{w}_{3} cdr(x)\big).$$

☞ Termination.

# Elimination of Equalities

**EXAMPLE**

$$\exists x \left( x = 0^{w}_{((car(x))^{w}+5)} \ \wedge \ car(x) <^{p}_{4} cdr(x) \right)$$

# Elimination of Equalities

EXAMPLE

$$\exists x \left( x = 0^{w}_{((car(x))^{w}+5)} \ \wedge \ car(x) <^{p}_{4} cdr(x) \right)$$

☞ Reverse Substitution $\Rightarrow$

$$\exists x \left( x = 0^{w}_{((car(x))^{w}+5)} \ \wedge \ car(0^{w}_{((car(x))^{w}+5)}) <^{p}_{4} cdr(0^{w}_{((car(x))^{w}+5)}) \right)$$

# Elimination of Equalities

**CONTINUE WITH**

$$\exists x \big(\, x = 0^{w}_{((car(x))^{w}+5)} \;\wedge\; car(0^{w}_{((car(x))^{w}+5)}) \prec^{p}_{4} cdr(0^{w}_{((car(x))^{w}+5)}) \,\big)$$

# Elimination of Equalities

**CONTINUE WITH**

$$\exists x \left( \, x = 0^{w}_{((car(x))^{w}+5)} \;\wedge\; car(0^{w}_{((car(x))^{w}+5)}) \prec^{p}_{4} cdr(0^{w}_{((car(x))^{w}+5)}) \right)$$

☞ Reduction to Integer Quantifiers $\Rightarrow$

$$\begin{array}{l} \exists (car(x))^{w} \\ \exists (cdr(x))^{w} \end{array} \left( \begin{array}{l} Tree((car(x))^{w}+5) \;\wedge\; Tree((cdr(x))^{w}+5) \\ \quad \wedge\; (\alpha)^{w} + (car(x))^{w} + (cdr(x))^{w} = (car(x))^{w} + 5 \\ \quad \wedge\; car(0^{w}_{((car(x))^{w}+5)}) \prec^{p}_{4} cdr(0^{w}_{((car(x))^{w}+5)}) \end{array} \right)$$

# Elimination of Equalities

**CONTINUE WITH**

$$\exists (car(x))^w \, \exists (cdr(x))^w \left( \begin{array}{l} Tree((car(x))^w + 5) \ \wedge \ Tree((cdr(x))^w + 5) \\ \ \wedge \ (\alpha)^w + (car(x))^w + (cdr(x))^w = (car(x))^w + 5 \\ \ \wedge \ car(0^w_{((car(x))^w+5)}) \prec^p_4 cdr(0^w_{((car(x))^w+5)}) \end{array} \right)$$

# Elimination of Equalities

CONTINUE WITH

$$\exists (car(x))^w \, \exists (cdr(x))^w \left( \begin{array}{l} Tree((car(x))^w + 5) \;\wedge\; Tree((cdr(x))^w + 5) \\ \;\wedge\; (\alpha)^w + (car(x))^w + (cdr(x))^w = (car(x))^w + 5 \\ \;\wedge\; car(0^w_{((car(x))^w+5)}) <^p_4 cdr(0^w_{((car(x))^w+5)}) \end{array} \right)$$

☞ Renaming $\Rightarrow$

$$\exists z \exists y \left( \begin{array}{l} Tree(z) \;\wedge\; Tree(y) \\ \;\wedge\; (\alpha)^w + z + y = z + 5 \\ \;\wedge\; car(0^w_{(z)}) <^p_4 cdr(0^w_{(z)}) \end{array} \right)$$

# Simplification of Selector Terms

**EXAMPLE**

$$car(0^{w}_{((car(x))^{w})})$$

# Simplification of Selector Terms

EXAMPLE

$$car(0^{w}_{((car(x))^{w})})$$

which simplifies to

$$0^{w}_{f_{car}((car(x))^{w})}$$

where $f_{car}(\cdot)$ is an integer function expressible in Presburger arithmetic.

