Knuth-Bendix Order and Its Decidability

Ting Zhang

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

Logic Colloquium 2008
July 6th, 2008
Outline

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

\[
\frac{A \lor C \quad \neg A' \lor C'}{(C \lor C')\sigma} \quad \sigma = \text{mgu}(A, A') \quad (\forall B \in C \cup C') \left[ A\sigma \not< B\sigma \right]
\]
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:
  \[
  \frac{A \lor C \quad \neg A' \lor C'}{\neg mgu(A, A')} \quad (C \lor C') \sigma
  \]
  \[
  \forall B \in C \cup C' \left[ A \sigma \not< B \sigma \right]
  \]

- **Ordered Rewriting.** To orient commutative equations:
  \[
  x + y \rightarrow y + x \quad \text{if} \quad (x + y) \sigma > (y + x) \sigma
  \]
  \[
  y + x \rightarrow x + y \quad \text{if} \quad (x + y) \sigma < (y + x) \sigma
  \]
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:
  \[
  \frac{A \lor C \quad \neg A' \lor C'}{(C \lor C')\sigma} \quad \sigma = \text{mgu}(A, A') \\
  (\forall B \in C \cup C') \left[ A\sigma \nless B\sigma \right]
  \]

- **Ordered Rewriting.** To orient commutative equations:
  \[
  x + y \rightarrow y + x \quad \text{if} \quad (x + y)\sigma > (y + x)\sigma
  \\
  y + x \rightarrow x + y \quad \text{if} \quad (x + y)\sigma < (y + x)\sigma
  \]

**Fundamental:** Satisfiability Problem of Ordering Constraints
Beyond Existential Fragments

**Total Simplification Rule:** [KKR90, CT97]

\[
\frac{s \rightarrow t \mid c}{s[v]_p \rightarrow t \mid (c \land c' \land s|_p = u)}
\]

\[
(u \rightarrow v \mid c')
\]
Beyond Existential Fragments

**TOTAL SIMPLIFICATION RULE:** [KKR90, CT97]

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

**IT STATES THAT**

\( s \rightarrow t \mid c \) is simplified (at position \( p \)) to

\( s[v]_p \rightarrow t \mid (c \land c' \land s|_p = u) \)

by \( u \rightarrow v \mid c' \) provided

\[
\mathcal{T} \models \forall V(s) \left( c \rightarrow \exists V(u) \left( c' \land s|_p = u \right) \right),
\]

which necessarily involves quantifier alternation.
## Widely Used Orderings

<table>
<thead>
<tr>
<th></th>
<th>Syntactic Nature</th>
<th>Hybrid Nature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RPO</td>
<td>KBO</td>
</tr>
<tr>
<td></td>
<td>MPO</td>
<td>LPO</td>
</tr>
<tr>
<td>syntactic precedence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>multiset ordering</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lexicographical ordering</td>
<td></td>
<td></td>
</tr>
<tr>
<td>numerical ordering</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Widely Used Orderings

<table>
<thead>
<tr>
<th></th>
<th>Syntactic Nature</th>
<th>Hybrid Nature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RPO</td>
<td>KBO</td>
</tr>
<tr>
<td></td>
<td>MPO</td>
<td>LPO</td>
</tr>
<tr>
<td>syntactic precedence</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>multiset ordering</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>lexicographical ordering</td>
<td></td>
<td></td>
</tr>
<tr>
<td>numerical ordering</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Widely Used Orderings

<table>
<thead>
<tr>
<th>Syntactic Nature</th>
<th>Hybrid Nature</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPO</td>
<td>KBO</td>
</tr>
<tr>
<td>MPO</td>
<td>LPO</td>
</tr>
<tr>
<td>syntactic</td>
<td>✓</td>
</tr>
<tr>
<td>precedence</td>
<td></td>
</tr>
<tr>
<td>multiset</td>
<td>✓</td>
</tr>
<tr>
<td>ordering</td>
<td></td>
</tr>
<tr>
<td>lexicographical</td>
<td></td>
</tr>
<tr>
<td>ordering</td>
<td></td>
</tr>
<tr>
<td>numerical</td>
<td></td>
</tr>
<tr>
<td>ordering</td>
<td></td>
</tr>
</tbody>
</table>
# Widely Used Orderings

<table>
<thead>
<tr>
<th></th>
<th>Syntactic Nature</th>
<th>Hybrid Nature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RPO</td>
<td>KBO</td>
</tr>
<tr>
<td></td>
<td>MPO</td>
<td>LPO</td>
</tr>
<tr>
<td>syntactic precedence</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>multiset ordering</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>lexicographical ordering</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>numerical ordering</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
## Decidability Status

<table>
<thead>
<tr>
<th></th>
<th>MPO</th>
<th>LPO</th>
<th>KBO</th>
</tr>
</thead>
<tbody>
<tr>
<td>QFT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UQT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GQT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

QFT: Quantifier-free Theory.

UQT: Unary Quantified Theory.

GQT: General Quantified Theory.
## Decidability Status

<table>
<thead>
<tr>
<th></th>
<th>MPO</th>
<th>LPO</th>
<th>KBO</th>
</tr>
</thead>
<tbody>
<tr>
<td>QFT</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>[JO91] [NRV99]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UQT</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[NR00]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GQT</td>
<td>?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

QFT: Quantifier-free Theory.

UQT: Unary Quantified Theory.

GQT: General Quantified Theory.
## Decidability Status

<table>
<thead>
<tr>
<th></th>
<th>MPO</th>
<th>LPO</th>
<th>KBO</th>
</tr>
</thead>
<tbody>
<tr>
<td>QFT</td>
<td>✓</td>
<td><img src="checkmark.png" alt="image" /></td>
<td><img src="crossmark.png" alt="image" /></td>
</tr>
<tr>
<td></td>
<td>[JO91] [NRV99]</td>
<td>[Com90] [Nie93]</td>
<td><img src="crossmark.png" alt="image" /></td>
</tr>
<tr>
<td>UQT</td>
<td>✓</td>
<td><img src="checkmark.png" alt="image" /></td>
<td><img src="crossmark.png" alt="image" /></td>
</tr>
<tr>
<td></td>
<td>[NR00]</td>
<td><img src="crossmark.png" alt="image" /></td>
<td><img src="crossmark.png" alt="image" /></td>
</tr>
<tr>
<td>GQT</td>
<td>?</td>
<td><img src="crossmark.png" alt="image" /></td>
<td><img src="crossmark.png" alt="image" /></td>
</tr>
<tr>
<td></td>
<td><img src="crossmark.png" alt="image" /></td>
<td><img src="crossmark.png" alt="image" /></td>
<td><img src="crossmark.png" alt="image" /></td>
</tr>
</tbody>
</table>

QFT: Quantifier-free Theory.