# Elimination of Negations

EXAMPLE

$$\neg\big(car(x) <^w_3 cdr(x)\big)$$

simplifies to

# Elimination of Negations

**EXAMPLE**

$$\neg\big(car(x) <_3^w cdr(x)\big)$$

simplifies to

$$\begin{array}{ll} & cdr(x) <_1^w car(x) \\ \vee & (cdr(x))^w = (car(x))^w \\ \vee & car(x) \leqq_1^w cdr(x) \\ \vee & car(x) \leqq_2^w cdr(x). \end{array}$$

# Termination

Termination is subtle as many complexity measures increase.

# Termination

Termination is subtle as many complexity measures increase.

☞ Depth reduction increases the depth of other variables.

# Termination

Termination is subtle as many complexity measures increase.

☞ Depth reduction increases the depth of other variables.

For example, $x \neq t$ becomes

$$\bigvee_{1 \leq i \leq k} s_i^{\alpha}(t) \neq x_i \;\vee\; \neg Is_{\alpha}(t).$$

# Termination

- ☞ Depth reduction introduces more existential quantifiers.

# Termination

☞ Depth reduction introduces more existential quantifiers.

For example, $(\exists \bar{\boldsymbol{x}})\, \varphi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ becomes

$$\exists x_1 \ldots \exists x_k \exists \bar{\boldsymbol{x}} \left( ls_\alpha(x) \;\wedge \bigwedge_{1 \le i \le k} s_i^\alpha(x) = x_i \;\wedge\; \varphi(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \right).$$

# Termination

☞ Depth reduction introduces more order literals.

# Termination

☞ Depth reduction introduces more order literals.

For example, $u \prec_5^w v$ becomes

$$\bigvee_{n_1+n_2+n_3=5} u \prec_{n_1}^{pl} 1_{(u^w)}^w \prec_{n_2}^w 0_{(v^w)}^w \prec_{n_3}^{pl} v.$$

# Termination

☞ Depth reduction introduces more equalities.

# Termination

☞ Depth reduction introduces more equalities.

For example, $x \prec^l t$ becomes

$$car(x) = car(t) \ \wedge \ cdr(x) \prec^{kb} cdr(t).$$

# Termination

☞ Depth reduction introduces more equalities.

For example, $x \prec^{l} t$ becomes

$$car(x) = car(t) \ \wedge \ cdr(x) \prec^{kb} cdr(t).$$

Why terminate?

# Termination

**REAL MEASURE**

Open Gap Order Literals: gap orders between ordinary terms.

# Termination

**REAL MEASURE**

Open Gap Order Literals: gap orders between ordinary terms.

**EXAMPLE**

| | | | | |
|---|---|---|---|---|
| ✔ | $u \prec_3^l v$ | $u \prec_3^p v$ | $u \prec_3^w v$ | $\dots$ |
| ✖ | $u \prec_3^l 1^w_{(u^w)}$ | $0^w_{(u^w)} \prec_3^p 1^w_{(u^w)}$ | $0^w_{(v^w)} \prec_3^l v$ | $\dots$ |

# Termination

**REAL MEASURE**

Open Gap Order Literals: gap orders between ordinary terms.

**EXAMPLE**

| | | | | |
|---|---|---|---|---|
| ✔ | $u <^l_3 v$ | $u <^p_3 v$ | $u <^w_3 v$ | $\dots$ |
| ✖ | $u <^l_3 1^w_{(u^w)}$ | $0^w_{(u^w)} <^p_3 1^w_{(u^w)}$ | $0^w_{(v^w)} <^l_3 v$ | $\dots$ |

**REASON**

- No transformation generates new OGOLs.
- The final elimination step removes at least one OGOL.
- Without OGOLs, the depths of terms strictly decrease!

# Example

Consider the KBO on LISP list structure parameterized with

$$w(cons) = w(nil) = 1 \quad and \quad nil \prec^{\sigma} cons.$$

Consider the formula

$$\exists x \left( car(x) \prec^{l}_{2} cdr(cdr(x)) \;\wedge\; cdr(cdr(car(x))) \prec^{l}_{3} y \right)$$

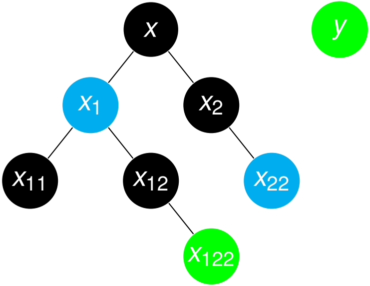
where $depth(x) = 3$.

# Example

$$\exists x \left( car(x) \prec^{l}_{2} cdr(cdr(x)) \ \wedge \ cdr(cdr(car(x))) \prec^{l}_{3} y \right)$$

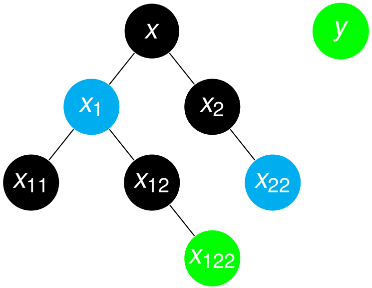
# Example

$$\exists x \left( car(x) \prec_2^l cdr(cdr(x)) \; \wedge \; cdr(cdr(car(x))) \prec_3^l y \right)$$



# Example

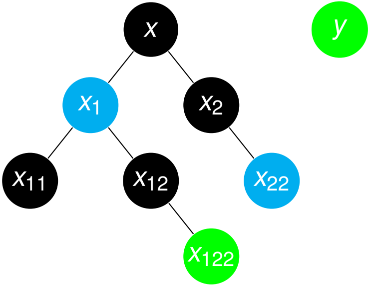
$$\exists x \left( car(x) <^{l}_{2} cdr(cdr(x)) \ \wedge \ cdr(cdr(car(x))) <^{l}_{3} y \right)$$




$$\begin{aligned} x_1 &: car(x) \\ x_2 &: cdr(x) \\ x_{11} &: car(car(x)) \\ x_{12} &: cdr(car(x)) \\ x_{22} &: cdr(cdr(x)) \\ x_{122} &: cdr(cdr(car(x))) \end{aligned}$$

# Example

$$\exists x \Big( car(x) \prec^l_2 cdr(cdr(x)) \;\wedge\; cdr(cdr(car(x))) \prec^l_3 y \Big)$$




$$\begin{aligned}
x_1 &: car(x)\\
x_2 &: cdr(x)\\
x_{11} &: car(car(x))\\
x_{12} &: cdr(car(x))\\
x_{22} &: cdr(cdr(x))\\
x_{122} &: cdr(cdr(car(x)))
\end{aligned}$$

Solution: $x = ?$

# Example

Select $x$.

# Example

Select $x$.

Decompose $x$ in terms of $car(x)$ and $cdr(x)$. We have

$$\exists x \exists x_1 \exists x_2 \Big( car(x) = x_1 \ \wedge \ cdr(x) = x_2$$
$$\wedge \ \boxed{car(x)} \prec^{l}_{2} cdr(\boxed{cdr(x)}) \ \wedge \ cdr(cdr(\boxed{car(x)})) \prec^{l}_{3} y \Big).$$

# Example

Select $x$.
Decompose $x$ in terms of $car(x)$ and $cdr(x)$. We have

$$\exists x \exists x_1 \exists x_2 \Big( car(x) = x_1 \ \wedge \ cdr(x) = x_2$$
$$\wedge \ \boxed{car(x)} \prec^l_2 cdr(\boxed{cdr(x)}) \ \wedge \ cdr(cdr(\boxed{car(x)})) \prec^l_3 y \Big).$$

Simplification.

$$\exists x_1 \exists x_2 \Big( \boxed{x_1} \prec^l_2 cdr(\boxed{x_2}) \ \wedge \ cdr(cdr(\boxed{x_1})) \prec^l_3 y \Big),$$

where $depth(x_1) = 2$ and $depth(x_2) = 1$.

# Example

Continue with

$$\exists x_1 \exists x_2 \big( \boxed{x_1 <^{l}_{2} cdr(x_2)} \wedge cdr(cdr(x_1)) <^{l}_{3} y \big).$$

# Example

Continue with

$$\exists x_1 \exists x_2 \Big( \boxed{x_1 <^l_2 cdr(x_2)} \wedge cdr(cdr(x_1)) <^l_3 y \Big).$$

Select $x_1$.