UQT: Unary Quantified Theory.

GQT: General Quantified Theory.
## Decidability Status

<table>
<thead>
<tr>
<th></th>
<th>MPO</th>
<th>LPO</th>
<th>KBO</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>QFT</strong></td>
<td><img src="sch%C3%B6nheit" alt="✓" /> [JO91] [NRV99]</td>
<td><img src="sch%C3%B6ne" alt="✓" /> [Com90] [Nie93]</td>
<td><img src="sch%C3%B6nheit" alt="✓" /> [KV00] [KV01]</td>
</tr>
<tr>
<td><strong>UQT</strong></td>
<td><img src="sch%C3%B6nheit" alt="✓" /> [NR00]</td>
<td><img src="sch%C3%B6ne" alt="✓" /> [NR00]</td>
<td><img src="sch%C3%B6nheit" alt="✓" /> [KV02]</td>
</tr>
<tr>
<td><strong>GQT</strong></td>
<td>?</td>
<td><img src="sch%C3%B6nheit" alt="✗" /> [Tre92, CT97]</td>
<td>?</td>
</tr>
</tbody>
</table>

QFT: Quantifier-free Theory.

UQT: Unary Quantified Theory.

GQT: General Quantified Theory.
### Decidability Status

<table>
<thead>
<tr>
<th></th>
<th>MPO</th>
<th>LPO</th>
<th>KBO</th>
</tr>
</thead>
<tbody>
<tr>
<td>QFT</td>
<td>✓ [JO91]</td>
<td>✓ [Com90]</td>
<td>✓ [KV00]</td>
</tr>
<tr>
<td></td>
<td>[NRV99]</td>
<td>[Nie93]</td>
<td>[KV01]</td>
</tr>
<tr>
<td>UQT</td>
<td>✓ [NR00]</td>
<td>✓ [NR00]</td>
<td>✓ [KV02]</td>
</tr>
<tr>
<td>GQT</td>
<td>?</td>
<td>✓ [Tre92, CT97]</td>
<td>✓ [ZSM05]</td>
</tr>
</tbody>
</table>

QFT: Quantifier-free Theory.
UQT: Unary Quantified Theory.
GQT: General Quantified Theory.
Our Approach

Reduce term constraints to integer constraints.

Our Approach

- Reduce term constraints to integer constraints.
  

- Reduce term quantifiers to integer quantifiers.
  
Our Approach

Reduction from term domain to integer domain!

- Reduce term constraints to integer constraints.
  

- Reduce term quantifiers to integer quantifiers.

Term Algebras

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

$\mathcal{TA}$: The term domain.
Term Algebras

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

- $\mathcal{T}A$: The term domain.

- $C$: set of constructors: $\alpha, \beta, \gamma, \ldots.$
**Term Algebras**

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

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

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

- $\mathcal{TA}$: The term domain.
- $C$: set of constructors: $\alpha, \beta, \gamma, \ldots$.
- $\mathcal{A}$: set of constants: $a, b, c, \ldots$. $\mathcal{A} \subseteq C$.
- $S$: set of selectors: $\alpha = (s_{1}^{\alpha}, \ldots, s_{k}^{\alpha})$. 

Knuth-Bendix Order and Its Decidability
A term algebra $\mathcal{T}A : \langle \mathcal{T}A; C, \mathcal{A}, S, \mathcal{T} \rangle$ consists of

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

Note that each element in $\mathcal{T}A$ is uniquely generated by constructors.
Term Algebras

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

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

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

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

NOTE THAT
Each element in $\mathcal{TA}$ is uniquely generated by constructors.
LISP List

\[
\langle \text{list}; \{\text{cons, nil}\}; \{\text{nil}\}; \{\text{car, cdr}\}; \{\text{Is}_{\text{nil}}, \text{Is}_{\text{cons}}\} \rangle
\]
**LISP List**

**SIGNATURE**

\[ \langle \text{list}; \{\text{cons, nil}\}; \{\text{nil}\}; \{\text{car, cdr}\}; \{\text{Is}_{\text{nil}}, \text{Is}_{\text{cons}}\} \rangle \]

**AXIOMS**

\[
\begin{align*}
\text{Is}_{\text{nil}}(x) & \leftrightarrow \neg \text{Is}_{\text{cons}}(x) \\
x & = \text{car}(\text{cons}(x, y)) \\
y & = \text{cdr}(\text{cons}(x, y)) \\
\text{Is}_{\text{nil}}(x) & \leftrightarrow \{\text{car, cdr}\}^+(x) = x \\
\text{Is}_{\text{cons}}(x) & \leftrightarrow \text{cons}(\text{car}(x), \text{cdr}(x)) = x
\end{align*}
\]
A Knuth-Bendix order (KBO) $\prec^{kb}$ is parametrically defined with

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

$w(\alpha(t_1, \ldots, t_k)) = w(\alpha) + \sum_{i=1}^{k} w(t_i)$. 
A Knuth-Bendix order (KBO) $\prec^{kb}$ is parametrically defined with

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

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

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

- $w(u) < w(v)$
Knuth-Bendix Order

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

- $w(u) < w(v)$
- $w(u) = w(v)$ and $\text{type}(u) <^\sigma \text{type}(v)$
Knuth-Bendix Order

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

- $w(u) < w(v)$
- $w(u) = w(v)$ and $\text{type}(u) <^{\sigma} \text{type}(v)$
- $w(u) = w(v)$, $u \equiv \alpha(u_1, \ldots, u_k)$, $v \equiv \alpha(v_1, \ldots, v_k)$, and
  
  $\exists i \left( 1 \leq i \leq k \land u_i <^{kb} v_i \land \forall j \left( 1 \leq j < i \rightarrow u_j = v_j \right) \right)$. 