# Example

Continue with

$$\exists x_1 \exists x_2 \Big( \boxed{x_1 <^l_2 cdr(x_2)} \wedge cdr(cdr(x_1)) <^l_3 y \Big).$$

Select $x_1$.
Decompose $x_1$.

$$\exists x_1 \exists x_2 \Big( \boxed{\boxed{car(x_1)} = car(cdr(x_2)) \wedge \boxed{cdr(x_1)} <^l_2 cdr(cdr(x_2))}$$

$$\wedge\, cdr(\boxed{cdr(x_1)}) <^l_3 y \Big).$$

# Example

Continue with

$$\exists x_1 \exists x_2 \Big( \boxed{x_1 <^l_2 cdr(x_2)} \;\wedge\; cdr(cdr(x_1)) <^l_3 y \Big).$$

Select $x_1$.
Decompose $x_1$.

$$\exists x_1 \exists x_2 \Big( \boxed{car(x_1) = car(cdr(x_2)) \;\wedge\; cdr(x_1) <^l_2 cdr(cdr(x_2))} \;\wedge\; cdr(cdr(x_1)) <^l_3 y \Big).$$

Simplification.

$$\exists x_2 \exists x_{11} \exists x_{12} \Big( x_{11} = car(cdr(x_2)) \;\wedge\; x_{12} <^l_2 cdr(cdr(x_2)) \;\wedge\; cdr(x_{12}) <^l_3 y \Big)$$

# Example

Continue with

$$\exists x_2 \exists x_{11} \exists x_{12} \Big( x_{11} = car(cdr(x_2)) \ \wedge \ \boxed{x_{12} <^l_2 cdr(cdr(x_2))} \ \wedge \ \boxed{cdr(x_{12}) <^l_3 y} \Big)$$

# Example

Continue with

$$\exists x_2 \exists x_{11} \exists x_{12} \Big( x_{11} = car(cdr(x_2)) \ \wedge \ \boxed{x_{12} <_2^l cdr(cdr(x_2))} \ \wedge \ \boxed{cdr(x_{12}) <_3^l y} \Big)$$

Elimination. Since $depth(x_{11}) = 0$, we have

$$\exists x_2 \exists x_{12} \Big( \boxed{x_{12} <_2^l cdr(cdr(x_2))} \ \wedge \ \boxed{cdr(x_{12}) <_3^l y} \Big)$$

# Example

Continue with

$$\exists x_2 \exists x_{12} \left( \boxed{x_{12} <^l_2 cdr(cdr(x_2))} \ \wedge \ cdr(x_{12}) <^l_3 y \right).$$

# Example

Continue with

$$\exists x_2 \exists x_{12} \Big( \boxed{x_{12} <_2^l cdr(cdr(x_2))} \;\wedge\; cdr(x_{12}) <_3^l y \Big).$$

Select $x_{12}$.

# Example

Continue with

$$\exists x_2 \exists x_{12} \left( \boxed{x_{12} <^{l}_{2} cdr(cdr(x_2))} \;\wedge\; cdr(x_{12}) <^{l}_{3} y \right).$$

Select $x_{12}$.
Decompose $x_{12}$.

$$\exists x_2 \exists x_{12} \left( \boxed{\boxed{car(x_{12})} = car(cdr(cdr(x_2))) \;\wedge\; \boxed{car(x_{12})} <^{l}_{2} cdr(cdr(x_2))} \;\wedge\; \boxed{cdr(x_{12})} <^{l}_{3} y \right).$$

# Example

Continue with

$$\exists x_2 \exists x_{12} \left( x_{12} <_2^l cdr(cdr(x_2)) \ \wedge \ cdr(x_{12}) <_3^l y \right).$$

Select $x_{12}$.
Decompose $x_{12}$.