Consider the KBO on LISP list structure parameterized with

\[ w(\text{cons}) = w(\text{nil}) = 1 \quad \text{and} \quad \text{nil} \prec_\sigma \text{cons}. \]
Example: Knuth-Bendix Order

Consider the KBO on LISP list structure parameterized with

\[ w(\text{cons}) = w(\text{nil}) = 1 \quad \text{and} \quad \text{nil} \prec_\sigma \text{cons}. \]
Outline

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

Suffices to eliminate $\exists$-quantifiers from primitive formulas

$$\exists \overline{x} \ (A_1(\overline{x}) \land \ldots \land A_n(\overline{x})),$$

where $A_i(\overline{x}) (1 \leq i \leq n)$ are literals.
Quantifier Elimination

- Suffices to eliminate \( \exists \)-quantifiers from primitive formulas

\[ \exists \bar{x} \ ( A_1(\bar{x}) \land \ldots \land A_n(\bar{x}) ), \]

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

- Suffices to assume \( A_i \not\equiv x = t \) if \( x \notin t \), because

\[ \exists x \ ( x = t \land \varphi(x, \bar{y}) ) \leftrightarrow \varphi(t, \bar{y}). \]
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 $s_1(\ldots (s_n(x)\ldots))$ 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 $\text{depth}_\varphi(x)$, 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.
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 $\text{depth}_\varphi(x)$, 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, $\text{car}(x) \neq \text{cdr}(x)$ abbreviates $\text{car}(x) \neq \text{cdr}(x) \land \text{Is}_{\text{cons}}(x)$.
Main Idea

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

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

Main Idea

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

- Depth Reduction. Transforming $\varphi(x, \bar{y})$ into a solved form amounts to peeling off selectors in front of $x$, since
  
  $\varphi(x, \bar{y})$ solved in $x$ if and only if $\text{depth} \varphi(x) = 0$. 
\( \varphi(x, \bar{y}) \) is solved in \( x \) if it is in the form

\[
\bigwedge_{i \leq m} u_i <_{kb} x \land \bigwedge_{j \leq n} x <_{kb} v_j \land \varphi'(\bar{y}),
\]

where \( x \) does not appear in \( u_i, v_i \) and \( \varphi' \).
Solved Form

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

\[
\bigwedge_{i \leq m} u_i \prec_{kb} x \land \bigwedge_{j \leq n} x \prec_{kb} v_j \land \varphi'(\bar{y}),
\]

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

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

\[
\varphi'(\bar{y}) \land \bigwedge_{i \leq m, j \leq n} u_i \prec_{2} v_j
\]

where \( x \prec_{n} y \), called gap order, states there is an increasing chain from \( x \) to \( y \) of length at least \( n \).
All occurrences of $x$ have depth greater than 0.

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

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

which can be rewritten to

$$\exists x_1, \ldots, \exists x_k \varphi'(x_1, \ldots, x_k, \bar{y}).$$
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 $x <_{nb}^k t$ and $t <_{nb}^k x$ using

$$s_1^\alpha(x), \ldots, s_k^\alpha(x).$$
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 $x <_{\kappa b}^n t$ and $t <_{\kappa b}^n x$ using

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

Then apply the reduction as in Case 1!
Language Extensions

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

Add boundary functions to delineate gap orders.

Add Presburger arithmetic explicitly to represent the weight function.

Extend all aforementioned notions to tuples of terms.
Language Extensions

- Decompose $<^{kb}$ into three disjoint suborders $<^w$, $<^p$ and $<^l$.
- Extend $<^w$, $<^p$ and $<^l$ to $<^w_n$, $<^p_n$ and $<^l_n$, 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**

\[ u <^w v \iff w(u) < w(v) \]
Suborders

WEIGHT ORDER

\[ u \prec^w v \quad \text{def} \quad w(u) < w(v) \]

PRECEDENCE ORDER

\[ u \prec^p v \quad \text{def} \quad w(u) = w(s) \land \text{type}(u) <^\sigma \text{type}(v) \]
Suborders

WEIGHT ORDER

\[ u <^w v \quad \overset{\text{def}}{=} \quad w(u) < w(v) \]

PRECEDENCE ORDER

\[ u <^p v \quad \overset{\text{def}}{=} \quad w(u) = w(s) \land \text{type}(u) <^\sigma \text{type}(v) \]

LEXICOGRAPHICAL ORDER

\[ u <^l v \quad \overset{\text{def}}{=} \quad w(u) = w(v) \land \text{type}(u) = \text{type}(v) \land u <^\text{kb} v \]
Gap Orders

\[ u <_{k^n}^b v \quad \overset{\text{def}}{=} \quad (\exists u_1 \cdots \exists u_n) \left( u <_{k^b}^b u_1 <_{k^b}^b \cdots <_{k^b}^b u_n \leq_{k^b}^b v \right) \]
Gap Orders

**KB GAP ORDER**

\[ u \prec_{kn}^v \overset{def}{=} (\exists u_1 \cdots \exists u_n)(u \prec_{kb}^v u_1 \prec_{kb} \cdots \prec_{kb}^v u_n \preceq_{kb}^v v) \]

**WEIGHT GAP ORDER**

\[ u \prec_{wn}^v \overset{def}{=} u \prec_{kn}^v v \wedge u \prec_{wn}^v v \]
Gap Orders

**KB GAP ORDER**

\[ u \prec_{kn}^kb v \quad \overset{\text{def}}{=} \quad (\exists u_1 \cdots \exists u_n)\left(u \prec_{kn}^kb u_1 \prec_{kn}^kb \cdots \prec_{kn}^kb u_n \preceq_{kn}^kb v\right) \]

**WEIGHT GAP ORDER**

\[ u \prec_{wn}^w v \quad \overset{\text{def}}{=} \quad u \prec_{kn}^kb v \land u \prec_{wn}^w v \]

**PRECEDENCE GAP ORDER**

\[ u \prec_{pn}^p v \quad \overset{\text{def}}{=} \quad u \prec_{kn}^kb v \land u \prec_{pn}^p v \]
**Gap Orders**

**KB GAP ORDER**

\[ u <_{kn}^kb v \overset{\text{def}}{=} (\exists u_1 \cdots \exists u_n)\left( u <_{kb}^k u_1 <_{kb}^k \cdots <_{kb}^k u_n \leq_{kb}^k v \right) \]

**WEIGHT GAP ORDER**

\[ u <_{wn}^w v \overset{\text{def}}{=} u <_{kn}^k v \land u <_{wn}^w v \]

**PRECEDENCE GAP ORDER**

\[ u <_{pn}^p v \overset{\text{def}}{=} u <_{kn}^k v \land u <_{pn}^p v \]

**LEXICOGRAPHICAL GAP ORDER**

\[ u <_{ln}^l v \overset{\text{def}}{=} u <_{kn}^k v \land u <_{ln}^l v \]
Boundary Functions

\[ 0^w, 1^w : \mathbb{N} \rightarrow TA; \quad 0^p, 1^p : \mathbb{N}^2 \rightarrow TA \] such that

\[ 0^w(n) : \text{the smallest term of weight } n \]

\[ 0^p(n, p) : \text{the smallest term of weight } n \text{ and type } \alpha_p \]

\[ 1^w(n) : \text{the largest term of weight } n \]

\[ 1^p(n, p) : \text{the largest term of weight } n \text{ and type } \alpha_p \]
Boundary Functions

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

$$0^w(n) : \text{the smallest term of weight } n$$

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

$$1^w(n) : \text{the largest term of weight } n$$

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

EXAMPLE

$$u \prec^w_5 v \iff \bigvee_{n_1 + n_2 + n_3 = 5} u \prec^p_{n_1} 1^w_{(u^w)} \prec^w_{n_2} 0^w_{(v^w)} \prec^p_{n_3} v$$
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**

\[ 0^w_{(x)} \overset{p}{<}^n 1^w_{(x)} \iff CNT_n(x) \]
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**

$0_x^w \prec_n 1_x^w \iff CNT_n(x)$

**NOTE THAT**

$CNT_n(x)$ is expressible in Presburger arithmetic.
Knuth-Bendix Order with Presburger Arithmetic

\[ \mathbf{KBO}^+ = \langle TA, PA, (.)^w, \prec^\#, 0^*(\ldots), 1^*(\ldots) \rangle \]

where \( n \in \mathbb{N}, \# \in \{w, p, l\}, * \in \{w, p\}, \)

\( (.)^w : \) weight function,

\( \prec^\# : \) gap orders,

\( 0^*(\ldots), 1^*(\ldots) : \) boundary functions
Knuth-Bendix Order with Presburger Arithmetic

\[ KBO^+ = \langle TA, PA, (.)^w, <^n_#, 0^*(\ldots), 1^*(\ldots) \rangle \]

where \( n \in \mathbb{N} \), \( \# \in \{w, p, l\} \), \( \ast \in \{w, p\} \),

\[ (.)^w : \text{weight function}, \]

\[ <^n_# : \text{gap orders}, \]

\[ 0^*(\ldots), 1^*(\ldots) : \text{boundary functions} \]

EXAMPLE

\[ \exists x : TA \left( 0^w_{(x^w)} <_2^l x <^l_3 1^w_{(x^w)} \right) \]
Quantifier Elimination for Knuth-Bendix Order

INPUT: \((\exists \bar{x}) \varphi(\bar{x}, \bar{y})\)
while \(\bar{x} \neq \emptyset\) do
  if \((\forall x \in \bar{x}) \text{depth}_\varphi(x) > 0\) then
     Depth Reduction:
     \[
     \begin{bmatrix}
     \text{variable selection} \\
     \text{decomposition} \\
     \text{simplification}
     \end{bmatrix}
     \]
  else \((\exists x \in \bar{x}) \text{depth}_\varphi(x) = 0\}\)
     Elimination
  end if
end while
Variable Selection

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

Variable Selection

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

**NOTE THAT**

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

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

$$\exists x_1 \ldots \exists x_k \exists \bar{x} \left( \bigwedge_{1 \leq i \leq k} s^\alpha_i(x) = x_i \land \varphi(\bar{x}, \bar{y}) \right).$$
Decomposition

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

\[
\exists x_1 \ldots \exists x_k \exists \bar{x} \left( \bigwedge_{1 \leq i \leq k} s_i^\alpha(x) = x_i \land \varphi(\bar{x}, \bar{y}) \right).
\]

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

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

\[
\exists x_1 \ldots \exists x_k \exists \bar{x} \left( \bigwedge_{1 \leq i \leq k} s_\alpha^i(x) = x_i \land \varphi(\bar{x}, \bar{y}) \right).
\]

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

Resuliting in

\[
\exists x_1 \ldots \exists x_k \exists \bar{x} \left( \bigwedge_{1 \leq i \leq k} s_\alpha^i(x) = x_i \land \varphi'(s_1^\alpha(x), \ldots, s_k^\alpha(x), (\bar{x} \setminus x), \bar{y}) \right).
\]
Simplification

\[ \exists x_1 \ldots \exists x_k \exists \bar{x} \left( \bigwedge_{1 \leq i \leq k} s_i^\alpha(x) = x_i \land \varphi'(s_1^\alpha(x), \ldots, s_k^\alpha(x), (\bar{x} \setminus x), \bar{y}) \right). \]
Simplification

\[ \exists x_1 \ldots \exists x_k \exists \bar{x} \left( \bigwedge_{1 \leq i \leq k} s_i^\alpha(x) = x_i \land \varphi'(s_1^\alpha(x), \ldots, s_k^\alpha(x), (\bar{x} \setminus x), \bar{y}) \right) \]

- Replace \( s_i^\alpha(x) \) by \( x_i \) in \( \varphi' \).
\[ \exists x_1 \ldots \exists x_k \exists \bar{x} \left( \bigwedge_{1 \leq i \leq k} s_{i}^{\alpha}(x) = x_i \land \varphi'(s_{1}^{\alpha}(x), \ldots, s_{k}^{\alpha}(x), (\bar{x} \setminus x), \bar{y}) \right). \]

- Replace \( s_{i}^{\alpha}(x) \) by \( x_i \) in \( \varphi' \).
- Remove \( \bigwedge_{1 \leq i \leq k} s_{i}^{\alpha}(x) = x_i \) from the matrix.
Simplification

\[ \exists x_1 \ldots \exists x_k \exists \bar{x} \left( \bigwedge_{1 \leq i \leq k} s_i^\alpha(x) = x_i \land \varphi'(s_1^\alpha(x), \ldots, s_k^\alpha(x), (\bar{x} \setminus x), \bar{y}) \right). \]

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

\[ \exists x_1 \ldots \exists x_k \exists \bar{x} \left( \bigwedge_{1 \leq i \leq k} s_i^\alpha(x) = x_i \land \varphi'(s_1^\alpha(x), \ldots, s_k^\alpha(x), (\bar{x} \setminus x), \bar{y}) \right) \].