$$\exists x_2 \exists x_{12} \left( car(x_{12}) = car(cdr(cdr(x_2))) \ \wedge \ car(x_{12}) <_2^l cdr(cdr(x_2)) \ \wedge \ cdr(x_{12}) <_3^l y \right).$$

Simplification.

$$\exists x_2 \exists x_{121} \exists x_{122} \left( x_{121} = car(cdr(cdr(x_2))) \ \wedge \ x_{121} <_2^l cdr(cdr(x_2)) \ \wedge \ x_{122} <_3^l y \right)$$

# Example

Continue with

$$\exists x_2 \exists x_{121} \exists x_{122} \Big( x_{121} = car(cdr(cdr(x_2))) \;\wedge\; x_{122} <^l_2 cdr(cdr(x_2)) \;\wedge\; x_{122} <^l_3 y \Big)$$

# Example

Continue with

$$\exists x_2 \exists x_{121} \exists x_{122} \Big( x_{121} = car(cdr(cdr(x_2))) \ \wedge \ \boxed{x_{122} <^l_2 cdr(cdr(x_2))} \ \wedge \ \boxed{x_{122} <^l_3 y} \Big)$$

Elimination. Since $depth(x_{121}) = 0$, we have

$$\exists x_2 \exists x_{122} \Big( \boxed{x_{122} <^l_2 cdr(cdr(x_2))} \ \wedge \ \boxed{x_{122} <^l_3 y} \Big).$$

# Example

Continue with

$$\exists x_2 \exists x_{122} \left( \; x_{122} <^{l}_{2} cdr(cdr(x_2)) \;\wedge\; x_{122} <^{l}_{3} y \; \right).$$

# Example

Continue with

$$\exists x_2 \exists x_{122} \left( x_{122} <^l_2 cdr(cdr(x_2)) \ \wedge \ x_{122} <^l_3 y \right).$$

Elimination. Guessing a gap order completion, we have

$$\exists x_2 \exists x_{122} \left( x_{122} <^l_2 cdr(cdr(x_2)) <^l_1 y \right),$$

# Example

Continue with

$$\exists x_2 \exists x_{122} \left( \boxed{x_{122} <^l_2 cdr(cdr(x_2)) \ \wedge \ x_{122} <^l_3 y} \right).$$

Elimination. Guessing a gap order completion, we have

$$\exists x_2 \exists x_{122} \left( \boxed{\boxed{x_{122}} <^l_2 cdr(cdr(x_2)) <^l_1 y} \right),$$

which simplifies to

$$\exists x_2 \left( \boxed{0^w_{((cdr(cdr(x_2)))^w)}} <^l_2 cdr(cdr(x_2)) <^l_1 y \right).$$

# Example

Continue with

$$\exists x_2 \exists x_{122} \left( x_{122} <^l_2 cdr(cdr(x_2)) \;\wedge\; x_{122} <^l_3 y \right).$$

Elimination. Guessing a gap order completion, we have

$$\exists x_2 \exists x_{122} \left( x_{122} <^l_2 cdr(cdr(x_2)) <^l_1 y \right),$$

which simplifies to

$$\exists x_2 \left( 0^w_{((cdr(cdr(x_2)))^w)} <^l_2 cdr(cdr(x_2)) <^l_1 y \right).$$

☞ The number of OGOLs reduced to 1!

# Example

Continue with

$$\exists\, x_2 \left( 0^w_{((\, cdr(cdr(x_2)) \,)^w)} \prec^l_2 cdr(cdr(x_2)) \prec^l_1 y \right).$$

# Example

Continue with

$$\exists \boxed{x_2} \left( 0^w_{((\boxed{cdr(cdr(x_2))})^w)} \prec^l_2 \boxed{cdr(cdr(x_2))} \prec^l_1 y \right).$$

Depth Reduction. Repeating twice the DEPTH-REDUCTION subprocedure, we have

$$\exists \boxed{x_{222}} \left( 0^w_{(\boxed{x_{222}}^w)} \prec^l_2 \boxed{x_{222}} \prec^l_1 y \right).$$

# Example

Continue with

$$\exists\, x_{222} \left( 0^{w}_{\left((x_{222})^{w}\right)} \prec^{l}_{2} x_{222} \prec^{l}_{1} y \right).$$