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

Resulting in

\[ \exists x_1 \ldots \exists x_k \exists (\bar{x} \setminus x) (\varphi'((\bar{x} \setminus x), x_1, \ldots, x_k, \bar{y})) \).
Elimination

\[ \exists x \left( \bigwedge_{i \leq m} u_i <^k b x \land \bigwedge_{j \leq n} x <^k b v_j \land \varphi'(\bar{y}) \right), \]
Elimination

\[ \exists x \left( \bigwedge_{i \leq m} u_i \prec^b x \land \bigwedge_{j \leq n} x \prec^b v_j \land \varphi'(\overline{y}) \right), \]

which simplifies to

\[ u_i' \prec^b v_j' \land \varphi'(\overline{y}) \land “u_i' is the greatest of \{u_i | i \leq m\}” \land “v_j' is the smallest of \{v_j | j \leq n\}”. \]
Elimination of Equalities.

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

- **Elimination of Equalities.**
  \[ \exists x \left( x = 0^w_{\text{car}(x)^w+5} \land \text{car}(x) <^p 4 \text{cdr}(x) \right). \]

- **Simplification of Selector Terms.**
  \[ \text{car}(0^w_{\text{car}(x)^w}). \]
Technical Tricks

- Elimination of Equalities.

\[ \exists x \left( x = 0^w_{\text{car}(x)^w + 5} \land \text{car}(x) \prec^p c\text{dr}(x) \right). \]

- Simplification of Selector Terms.

\[ \text{car}(0^w_{\text{car}(x)^w}). \]

- Elimination of Negations.

\[ \neg \left( \text{car}(x) \prec^w_3 \text{cdr}(x) \right). \]
Technical Tricks

- Elimination of Equalities.
  \[ \exists x \left( x = 0^w_{(\text{car}(x))^w+5} \land \text{car}(x) <^p_4 \text{cdr}(x) \right). \]

- Simplification of Selector Terms.
  \[ \text{car}(0^w_{(\text{car}(x))^w}). \]

- Elimination of Negations.
  \[ \neg \left( \text{car}(x) <^w_3 \text{cdr}(x) \right). \]

- Termination.
Elimination of Equalities

\[ \exists x \left( x = 0^w_{(\text{car}(x))^w + 5} \land \text{car}(x) \prec^p_4 \text{cdr}(x) \right) \]
Elimination of Equalities

\[ \exists x \left( x = 0^w_{\left((\text{car}(x))^w + 5\right)} \land \text{car}(x) <^p_4 \text{cdr}(x) \right) \]

Reverse Substitution \(\Rightarrow\)

\[ \exists x \left( x = 0^w_{\left((\text{car}(x))^w + 5\right)} \land \text{car}(0^w_{\left((\text{car}(x))^w + 5\right)}) <^p_4 \text{cdr}(0^w_{\left((\text{car}(x))^w + 5\right)}) \right) \]
Elimination of Equalities

\[ \exists x \left( x = 0^w_{((\text{car}(x))^w + 5)} \land \text{car}(0^w_{((\text{car}(x))^w + 5)}) \prec_4 \text{cdr}(0^w_{((\text{car}(x))^w + 5)}) \right) \]
Elimination of Equalities

CONTINUE WITH

\[ \exists x \left( x = 0^w_{((\text{car}(x))^w + 5)} \land \text{car}(0^w_{((\text{car}(x))^w + 5)}) <^p 4 \text{cdr}(0^w_{((\text{car}(x))^w + 5)}) \right) \]

Reduction to Integer Quantifiers \[\Rightarrow \]

\[ \exists (\text{car}(x))^w \left( \text{Tree}((\text{car}(x))^w + 5) \land \text{Tree}((\text{cdr}(x))^w + 5) \land (\alpha)^w + (\text{car}(x))^w + (\text{cdr}(x))^w = (\text{car}(x))^w + 5 \land \text{car}(0^w_{((\text{car}(x))^w + 5)}) <^p 4 \text{cdr}(0^w_{((\text{car}(x))^w + 5)}) \right) \]
Elimination of Equalities

\[
\exists (\text{car}(x))^w \left( \begin{array}{l}
\text{Tree}((\text{car}(x))^w + 5) \land \text{Tree}((\text{cdr}(x))^w + 5) \\
\land (\alpha)^w + (\text{car}(x))^w + (\text{cdr}(x))^w = (\text{car}(x))^w + 5 \\
\land \text{car}(0^w_{((\text{car}(x))^w + 5)}) <^p 4 \text{cdr}(0^w_{((\text{car}(x))^w + 5)})
\end{array} \right)
\]
Elimination of Equalities

\[
\exists (\text{car}(x))^w \left( \begin{array}{l}
\text{Tree}((\text{car}(x))^w + 5) \land \text{Tree}((\text{cdr}(x))^w + 5) \\
\land (\alpha)^w + (\text{car}(x))^w + (\text{cdr}(x))^w = (\text{car}(x))^w + 5 \\
\land \text{car}(0^w((\text{car}(x))^w+5)) \prec_p 4 \text{cdr}(0^w((\text{car}(x))^w+5))
\end{array} \right)
\]

Renaming \( \Rightarrow \)

\[
\exists z \exists y \left( \begin{array}{l}
\text{Tree}(z) \land \text{Tree}(y) \\
\land (\alpha)^w + z + y = z + 5 \\
\land \text{car}(0^w(z)) \prec_p 4 \text{cdr}(0^w(z))
\end{array} \right)
\]
Simplification of Selector Terms

EXAMPLE

\[ \text{car}(0^w((\text{car}(x))^w)) \]
Simplification of Selector Terms

**EXAMPLE**

\[ \text{car}(0^w_{((\text{car}(x))^w)}) \]

which simplifies to

\[ 0^w_{f_{\text{car}}((\text{car}(x))^w)} \]

where \( f_{\text{car}}(\cdot) \) is an integer function expressible in Presburger arithmetic.
Elimination of Negations


\neg \left( car(x) \prec^w_3 cdr(x) \right)

simplifies to
Elimination of Negations

\[ \neg (\text{car}(x) \prec^w_3 \text{cdr}(x)) \]

simplifies to

\[ \text{cdr}(x) \prec^w_1 \text{car}(x) \]
\[ \lor (\text{cdr}(x))^w = (\text{car}(x))^w \]
\[ \lor \text{car}(x) \preceq^w_1 \text{cdr}(x) \]
\[ \lor \text{car}(x) \preceq^w_2 \text{cdr}(x). \]
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 is subtle as many complexity measures increase.

- Depth reduction increases the depth of other variables.

For example, \( x \neq t \) becomes

\[
\bigvee_{1 \leq i \leq k} s_i^\alpha(t) \neq x_i \lor \neg l s_\alpha(t).
\]
Termination

Depth reduction introduces more existential quantifiers.
Depth reduction introduces more existential quantifiers.

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

\[
\exists x_1 \ldots \exists x_k \exists \bar{x} \left( l_s(\alpha)(x) \land \bigwedge_{1 \leq i \leq k} s_i^\alpha(x) = x_i \land \varphi(\bar{x}, \bar{y}) \right).
\]
Depth reduction introduces more order literals.
Depth reduction introduces more order literals.

For example, \( u \prec_{5}^{w} v \) becomes

\[
\bigvee_{n_1 + n_2 + n_3 = 5} u \prec_{n_1}^{pl} t_{(u^w)}^{w} \prec_{n_2}^{w} o_{(v^w)}^{w} \prec_{n_3}^{pl} v.
\]
Depth reduction introduces more equalities.
Depth reduction introduces more equalities.

For example, $x <^l t$ becomes

\[ \text{car}(x) = \text{car}(t) \land \text{cdr}(x) <^{kb} \text{cdr}(t). \]
Depth reduction introduces more equalities.

For example, $x <^l t$ becomes

$$car(x) = car(t) \land cdr(x) <^{kb} cdr(t).$$

Why terminate?
Open Gap Order Literals: gap orders between ordinary terms.
Open Gap Order Literals: gap orders between ordinary terms.

Example:

- \( u \prec_3 v \)
- \( u \prec_3 1_{(u^w)}^w \)
- \( 0_{(u^w)}^w \prec_3 1_{(u^w)}^w \)
- \( 0_{(v^w)}^w \prec_3 1_{(v^w)}^w \)

- \( \checkmark \)

- \( \times \)
Open Gap Order Literals: gap orders between ordinary terms.

**REAL MEASURE**

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

**EXAMPLE**

- $u \prec^l_3 v$
- $u \prec^p_3 v$
- $u \prec^w_3 v$
- $u \prec^l_3 w(u_w)$
- $u \prec^p_3 w(u_w)$
- $u \prec^w_3 w(u_w)$
- $u \prec^l_3 w(v_w)$
- $u \prec^p_3 w(v_w)$
- $u \prec^w_3 w(v_w)$

**REASON**

- $u \prec^l_3 v$
- $u \prec^p_3 v$
- $u \prec^w_3 v$
- $u \prec^l_3 1_{(u_w)}$
- $u \prec^p_3 0_{(w_u)}$
- $u \prec^w_3 1_{(u_w)}$
- $u \prec^l_3 0_{(v_w)}$
- $u \prec^p_3 1_{(u_w)}$
- $u \prec^w_3 0_{(v_w)}$

...
Example

Consider the KBO on LISP list structure parameterized with

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

Consider the formula

\[ \exists x \left( \text{car}(x) \prec_2 \text{cdr}(\text{cdr}(x)) \land \text{cdr}(\text{cdr}(\text{car}(x))) \prec_3 y \right) \]

where \( \text{depth}(x) = 3 \).
Example

\[ \exists x \left( \text{car}(x) \prec_2 \text{cdr}(\text{cdr}(x)) \land \text{cdr}(\text{cdr}(\text{car}(x))) \prec_3 y \right) \]
Example

$$\exists x \left( \text{car}(x) <^1_2 \text{cdr}(\text{cdr}(x)) \land \text{cdr}(\text{cdr}(\text{car}(x))) <^1_3 y \right)$$

Solution:

$$x =? x_1 : \text{car}(x)$$

$$x_2 : \text{cdr}(x)$$

$$x_{11} : \text{car}(x_{12})$$

$$x_{12} : \text{cdr}(x_{12})$$

$$x_{22} : \text{cdr}(x_{22})$$

$$x_{122} : \text{cdr}(\text{car}(x_{122}))$$
Example

$$\exists x \left( \text{car}(x) \prec^l_2 \text{cdr}(\text{cdr}(x)) \land \text{cdr}(\text{cdr}(\text{car}(x))) \prec^l_3 y \right)$$

Solution:

- $x_1 : \text{car}(x)$
- $x_2 : \text{cdr}(x)$
- $x_{11} : \text{car}(\text{car}(x))$
- $x_{12} : \text{cdr}(\text{car}(x))$
- $x_{22} : \text{cdr}(\text{cdr}(x))$
- $x_{122} : \text{cdr}(\text{cdr}(\text{car}(x)))$
Example

\[ \exists x \left( \text{car}(x) \triangleleft_2 \text{cdr}(	ext{cdr}(x)) \land \text{cdr}(	ext{cdr}(	ext{car}(x))) \triangleleft_3 y \right) \]

Solution: \( x = ? \)
Example

Select $x$. 
Example

Select $x$.
Decompose $x$ in terms of $\text{car}(x)$ and $\text{cdr}(x)$. We have

$$\exists x \exists x_1 \exists x_2 \left( \text{car}(x) = x_1 \land \text{cdr}(x) = x_2 \land \text{car}(x) <^l_2 \text{cdr}(\text{cdr}(x)) \land \text{cdr}(\text{cdr}(\text{car}(x))) <^l_3 y \right).$$
Example

Select $x$.
Decompose $x$ in terms of $\text{car}(x)$ and $\text{cdr}(x)$. We have

$$\exists x \exists x_1 \exists x_2 \left( \text{car}(x) = x_1 \land \text{cdr}(x) = x_2 \right) \land \text{car}(x) <^l_2 \text{cdr}(\text{cdr}(x)) \land \text{cdr}(\text{cdr}(\text{car}(x))) <^l_3 y \right).$$

Simplification.

$$\exists x_1 \exists x_2 (x_1 <^l_2 \text{cdr}(x_2) \land \text{cdr}(\text{cdr}(x_1)) <^l_3 y),$$

where $\text{depth}(x_1) = 2$ and $\text{depth}(x_2) = 1$. 

46 / 57
Example

Continue with

\[ \exists x_1 \exists x_2 \left( x_1 \prec_2 \text{cdr}(x_2) \land \text{cdr} \left( \text{cdr} \left( x_1 \right) \right) \prec_3 y \right). \]
Example

Continue with

$$\exists x_1 \exists x_2 \left( x_1 \prec_l^2 cdr(x_2) \land cdr(cdr(x_1)) \prec^l_3 y \right).$$

Select $x_1$. 
Example

Continue with

\[ \exists x_1 \exists x_2 \left( x_1 \prec_{\text{c}}^l \text{cdr}(x_2) \land \text{cdr} \left( \text{cdr}(x_1) \right) \prec_{\text{c}}^l y \right). \]

Select \( x_1 \).