# Example

Continue with

$$\exists\, x_{222} \left( 0^w_{\left((x_{222})^w\right)} \prec^l_2 x_{222} \prec^l_1 y \right).$$

Reduce term quantifiers to integer quantifiers.

$$\exists\, z \left( 0^w_{(z)} \prec^l_3 y \wedge Tree(z) \right).$$

# Example

Continue with

$$\exists\, x_{222} \left( 0^{w}_{((x_{222})^{w})} \prec^{l}_{2} x_{222} \prec^{l}_{1} y \right).$$

Reduce term quantifiers to integer quantifiers.

$$\exists\, z \left( 0^{w}_{(z)} \prec^{l}_{3} y \wedge Tree(z) \right).$$

# Example

Continue with

$$\exists\, \boxed{x_{222}} \left( \boxed{0^w_{(\boxed{(x_{222})^w})} \prec^l_2 \boxed{x_{222}} \prec^l_1 y} \right).$$

Reduce term quantifiers to integer quantifiers.

$$\exists\, \boxed{z} \left( \boxed{0^w_{(\boxed{z})} \prec^l_3 y \wedge Tree(\boxed{z})} \right).$$

Eliminate integer quantifiers.

$$0^w_{(\boxed{y^w})} \prec^l_3 y \wedge Tree(\boxed{y^w}).$$

# Example

Continue with

$$\exists\, x_{222} \left( 0^{w}_{((x_{222})^{w})} <^{l}_{2} x_{222} <^{l}_{1} y \right).$$

Reduce term quantifiers to integer quantifiers.

$$\exists\, z \left( 0^{w}_{(z)} <^{l}_{3} y \wedge Tree(z) \right).$$

Eliminate integer quantifiers.

$$0^{w}_{(y^{w})} <^{l}_{3} y \wedge Tree(y^{w}).$$

As $0^{w}_{(y^{w})} <^{l}_{3} y \Rightarrow Tree(y^{w})$, we have

$$0^{w}_{(y^{w})} <^{l}_{3} y.$$

# Example

In summary,

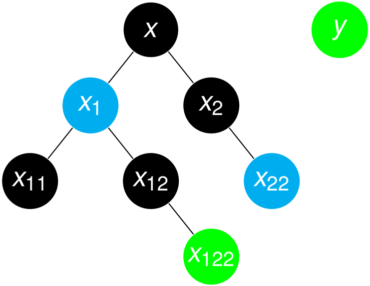
$$0^{w}_{(y^{w})} \prec^{l}_{3} y \implies$$
$$\exists x \left( car(x) \prec^{l}_{2} cdr(cdr(x)) \;\wedge\; cdr(cdr(car(x))) \prec^{l}_{3} y \right)$$

# Example

In summary,

$$0^{w}_{(y^{w})} \prec^{l}_{3} y \implies$$

$$\exists x \left( car(x) \prec^{l}_{2} cdr(cdr(x)) \ \wedge \ cdr(cdr(car(x))) \prec^{l}_{3} y \right)$$

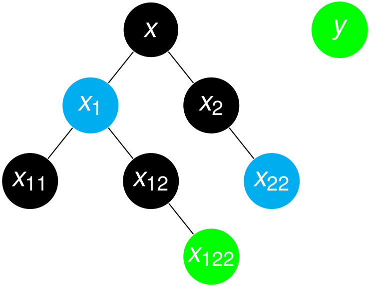


# Example

In summary,

$$0^{w}_{(y^{w})} \prec^{l}_{3} y \Longrightarrow$$

$$\exists x \Big( car(x) \prec^{l}_{2} cdr(cdr(x)) \;\wedge\; cdr(cdr(car(x))) \prec^{l}_{3} y \Big)$$