Decompose \( x_1 \).

\[ \exists x_1 \exists x_2 \left( \text{car}(x_1) = \text{car}(\text{cdr}(x_2)) \land \text{cdr}(x_1) \prec_{\text{c}}^l \text{cdr}(\text{cdr}(x_2)) \right. \]

\[ \left. \land \text{cdr} \left( \text{cdr}(x_1) \right) \prec_{\text{c}}^l y \right). \]
Example

Continue with

\[ \exists x_1 \exists x_2 \left( x_1 \prec^l_2 cdr(x_2) \land cdr(cdr(x_1)) \prec^l_3 y \right). \]

Select \( x_1 \).
Decompose \( x_1 \).

\[ \exists x_1 \exists x_2 \left( \text{car}(x_1) = \text{car}(cdr(x_2)) \land cdr(x_1) \prec^l_2 cdr(cdr(x_2)) \right. \]

\[ \left. \land cdr(cdr(x_1)) \prec^l_3 y \right). \]

Simplification.

\[ \exists x_2 \exists x_{11} \exists x_{12} \left( x_{11} = \text{car}(cdr(x_2)) \land x_{12} \prec^l_2 cdr(cdr(x_2)) \right. \]

\[ \left. \land cdr(x_{12}) \prec^l_3 y \right). \]
Example

Continue with

$$\exists x_2 \exists x_{11} \exists x_{12} \left( x_{11} = \text{car}(\text{cdr}(x_2)) \land x_{12} \prec_{\text{l}_2} \text{cdr}(	ext{cdr}(x_2)) \land \text{cdr}(x_{12}) \prec_{\text{l}_3} y \right)$$
Continue with

$$\exists x_2 \exists x_{11} \exists x_{12} \left( x_{11} = \text{car}(\text{cdr}(x_2)) \land x_{12} <^l_2 \text{cdr}(\text{cdr}(x_2)) \land \text{cdr}(x_{12}) <^l_3 y \right)$$

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

$$\exists x_2 \exists x_{12} \left( x_{12} <^l_2 \text{cdr}(\text{cdr}(x_2)) \land \text{cdr}(x_{12}) <^l_3 y \right)$$
Example

Continue with

$$\exists x_2 \exists x_{12} \left( x_{12} \prec^n_{2} \text{cdr}(\text{cdr}(x_2)) \land \text{cdr}(x_{12}) \prec^n_{3} y \right).$$
Example

Continue with

\[ \exists x_2 \exists x_{12} ( x_{12} \prec_2 \text{cdr(cdr}(x_2)) \land \text{cdr}(x_{12}) \prec_3 y ). \]

Select \( x_{12} \).
Example

Continue with

$$\exists x_2 \exists x_{12} \left( x_{12} \prec^l_2 \text{cdr}(\text{cdr}(x_2)) \land \text{cdr}(x_{12}) \prec^l_3 y \right).$$

Select $x_{12}$.

Decompose $x_{12}$.

$$\exists x_2 \exists x_{12} \left( \text{car}(x_{12}) = \text{car}(\text{cdr}(\text{cdr}(x_2))) \land \text{car}(x_{12}) \prec^l_2 \text{cdr}(\text{cdr}(x_2)) \land \text{cdr}(x_{12}) \prec^l_3 y \right).$$
Continue with

\[ \exists x_2 \exists x_{12} \left( x_{12} <_{2}^{l} cdr(cdr(x_2)) \land cdr(x_{12}) <_{3}^{l} y \right). \]

Select \( x_{12} \).

Decompose \( x_{12} \).

\[ \exists x_2 \exists x_{12} \left( \begin{array}{c}
\text{car}(x_{12}) = car(cdr(cdr(x_2))) \\
\land \text{car}(x_{12}) <_{2}^{l} cdr(cdr(x_2)) \\
\land cdr(x_{12}) <_{3}^{l} y
\end{array} \right). \]

Simplification.

\[ \exists x_2 \exists x_{121} \exists x_{122} \left( x_{121} = car(cdr(cdr(x_2))) \land x_{121} <_{2}^{l} cdr(cdr(x_2)) \land x_{122} <_{3}^{l} y \right). \]
Continue with

$$\exists x_2 \exists x_{121} \exists x_{122} \left( x_{121} = \text{car}(\text{cdr}(\text{cdr}(x_2))) \land x_{122} <^l_{2} \text{cdr}(\text{cdr}(x_2)) \land x_{122} <^l_{3} y \right)$$
Example

Continue with

\[ \exists x_2 \exists x_{121} \exists x_{122} \left( x_{121} = \text{car}(\text{cdr}(\text{cdr}(x_2))) \land x_{122} \prec^l_{2} \text{cdr}(\text{cdr}(x_2)) \right) \]

\[ \land x_{122} \prec^l_{3} y \]

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

\[ \exists x_2 \exists x_{122} \left( x_{122} \prec^l_{2} \text{cdr}(\text{cdr}(x_2)) \land x_{122} \prec^l_{3} y \right). \]
Example

Continue with

$$\exists x_2 \exists x_{122} \left( x_{122} \prec^l_2 \text{cdr}(\text{cdr}(x_2)) \land x_{122} \prec^l_3 y \right).$$
Example

Continue with

\[ \exists x_2 \exists x_{122} \left( x_{122} \prec^l \text{cdr}(\text{cdr}(x_2)) \land x_{122} \prec^l y \right). \]

Elimination. Guessing a gap order completion, we have

\[ \exists x_2 \exists x_{122} \left( x_{122} \prec^l \text{cdr}(\text{cdr}(x_2)) \prec^l y \right), \]
Example

Continue with

\[ \exists x_2 \exists x_{122} \left( x_{122} \preceq^l_2 \text{cdr} \left( \text{cdr} \left( x_2 \right) \right) \land x_{122} \preceq^l_3 y \right) \].

Elimination. Guessing a gap order completion, we have

\[ \exists x_2 \exists x_{122} \left( x_{122} \preceq^l_2 \text{cdr} \left( \text{cdr} \left( x_2 \right) \right) \preceq^l_1 y \right), \]

which simplifies to

\[ \exists x_2 \left( 0^w \left( \left( \text{cdr} \left( \text{cdr} \left( x_2 \right) \right) \right)^w \right) \preceq^l_2 \text{cdr} \left( \text{cdr} \left( x_2 \right) \right) \preceq^l_1 y \right). \]
Continue with

$$\exists x_2 \exists x_{122} \left( x_{122} \prec_{l}^{1} \text{cdr}(\text{cdr}(x_2)) \land x_{122} \prec_{l}^{3} y \right).$$

**Elimination.** Guessing a gap order completion, we have

$$\exists x_2 \exists x_{122} \left( x_{122} \prec_{l}^{1} \text{cdr}(\text{cdr}(x_2)) \prec_{l}^{1} y \right),$$

which simplifies to

$$\exists x_2 \left( 0^w \left( ((\text{cdr}(\text{cdr}(x_2)))^w \right) \prec_{l}^{2} \text{cdr}(\text{cdr}(x_2)) \prec_{l}^{1} y \right).$$

The number of OGOLs reduced to 1!
Example

Continue with

\[ \exists x_2 \left( 0^w \left( \left( \text{cdr}(\text{cdr}(x_2)) \right)^w \right) \preceq^l_2 \text{cdr}(\text{cdr}(x_2)) \preceq^l_1 y \right). \]
Continue with

\[ \exists x_2 \left( 0^w \left( \frac{\text{cdr(\text{cdr}(x_2))}}{w} \right)^w \right) \prec_2 \frac{\text{cdr(\text{cdr}(x_2))}}{\text{y}} \prec_1 y \).

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

\[ \exists x_{222} \left( 0^w \left( \frac{\text{x_{222}}}{w} \right)^w \right) \prec_2 \frac{x_{222}}{\text{y}} \prec_1 y \).
Example

Continue with

\[ \exists x_{222} \left( 0^w x_{222}^w \prec_l^2 x_{222} \prec_l^1 y \right). \]
Continue with

\[ \exists x_{222} \left( 0^w \left( (x_{222})^w \right) \prec_2 x_{222} \prec_1 y \right). \]

Reduce term quantifiers to integer quantifiers.

\[ \exists z \left( 0^w \left( z \right) \prec_3 y \land Tree\left( z \right) \right). \]
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.

\[ \exists z \left( 0^w \left( z \right) \prec^l_3 y \land \text{Tree}(z) \right). \]
Continue with

$$\exists x_{222} \left( 0^w \left( (x_{222})^w \right) \preceq _2 x_{222} \preceq _1 y \right) .$$

Reduce term quantifiers to integer quantifiers.

$$\exists z \left( 0^w \left( z \right) \preceq _3 y \land \text{Tree}(z) \right) .$$

Eliminate integer quantifiers.

$$0^w \left( y^w \right) \preceq _3 y \land \text{Tree}(y^w) .$$
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.

$$\exists z \left( 0^w \left( z \right) \prec^l_3 y \land Tree(z) \right).$$

Eliminate integer quantifiers.

$$0^w \left( y^w \right) \prec^l_3 y \land Tree(y^w).$$

As $$0^w_{(y^w)} \prec^l_3 y \Rightarrow Tree(y^w),$$ we have

$$0^w_{(y^w)} \prec^l_3 y.$$
Example

In summary,

\[ 0^w_{(y^w)} \prec_3^I y \implies \exists x \left( \text{car}(x) \prec_2^I \text{cdr}(\text{cdr}(x)) \land \text{cdr}(\text{cdr}(\text{car}(x))) \prec_3^I y \right) \]

\[ x_1^0 \] \[ x_11 \] \[ x_12 \] \[ x_122 \] \[ x_2 \] \[ x_22 \] \[ x_11 : \text{car}(\text{car}(x)) \] \[ x_12 : \text{cdr}(\text{car}(x)) \] \[ x_22 : \text{cdr}(\text{cdr}(x)) \] \[ x_122 : \text{cdr}(\text{cdr}(\text{car}(x))) \]
In summary,

\[ O^w_{(y^w)} <^3_l y \implies \exists x \left( \text{car}(x) <^2_l \text{cdr}(\text{cdr}(x)) \land \text{cdr}(\text{cdr}(\text{car}(x))) <^3_l y \right) \]
Example

In summary,

\[ O^w_{(w^w)} \prec^l_3 y \implies \exists x \left( \text{car}(x) \prec^l_2 \text{cdr}(\text{cdr}(x)) \land \text{cdr}(\text{cdr}(\text{car}(x))) \prec^l_3 y \right) \]

\[
\begin{align*}
x_1 & : \text{car}(x) \\
x_2 & : \text{cdr}(x) \\
x_{11} & : \text{car}(\text{car}(x)) \\
x_{12} & : \text{cdr}(\text{car}(x)) \\
x_{22} & : \text{cdr}(\text{cdr}(x)) \\
x_{122} & : \text{cdr}(\text{cdr}(\text{car}(x)))
\end{align*}
\]
In summary,

\[ 0^w_w <^l_3 y \implies \exists x \left( \text{car}(x) <^l_2 \text{cdr}(	ext{cdr}(x)) \land \text{cdr}(	ext{cdr}(	ext{car}(x))) <^l_3 y \right) \]

Solution: \( x_{122} = 0^w_w \)!
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. 

Hubert Comon and Ralf Treinen. 
The first-order theory of lexicographic path orderings is undecidable. 

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. 

C. Kirchner, H. Kirchner, and M. Rusinowitch. 
Deduction with symbolic constraints. 
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.

**Konstantin Korovin and Andrei Voronkov.**
Knuth-Bendix constraint solving is NP-complete.

**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)*,
Knuth-Bendix Order and Its Decidability


**Robert Nieuwenhuis.**
Simple LPO constraint solving methods.

**Paliath Narendran and Michael Rusinowitch.**
The theory of total unary RPO is decidable.

**Paliath Narendran, Michael Rusinowitch, and Rakesh M. Verma.**
RPO constraint solving is in NP.

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

**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.

**Hubert Comon and Ralf Treinen.**
The first-order theory of lexicographic path orderings is undecidable.

**Jean-Pierre Jouannaud and Mitsuhiro Okada.**
Satisfiability of systems of ordinal notation with the subterm property is decidable.


**C. Kirchner, H. Kirchner, and M. Rusinowitch.**
Deduction with symbolic constraints.
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.

**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


Paliath Narendran, Michael Rusinowitch, and Rakesh M. Verma.
RPO constraint solving is in NP.

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

Ting Zhang, Henny B. Sipma, and Zohar Manna.
The decidability of the first-order theory of term algebras with Knuth-Bendix order.