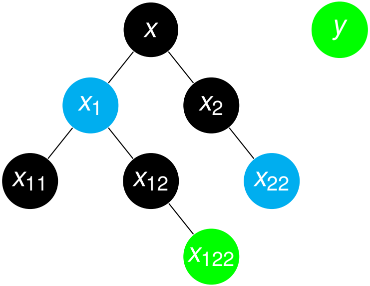

$$\begin{aligned}
x_1 &: car(x)\\
x_2 &: cdr(x)\\
x_{11} &: car(car(x))\\
x_{12} &: cdr(car(x))\\
x_{22} &: cdr(cdr(x))\\
x_{122} &: cdr(cdr(car(x)))
\end{aligned}$$

# Example

In summary,

$$0^{w}_{(y^{w})} \prec^{l}_{3} y \implies$$

$$\exists x \left( car(x) \prec^{l}_{2} cdr(cdr(x)) \;\wedge\; cdr(cdr(car(x))) \prec^{l}_{3} y \right)$$




$$\begin{aligned} x_1 &: car(x) \\ x_2 &: cdr(x) \\ x_{11} &: car(car(x)) \\ x_{12} &: cdr(car(x)) \\ x_{22} &: cdr(cdr(x)) \\ x_{122} &: cdr(cdr(car(x))) \end{aligned}$$

Solution: $x_{122} = 0^{w}_{(y^{w})}$!

# Outline

# Conclusions and Open Problems

- Orderings with Partial Precedence

  Knuth-Bendix Order with Partial Precedence

# Conclusions and Open Problems

- Orderings with Partial Precedence

  Knuth-Bendix Order with Partial Precedence

- ☞ Orderings on Nonground Term Domain

  Knuth-Bendix Order on Nonground Term Domain

# Conclusions and Open Problems

- Orderings with Partial Precedence

  Knuth-Bendix Order with Partial Precedence

- Orderings on Nonground Term Domain

  Knuth-Bendix Order on Nonground Term Domain

- ☞ Multiple Orderings on One Term Domain

  Two Knuth-Bendix Orders

# Conclusions and Open Problems

- Orderings with Partial Precedence

  Knuth-Bendix Order with Partial Precedence

- Orderings on Nonground Term Domain

  Knuth-Bendix Order on Nonground Term Domain

- Multiple Orderings on One Term Domain

  Two Knuth-Bendix Orders

Difficulty: Lack of technique to deal with partial orderings.

# Thank you for your attention!

Questions?

**Hubert Comon.**
Solving symbolic ordering constraints.
*International Journal of Foundations of Computer Science*, 1(4):387–411, 1990.

**Hubert Comon and Ralf Treinen.**
The first-order theory of lexicographic path orderings is undecidable.
*Theoretical Computer Science*, 176(1-2):67–87, 1997.

**Jean-Pierre Jouannaud and Mitsuhiro Okada.**
Satisfiability of systems of ordinal notation with the subterm property is decidable.
In *18th International Colloquium on Automata, Languages and Programming*, volume 510 of *LNCS*, pages 455–468. Springer-Verlag, 1991.

**C. Kirchner, H. Kirchner, and M. Rusinowitch.**
Deduction with symbolic constraints.
*Revue Francaise d' Intelligence Artificielle*, 4(3):9–52, 1990.

Special issue on automated deduction.

**Konstantin Korovin and Andrei Voronkov.**
A decision procedure for the existential theory of term algebras with the Knuth-Bendix ordering.
In *Proc. 15th IEEE Symp. Logic in Comp. Sci.*, pages 291 – 302, 2000.

**Konstantin Korovin and Andrei Voronkov.**
Knuth-Bendix constraint solving is NP-complete.
In *Proceedings of 28th International Colloquium on Automata, Languages and Programming (ICALP'01)*, volume 2076 of *Lecture Notes in Computer Science*, pages 979–992. Springer-Verlag, 2001.

**Konstantin Korovin and Andrei Voronkov.**
The decidability of the first-order theory of the Knuth-Bendix order in the case of unary signatures.
In *Proceedings of the 22th Conference on Foundations of Software Technology and Theoretical Computer Science, (FSTTCS'02)*,

volume 2556 of *Lecture Notes in Computer Science*, pages 230–240. Springer-Verlag, 2002.

**Robert Nieuwenhuis.**
Simple LPO constraint solving methods.
*Information Processing Letters*, 47(2):65–69, 1993.

**Paliath Narendran and Michael Rusinowitch.**
The theory of total unary RPO is decidable.
In *CL 2000*, volume 1861 of *Lecture Notes in Artificial Intelligence*, pages 660–672. Springer-Verlag, 2000.

**Paliath Narendran, Michael Rusinowitch, and Rakesh M. Verma.**
RPO constraint solving is in NP.
In *Proceedings of the 12th International Workshop on Computer Science Logic (CSL 98)*, volume 1584 of *LNCS*, pages 385 – 398. Springer-Verlag, 1999.

**Ralf Treinen.**
A new method for undecidability proofs of first order theories.

*Journal of Symbolic Computation*, 14:437–457, 1992.

**Ting Zhang, Henny B. Sipma, and Zohar Manna.**
The decidability of the first-order theory of term algebras with Knuth-Bendix order.
In Robert Nieuwenhuis, editor, *the 20th International Conference on Automated Deduction (CADE'05)*, volume 3632 of *LNCS*, pages 131–148. Springer-Verlag, 2005.

**Hubert Comon.**
Solving symbolic ordering constraints.
*International Journal of Foundations of Computer Science*, 1(4):387–411, 1990.

**Hubert Comon and Ralf Treinen.**
The first-order theory of lexicographic path orderings is undecidable.
*Theoretical Computer Science*, 176(1-2):67–87, 1997.

**Jean-Pierre Jouannaud and Mitsuhiro Okada.**

Satisfiability of systems of ordinal notation with the subterm property is decidable.
In *18th International Colloquium on Automata, Languages and Programming*, volume 510 of *LNCS*, pages 455–468. Springer-Verlag, 1991.

**C. Kirchner, H. Kirchner, and M. Rusinowitch.**
Deduction with symbolic constraints.
*Revue Francaise d' Intelligence Artificielle*, 4(3):9–52, 1990.
Special issue on automated deduction.

**Konstantin Korovin and Andrei Voronkov.**
A decision procedure for the existential theory of term algebras with the Knuth-Bendix ordering.
In *Proc. 15th IEEE Symp. Logic in Comp. Sci.*, pages 291 – 302, 2000.

**Konstantin Korovin and Andrei Voronkov.**
Knuth-Bendix constraint solving is NP-complete.
In *Proceedings of 28th International Colloquium on Automata, Languages and Programming (ICALP'01)*, volume 2076 of *Lecture*

*Notes in Computer Science*, pages 979–992. Springer-Verlag, 2001.

**Konstantin Korovin and Andrei Voronkov.**
The decidability of the first-order theory of the Knuth-Bendix order in the case of unary signatures.
In *Proceedings of the 22th Conference on Foundations of Software Technology and Theoretical Computer Science, (FSTTCS'02)*, volume 2556 of *Lecture Notes in Computer Science*, pages 230–240. Springer-Verlag, 2002.

**Robert Nieuwenhuis.**
Simple LPO constraint solving methods.
*Information Processing Letters*, 47(2):65–69, 1993.

**Paliath Narendran and Michael Rusinowitch.**
The theory of total unary RPO is decidable.
In *CL 2000*, volume 1861 of *Lecture Notes in Artificial Intelligence*, pages 660–672. Springer-Verlag, 2000.

**Paliath Narendran, Michael Rusinowitch, and Rakesh M. Verma.**
RPO constraint solving is in NP.
In *Proceedings of the 12th International Workshop on Computer Science Logic (CSL 98)*, volume 1584 of *LNCS*, pages 385 – 398. Springer-Verlag, 1999.

**Ralf Treinen.**
A new method for undecidability proofs of first order theories.
*Journal of Symbolic Computation*, 14:437–457, 1992.

**Ting Zhang, Henny B. Sipma, and Zohar Manna.**
The decidability of the first-order theory of term algebras with Knuth-Bendix order.
In Robert Nieuwenhuis, editor, *the 20th International Conference on Automated Deduction (CADE'05)*, volume 3632 of *LNCS*, pages 131–148. Springer-Verlag, 2005.